RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département de Génie Industriel

Thesis submitted in partial fulfilment of the requirements for the Industrial Engineering degree

Option: Data Science and Artificial Intelligence

Cash Flow Management Optimisation Using Statistical and Machine Learning Techniques.

Application : Client Company of PwC

Prepared by: :
Nacerdine Dounia Amira

Supervised by: :
Madame Bahia BOUCHAFAA

Publicly presented and defended on (17/07/2024)

Jury Composition

President	Dr. Iskander ZOUAGHI	MCA	ENP
Examiner	Dr. Hakim FOURAR LAIDI	MCA	ENP
Promoter	Dr. Bahia BOUCHAFAA	MCA	ENP

ENP 2024

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département de Génie Industriel

Thesis submitted in partial fulfilment of the requirements for the Industrial Engineering degree

Option: Data Science and Artificial Intelligence

Cash Flow Management Optimisation Using Statistical and Machine Learning Techniques.

Application : Client Company of PwC

Prepared by: :
Nacerdine Dounia Amira

Supervised by: :
Madame Bahia BOUCHAFAA

Publicly presented and defended on (17/07/2024)

Jury Composition

President	Dr. Iskander ZOUAGHI	MCA	ENP
Examiner	Dr. Hakim FOURAR LAIDI	MCA	ENP
Promoter	Dr. Bahia BOUCHAFAA	MCA	ENP

ENP 2024

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département de Génie Industriel

Thèse soumise en vue de l'obtention partielle du diplôme de Génie Industriel
Option : Data Science et Intelligence Artificielle

Optimisation de la Gestion des Flux de Trésorerie à l'aide de Techniques Statistiques et d'Apprentissage Automatique.

Application : Entreprise Cliente de PwC

Preparé par: :
Nacerdine Dounia Amira

Supervisé par : :
Madame Bahia BOUCHAFAA

Présentée et soutenue publiquement le (17/07/2024)

Jury Composition

Président	Dr. Iskander ZOUAGHI	MCA	ENP
Examinateur	Dr. Hakim FOURAR LAIDI	MCA	ENP
Encadrant	Dr. Bahia BOUCHAFAA	MCA	ENP

ENP 2024

ملخص

يهدف المشروع إلى تحسين إدارة التدفقات النقدية لشركة عميلة من خلال استخدام تقنيات التحليل التنبؤي المتقدمة، بما في ذلك نماذج التعلم الآلي والتنبؤ بالسلاسل الزمنية. الهدف هو توفير رؤى مستقبلية حول اتجاهات التدفقات النقدية، خاصة فيما يتعلق بالحسابات المدينة والحسابات الدائنة. هذا يتيح تخطيط مالي استراتيجي أفضل، وإدارة محسنة للسيولة، وتوزيع فعال للموارد، مما يعزز الاستقرار المالي والكفاءة التشغيلية للشركة.

الكلمات الرئيسية : التدفقات النقدية، التعلم الآلي، السلاسل الزمنية، الحسابات الدائنة، الحسابات المدينة، صور (برايس ووترهاوس كوبرز).

Résumé

Le projet vise à optimiser la gestion des flux de trésorerie pour une entreprise cliente en utilisant des techniques analytiques prédictives avancées, y compris des modèles d'apprentissage automatique et de prévision des séries temporelles. L'objectif est de fournir des perspectives futures sur les tendances des flux de trésorerie, en particulier pour les comptes clients et les comptes fournisseurs. Cela permet une meilleure planification financière stratégique, une gestion améliorée de la liquidité et une allocation efficace des ressources, renforçant ainsi la stabilité financière et l'efficacité opérationnelle de l'entreprise.

Mots-clés : Flux de trésorerie, Apprentissage automatique, Séries temporelles, Comptes fournisseurs, Comptes clients, PwC (PricewaterhouseCoopers).

Abstract

The project focuses on optimizing cash flow management for a client company by leveraging advanced predictive analytics, including machine learning and time series forecasting models. The aim is to provide future insights into cash flow trends, particularly for accounts receivable and accounts payable. This allows for better strategic financial planning, improved liquidity management, and efficient resource allocation, ultimately enhancing the firm's financial stability and operational efficiency.

Keywords : Cash Flow, Machine Learning, Time Series, Accounts Payable, Accounts Receivable, PwC (PricewaterhouseCoopers).

Dedications

As I pen down these dedications, I am filled with a mix of disbelief and nostalgia, realizing that my journey at the Polytechnic School is drawing to a close. These past five years have swiftly unfolded, leaving behind a treasure trove of unforgettable memories. I am profoundly grateful for the opportunity to have been part of this remarkable institution and for everyone who has made this journey a significant chapter in my life.

The list of individuals I wish to thank and dedicate my final graduation project to seems endless, and I will strive to remember each one.

To my beloved parents, Mum and Dad, and my little sister—my everything. Without you, I cannot fathom being who I am today.

To my grandparents, aunts, uncles, and cousins, who have always taken pride in my achievements and believed in me wholeheartedly.

In loving memory of my grandfathers—I wish you could see how far I have come. Your pride would have filled my heart.

To my friends: Manar, my lifelong companion through childhood and into adulthood; Ilhem, the friend with a gentle, big heart, who makes me feel a profound sense of belonging; Sabrin and Lina, with whom I embarked on this Polytechnic adventure; Yousra, Samah, Sihem, Boutheina—the best companions to start and end these three years with. Life would not have been as flavorful without you all, and the memories would not have been as vivid.

To the IEC, the finest club of all time, my experiences with you allowed me to embark on various adventures, learn abundantly, enjoy every moment, and make incredible acquaintances for which I am eternally grateful.

And to everyone who, in big ways or small, contributed to the success of the person behind this thesis.

Thank you all from the bottom of my heart.

Amira,

Acknowledgments

I express my sincere gratitude to Mrs. Bouchafaa for her invaluable advice, availability, and rigorous and consistent support, which were crucial for the completion of my final year project.

I would also like to extend my warm thanks to the president of the jury, Mr. Zouaghi, and the examiner, Mr. Fourar, for dedicating their time and sharing their expertise during the evaluation of this project.

I am equally grateful to my supervisor at PwC, Mr. Gassem for his guidance and assistance in addressing the project's challenges.

Additionally, I would like to thank the entire Consulting team at PwC for their support and help, which significantly contributed to my development.

Finally, I wish to express my appreciation to all the teachers in the Industrial Engineering department, who generously shared their expertise and knowledge throughout my education.

Contents

List of Tables

List of Figures

List of Abbreviations

- **AP** : Accounts Payable
- **AR** : Accounts Receivable
- **ACF** : Autocorrelation Function
- **ADF** : Augmented Dickey-Fuller
- **AI** : Artificial Intelligence
- **ARIMA** : AutoRegressive Integrated Moving Average
- **ARMA** : Autoregressive Moving Average
- **CapEx** : Capital Expenditures
- **CAGR** : Compound Annual Growth Rate
- **CART** : Classification and Regression Trees
- **CFF** : Cash Flow from Financing Activities
- **CFO** : Cash Flow from Operations
- **CFI** : Cash Flow from Investing Activities
- **EDA** : Exploratory Data Analysis
- **ERP** : Enterprise Resource Planning
- **FCF** : Free Cash Flow
- **GARCH** : Generalized Autoregressive Conditional Heteroskedasticity
- **GOSS** : Gradient-based One-Side Sampling
- **I2C** : Invoice-to-Cash
- **LB** : Ljung-Box
- **MA** : Moving Average
- **MAE** : Mean Absolute Error
- **ML** : Machine Learning
- **MSE** : Mean Squared Error

- **NWC** : Net Working Capital
- **O2C** : Order-to-Cash
- **PCA** : Principal Component Analysis
- **PACF** : Partial Autocorrelation Function
- **PP** : Phillips-Perron
- **Prophet** : Facebook Prophet Forecasting Model
- **PwC** : PricewaterhouseCoopers
- **R^2** : Coefficient of Determination
- **RBMs** : Restricted Boltzmann Machines
- **RBF** : Radial Basis Function
- **ROI** : Return on Investment
- **RMSE** : Root Mean Squared Error
- **SARIMA** : Seasonal AutoRegressive Integrated Moving Average
- **SAP** : Systems, Applications, and Products in Data Processing
- **SAP HANA** : Systems, Applications, and Products in Data Processing High-Performance Analytic Appliance
- **S/4HANA** : SAP Business Suite 4 SAP HANA
- **SQL** : Structured Query Language
- **SVR** : Support Vector Regression
- **SVM** : Support Vector Machines
- **TLS** : Tax and Legal Services
- **WACC** : Weighted Average Cost of Capital
- **XGB** : XGBoost

General Introduction

In the dynamic landscape of the financial sector, effective cash flow management has emerged as a cornerstone for sustaining business operations and achieving long-term growth. The financial sector plays a critical role in the global economy, encompassing a wide range of businesses that manage money, including banks, investment firms, insurance companies, and real estate firms. This sector is pivotal in facilitating economic activities, providing essential services such as savings and loans, investment opportunities, and risk management solutions.

Cash flow management is essential for the financial health of any organization. It involves tracking the inflow and outflow of cash to ensure that a company has enough liquidity to meet its obligations and invest in its growth. Effective cash flow management enables businesses to maintain operational stability, avoid insolvency, and capitalize on investment opportunities.

The primary objective of this study is to integrate advanced predictive statistical techniques to gain future insights and understand cash flow trends. By leveraging machine learning and time series analysis, this study aims to enhance the existing cash flow management practices for a client company of PwC. The ultimate goal is to provide a robust, data-driven approach to predicting and managing cash flows, thereby improving financial planning and decision-making processes.

The thesis is structured as follows:

First Chapter presents an overview of PwC, highlighting key figures, activities, and related information. This chapter will provide a comprehensive background on PwC's global presence, its range of services, and its market position. Key performance metrics and achievements will also be discussed to provide context for the firm's operations and capabilities.

Second Chapter delves into the concept of cash flow, exploring its importance and composition. This chapter will explain the fundamental aspects of cash flow, including operating, investing, and financing, along with an exploration of various cash flow metrics used in financial analysis.

Third Chapter focuses on the management of accounts receivable (AR) and accounts payable (AP), examining traditional practices, identifying areas for improvement, and defining key challenges. This chapter will discuss the role of AR and AP in cash flow management, highlighting common practices and strategies used to optimize these processes. Potential inefficiencies and pain points in traditional AR and AP management will be identified, setting the stage for the proposed enhancements.

Fourth and Fifth Chapters provide a detailed explanation of machine learning and time series analysis, exploring theoretical aspects of model architectures and mathematical presentations. Chapter 4 will introduce machine learning concepts, and techniques. Various machine learning models will be discussed. Chapter 5 will focus on time series analysis, covering key methods such as ARIMA, and SARIMA. The mathematical foundations of these models will be explained, along with their relevance to cash flow forecasting.

Sixth Chapter presents the proposed solution for optimizing cash flow management, detailing the steps followed from data collection to model evaluation. This chapter will outline the methodology used in the study, including data preprocessing, feature engineering, model selection, and validation. The implementation of machine learning and time series models will be described, and the results of the predictive analysis will be presented. The effectiveness of the proposed solution will be assessed based on its accuracy and impact on cash flow management practices.

The final section will reflect on the achievements of the study, discuss its limitations, and propose areas for further investigation. The added value of the project to the client company and the broader financial sector will be emphasized, showcasing the potential benefits of integrating advanced predictive techniques into cash flow management.

Chapter 1

Presentation of the Host Organization

1.1 Introduction

In this chapter, I will provide a comprehensive overview of PricewaterhouseCoopers (PwC), the firm where I completed my internship, focusing on its global operations and particularly its activities in Algeria. This section will also offer an in-depth examination of the consulting department where my internship occurred, emphasizing its critical role within the firm and the broad spectrum of services it provides. Furthermore, this chapter aims to clarify and restate the context of my study.

1.2 Consulting definition

Consulting is a professional service provided by experts who offer specialized advice and guidance to organizations aiming to enhance their performance and efficiency. Typically, consulting firms focus on areas such as management, strategy, operations, technology, and human resources. Consultants analyze existing organizational problems, develop plans for improvement, and assist in the implementation of those solutions. The field of consulting has grown to accommodate a wide range of specialties, each tailored to the specific needs of different industries such as healthcare, finance, and technology. This versatility enables consultants to bring innovative solutions to complex challenges, leveraging their deep industry knowledge and analytical skills. Additionally, consulting services are not only aimed at addressing short-term issues but also at devising long-term strategies that foster sustainable growth and competitiveness. As a result, consulting plays a crucial role in driving organizational change, optimizing operations, and facilitating the adoption of new technologies and processes that keep businesses ahead in a rapidly evolving economic landscape.

1.3 Analysis of the consulting services market

- The consulting services market is projected to grow from an estimated $323.88 billion in 2024 to $431.89 billion by 2029, at a compound annual growth rate (CAGR) of 4.96% during the forecast period. Management consulting firms provide services that help organizations enhance their efficiency. There is an increasing demand for management consulting services due to strong economic growth in European markets, regulatory reforms in the financial sector, outsourcing of backend operations to low-cost economies, and public sector investments.

- The United States is the largest revenue-generating consulting services market in the world, as it hosts the largest global consulting firms catering to a wide range of end-user vertical sectors. Additionally, the highly volatile American economy, coupled with ongoing government regulatory reforms, compels businesses to seek management consulting providers for assistance with their financial operations across the country [18].

Metric	Value
Study Period	2019 - 2029
Market Size (2024)	USD 323.88 billion
Market Size (2029)	USD 431.89 billion
CAGR (2024-2029)	4.96%
Fastest Growing Market	Asia-Pacific
Largest Market	North America
Key Players	Accenture, Capgemini, EY, Deloitte, PwC

Table 1.1: Overview of the Global Consulting Services Market (2019-2029) [18]

1.4 Overview of PwC Firm

1.4.1 PwC International

PricewaterhouseCoopers (PwC) is one of the "Big Four" audit and consulting firms globally, alongside Deloitte, Ernst & Young, and KPMG. PwC operates as a vast network of UK-based firms specializing in auditing, accounting expertise, and consulting, with a focus on sector-specific approaches to meet the unique needs of businesses.

PricewaterhouseCoopers (PwC), a leading name among the "Big Four" audit and consulting firms, continues to demonstrate significant growth and presence globally. As of June 30, 2023, PwC reported impressive revenues of $53.1 billion for the fiscal year. The firm boasts a vast network, with 364,232 professionals spread across 688 cities in 151 countries, showcasing its extensive global reach and commitment to serving a diverse client base. Notably, PwC has successfully catered to 87% of the Fortune Global 500 companies, highlighting its prominence and trust within the corporate world. With over 178,000 clients worldwide, PwC remains a pivotal force in shaping global business practices and innovations in numerous sectors.

Metric	Value
Revenue ($ billions)	53.1
Professionals	364,232
Cities	688
Countries	151
Fortune Global 500 Companies (%)	87
Clients Worldwide	178,000

Table 1.2: PwC Global Presence and Operations (2023)

Figure 1.1: PwC Logo

History of the Firm:

PwC leverages its expertise to help today's companies ensure their future success. This is supported by an experience accumulated over more than 160 years of the firm's existence:

- **1849:** Auditor Samuel Lowell Price opens a law firm in London.
- **1854:** William Cooper establishes his own firm in London, which later becomes Cooper Brothers.
- **1865:** Price, Holyland, and Waterhouse join forces.
- **1874:** The firm is renamed Price, Waterhouse & Co.
- **1898:** Robert H. Montgomery, William M. Lybrand, Adam A. Ross Jr., and his brother T. Edward Ross establish Lybrand, Ross Brothers and Montgomery.
- **1957:** Cooper Brothers & Co (UK), McDonald, Currie and Co (Canada), and Lybrand, Ross Bros & Montgomery (USA) merge to form Coopers & Lybrand.
- **1982:** Launch of Price Waterhouse World Firm.
- **1990:** Coopers & Lybrand merges with Deloitte Haskins & Sells in several countries around the world.
- **1998:** Global merger of Price Waterhouse and Coopers & Lybrand to create PricewaterhouseCoopers.

- **2002:** PricewaterhouseCoopers concludes the sale of its management consulting department, PwC Consulting, to IBM.

- **2004:** PricewaterhouseCoopers implements the Connected Thinking methodology.

- **2008:** Tenth anniversary of the merger of PricewaterhouseCoopers.

- **2010:** PricewaterhouseCoopers formally shortens its brand name to PwC but legally remains PricewaterhouseCoopers.

- **2014:** In April, PwC merges with the international consulting firm Booz & Company.

1.4.1.1 PwC's Alliances and Ecosystems

PwC has established partnerships with a wide array of firms across various sectors, leveraging these alliances to enhance its service offerings and business solutions. Significant partners include major technology firms like Microsoft, with whom PwC collaborates to drive digital transformation and cultural change through innovative solutions such as Azure Cloud and Microsoft Dynamics 365. They also work closely with Salesforce, focusing on optimizing sales and marketing operations, enhancing user adoption of Salesforce technologies, and driving business transformation.

Other notable partners of PwC include Adobe, Amazon Web Services (AWS), Google, Oracle, SAP, and Workday, each partnership focusing on different aspects of business improvement, from cloud solutions and customer experience enhancements to operational efficiency and advanced analytics.

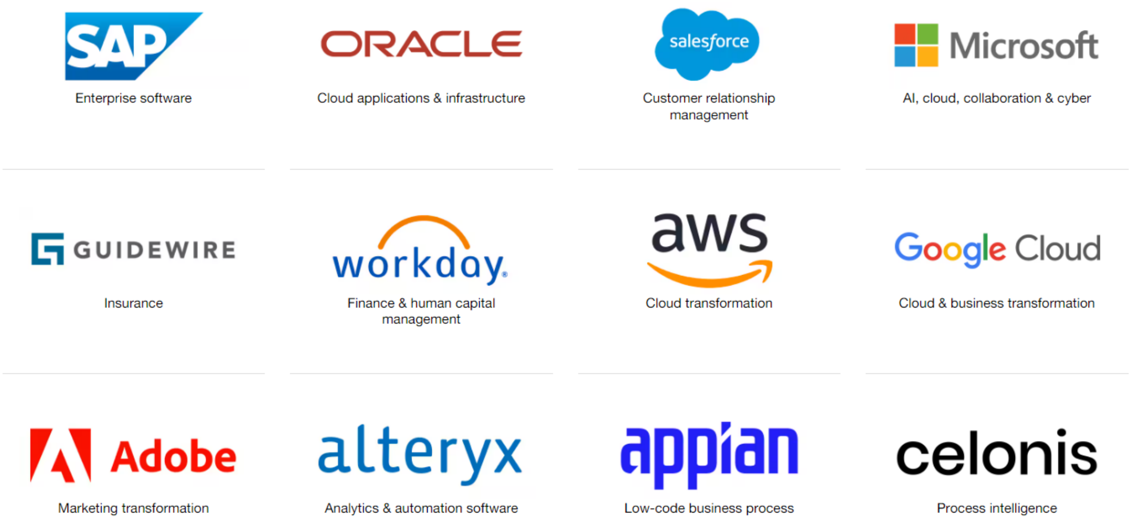

SAP
Enterprise software
ORACLE
Cloud applications & infrastructure
salesforce
Customer relationship management
Microsoft
AI, cloud, collaboration & cyber
GUIDEWIRE
Insurance
workday.
Finance & human capital management
aws
Cloud transformation
Google Cloud
Cloud & business transformation
Adobe
Marketing transformation
alteryx
Analytics & automation software
appian
Low-code business process
celonis
Process intelligence

Figure 1.2: Some of PwC's Alliances

1.4.2 PwC France and Maghreb

PwC operates 14 sites in France and three territories in the Maghreb, employing 6,750 associates and collaborators. The firm reports significant financial health with a revenue of 1 billion euros, highlighting its strong market presence and operational success in consulting.

1.4.3 PwC Algeria

In 2008, PwC established its Algerian entity, EURL PwC Algeria, aiming to expand its presence in the Mediterranean region. Part of the PwC France and Maghreb network, PwC Algeria employs over 100 staff at its Algiers office. Leveraging deep local economic insights, PwC Algeria provides comprehensive solutions to client challenges.

The firm has cultivated extensive expertise in the Algerian legal, tax, economic, and financial environment through its work with local companies and subsidiaries of foreign firms based in Algeria. Operating under two legal entities, PricewaterhouseCoopers Algeria and PASA Audit Services Algeria, they collaborate closely with other members within the PwC International Ltd network as part of the PwC France and Maghreb organization. This collaboration enables them to offer clients in Algeria the technical and sector-specific expertise of the entire network.

PwC contributes to the development of various sectors in Algeria, including agribusiness, manufacturing, financial services, oil & gas, iron & steel, and pharmaceuticals.

As of June 30, 2022, PwC Algeria boasts a dynamic team of 105 individuals, including 8 associates/directors, operating from a single office and achieving a substantial annual turnover of 775 million Algerian Dinars (DZD).

1.4.3.1 PwC's Global Strategy

In the strategic framework titled "The New Equation", PwC delineates its global strategy focused on two pivotal elements: "Building Trust" and "Sustained Outcomes." This strategy is augmented by a combination of human-centric approaches and technological prowess, ensuring the highest quality in all facets of their operations. Central to this strategy is PwC's "Community of Solvers," which underscores the firm's commitment to addressing complex industry challenges by uniting diverse expertise and viewpoints. This initiative highlights the firm's dedication to fostering trust and driving transformative outcomes, pivotal in supporting businesses globally.

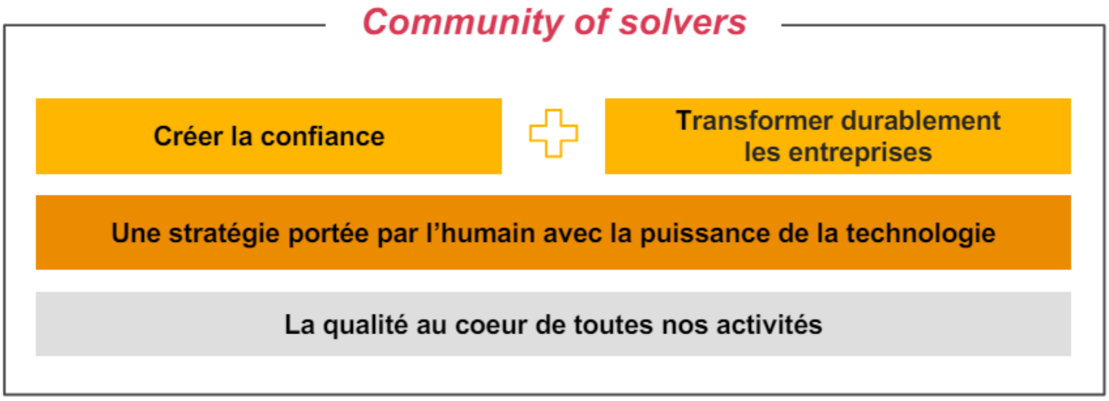

Community of solvers
Créer la confiance
Transformer durablement les entreprises
Une stratégie portée par l'humain avec la puissance de la technologie
La qualité au coeur de toutes nos activités

Figure 1.3: PwC's New Equation

1.4.3.2 PwC Algeria's values

- **Act with Integrity**
  - Speak out for just causes, especially when it seems difficult.

    - Aim and achieve excellence in results.
    - Make decisions and act as if one's own reputation was at stake.

- **Make a Difference**
    - Stay informed and question the future of the world we live in.
    - Have an impact on colleagues, clients, and society through our actions.
    - Adapt with agility to the environment in which we evolve.

- **Care**
    - Make an effort to understand each speaker and their priorities.
    - Recognize the value contributed by each.
    - Encourage everyone to progress and develop through their work and realize their potential.

- **Work Together**
    - Break down barriers to collaborate and share relationships, ideas, and knowledge.
    - Value the diversity of visions, ideas, and people.
    - Ask for and give feedback to improve and help others progress.

- **Reimagine the Possible**
    - Dare to challenge established situations and try new approaches.
    - Innovate, experiment, and learn from failures.
    - Explore the range of possibilities offered by each new idea.

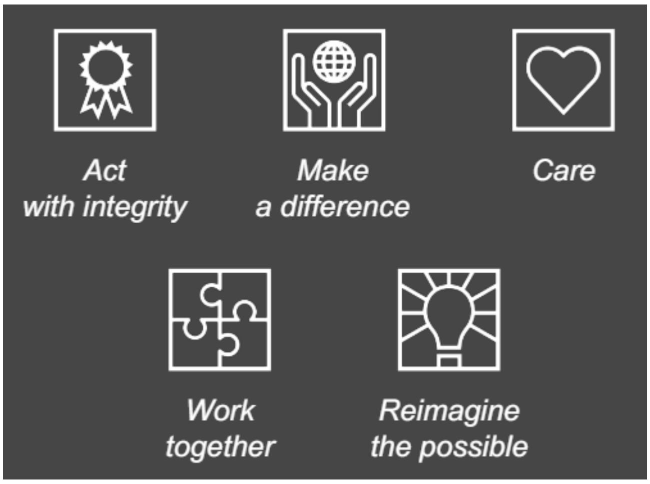


Figure 1.4: PwC's Values

1.4.3.3 Organizational Chart of PwC Algeria

The structure of PwC Algeria is distinguished by a certain autonomy granted to each team, while also fostering strong cooperation and exchange of expertise on typically multidisciplinary missions. This structure is illustrated in the figure below:

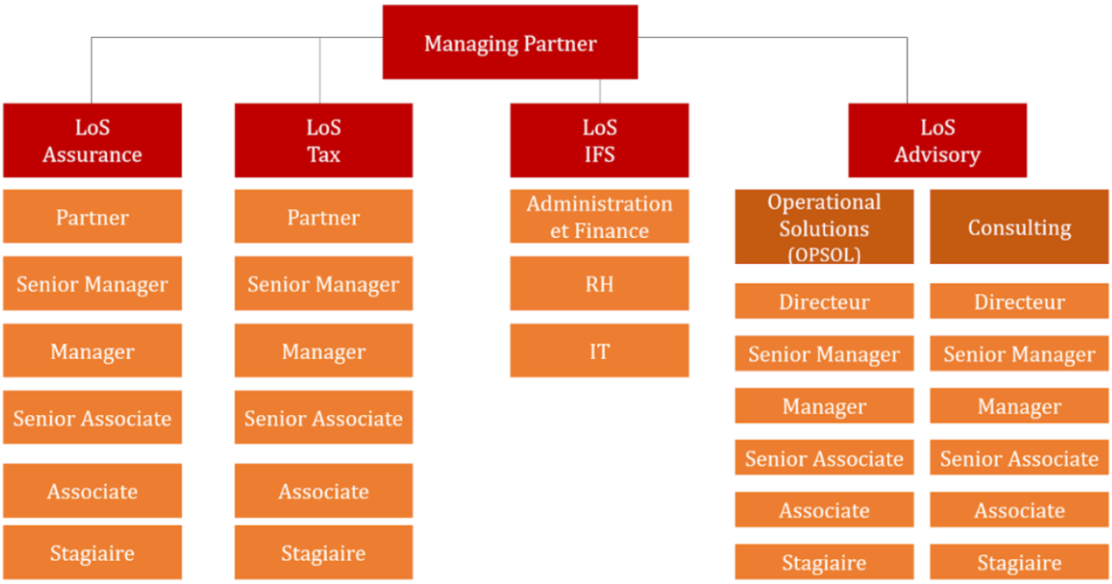


Figure 1.5: PwC's Organizational Chart

Through the organizational chart presented, it can be observed that the structure of PwC Algeria is composed of two main components:

- **Operational Component:** This includes the various departments that provide the services previously mentioned.
- **Administrative Wing (Support):** This encompasses various support functions necessary for managing the firm, such as HR, Accounting, IT, etc.

1.4.3.4 Service Areas of PwC Algeria

PwC Algeria offers a broad spectrum of services ranging from audit and strategic, management, transactional, legal, and tax consulting to a diverse clientele. These services cater to entities from small businesses to large multinationals, both in the public and private sectors, within Algeria and internationally. PwC Algeria specializes in the following areas:

- **Assurance:** Statutory audit services and risk management consulting.
- **Consulting:** Strategy, management, and operational solutions consulting.
- **Deals:** Support for companies in their acquisition, disposal, and restructuring missions.
- **TLS (Tax and Legal Services):** A multidisciplinary law firm offering services in taxation, business law, and labor law, integrating its expertise with other PwC sectors as needed.
- **Internal Functions:** Support for associates and employees in their daily tasks.

1.4.3.5 Consulting Department

By dedicating time to comprehend the specific challenges each client encounters, PwC's Consulting team applies both local and global insights. They aim to not only challenge conventional approaches but also to tailor and implement strategies that are uniquely effective for each situation. They combine expertise and industry insights to address client issues through Business Consulting. The areas of focus include:

- **Strategy & Execution:** Helping to develop or review corporate strategy, set the right strategic priorities for profitable growth, and support practical solutions into achieving these growth objectives. They also assist in strategy execution through performance management and establishing a Project Management Office focused on strategy execution.
- **Finance:** Supporting CEOs, CFOs, controllers, and treasurers to optimize the structure of finance functions to enhance their contribution to the business, addressing challenges of maintaining appropriate standards of control, and providing insight and challenge.
- **Operations:** Empowering operational leaders to become linchpins in achieving company profitability and growth goals, transcending daily business requirements with innovative thinking and agile responses to rapidly changing market conditions.
- **People & Change:** Providing expertise in change management challenges, including creating or modifying operating models, organizational restructuring, and defining clear roles and responsibilities.
- **Risk & Compliance:** Addressing the increasing complexity and variety of risks from new technologies, cyber threats, and regulatory changes. Strategies include building a risk-aware strategy, enhancing decision-making, and optimizing governance.
- **Technology:** Enabling business transformation through better utilization of technology investments, enhancing processes, IT strategy, architecture, and design, and managing enterprise applications and IT operations. This includes comprehensive SAP Consulting services to ensure optimal integration and performance of SAP solutions tailored to meet specific business needs.

SAP Consulting Overview

SAP Consulting services play an indispensable role in optimizing business operations and strategic decision-making through technological advancements. These services encompass a comprehensive suite of strategies to deploy, enhance, and maintain SAP systems effectively across various industries. SAP consultants specialize in customizing solutions to align with an organization's unique business processes, operational needs, and long-term goals. By integrating SAP's robust applications with enterprise systems, consultants ensure that businesses can leverage real-time data analytics, streamlined workflows, and enhanced productivity. Key areas of focus include SAP ERP implementations, system upgrades, migration to SAP S/4HANA, and application management services, all designed to drive significant improvements in business efficiency and intelligence.

PwC Algeria is the only Platinum partner in Algeria. This is the highest partnership level globally and partners can only benefit from it by invitation from SAP. The selection procedure takes into account the partner's ability to successfully carry out SAP transformation projects, an impeccable track record in customer service quality, and therefore the overall sustainability of the company's business model.

In the field of SAP services, PwC Algeria focuses its efforts on supporting its clients in the S/4HANA environment. It also supports the optimization of SAP architectures with solutions such as SAP Central Finance or procurement solutions like Ariba or SAP Central Procurement.

The company relies on the combination of business and technological expertise and carries out client projects from the strategic phase to implementation using a holistic approach. PwC employs over 11,000 staff worldwide focused on SAP services.

Figure 1.6: SAP Platinum Partner

1.5 Study Context

At PwC, one of the key objectives of our consulting department is to guide client firms towards implementing more sophisticated financial management strategies. In line with this mission, our current project involves a client—whose specific identity remains undisclosed due to confidentiality agreements. Our goal is to develop and deploy an advanced tool designed to enhance cash flow management for the company. This tool will enable the client to gain valuable insights into future cash flow projections, ultimately improving the firm's capacity for strategic financial decision-making.

To effectively enhance cash flow management for this company, I conducted an initial study focused on understanding the nuances of cash flow and cash flow management, including its critical components and the key factors that directly influence it like Accounts Payable and Accounts Receivable. The findings are comprehensively detailed in the two subsequent chapters, providing a thorough exploration of the strategies employed to optimize the client's financial operations.

Chapter 2

Introduction to Cash Flow and its Components

2.1 Introduction

To gain a deeper comprehension of the context surrounding our research and effectively address the central problem while achieving our outlined objectives, this initial chapter endeavors to introduce the critical concepts of cash flow and cash flow forecasting. It aims to lay a robust foundation for understanding the intricate dynamics and significance of cash management in businesses. By delving into the mechanisms and nuances of cash flow, including its sources, uses, and its components providing a detailed explanation of Accounts Receivable and Accounts Payable Management and their relationship with cash flow. We will also explore avenues for improvement and revisit the problem statement to better articulate the challenges and strategic responses involved.

2.2 Cash Flow

2.2.1 What is Cash Flow?

Cash flow is the net cash and cash equivalents transferred in and out of a company. Cash received represents inflows, while money spent represents outflows. A company creates value for shareholders through its ability to generate positive cash flows and maximize long-term free cash flow (FCF). FCF is the cash from normal business operations after subtracting any money spent on capital expenditures (CapEx) .

Businesses take in money from sales as revenues and spend money on expenses. They may also receive income from interest, investments, royalties, and licensing agreements and sell products on credit. Assessing cash flows is essential for evaluating a company's liquidity, flexibility, and overall financial performance .

Positive cash flow indicates that a company's liquid assets are increasing, enabling it to cover obligations, reinvest in its business, return money to shareholders, pay expenses, and provide a buffer against future financial challenges. Companies with strong financial flexibility fare better in a downturn by avoiding the costs of financial distress .

Cash flows are analyzed using the cash flow statement, it is one of many **financial statements** that provides aggregate data regarding all cash inflows that a company receives from its ongoing operations and external investment sources. It also includes all cash outflows that pay for business activities and investments during a given period. A company's financial statements offer investors and analysts a portrait of all the transactions that go through the business, where every transaction contributes to its success. The cash flow statement is believed to be the most intuitive of all the financial statements because it follows the cash made by the business in three main ways: through operations, investment, and financing. The sum of these three segments is called net cash flow [6].

2.2.2 Cash Flow Activities

2.2.2.1 Cash Flow from Operations

Cash flow from operations (CFO) corresponds to the cash inflows and outflows generated by the main activities of the company. It represents the cash movements related to sales, purchases of raw materials, salaries, social and tax charges, as well as all other elements that directly affect the operational result of the company.

- **Inflows from Operating Activities:**

  The main inflows from operating activities are the sales of goods and services. They can be collected immediately or be subject to payment terms, such as invoices payable in 30, 60, or 90 days. Therefore, it is important to closely monitor customer payment terms to anticipate cash inflows in the short and medium term and thus control the need for working capital (NWC).

- **Outflows from Operating Activities:**

  Outflows from operating activities correspond to expenses related to the main activity of the company. These may include the purchase of raw materials, supplies, employee remuneration, social and tax charges, etc. To control outflows from operating activities, it is essential to monitor costs, negotiate with suppliers, and optimize inventory management.

- **Key Performance Indicators of CFO:**

  Key performance indicators of CFO allow measuring the financial health of the company and its ability to generate positive cash flows from its main activity. The main indicators related to CFO to analyze are:

  - **Revenue** : It represents the total amount of sales made by the company over a given period.
  - **Average customer payment period** : It measures the time required for customers to pay their invoices. The longer the period, the higher the risk of unpaid invoices.
  - **Average supplier payment period** : It measures the time required for the company to pay its suppliers. The longer the period, the more surplus cash the company has.
  - **Operating profit** : It represents the difference between the operating revenue and expenses of the company.

2.2.2.2 Cash Flow from Investing Activities

Cash flow from investing activities (CFI) corresponds to the cash inflows and outflows related to the investments of the company. They concern mainly the acquisitions of long-term assets, such as machinery, equipment, buildings, or financial investments, such as stocks or bonds. CFI is therefore related to the long-term strategy of the company and has a significant impact on its cash flow.

- **Inflows from Investing Activities:**

  Inflows from investing activities correspond to the amounts received from the sale of long-term assets or the sale of subsidiaries. They can also come from the sale of financial securities. These cash inflows are generally infrequent but can have a significant impact on the company's cash flow.

- **Outflows from Investing Activities:**

  Outflows from investing activities are related to expenses related to the company's investments. They can be significant and significantly affect the company's cash flow. The main outflows from investing activities include the purchase of long-term assets such as machinery, equipment, buildings, or financial investments such as stocks or bonds.

- **Key Performance Indicators of CFI:**

  Key performance indicators of CFI allow measuring the impact of investments on the company's cash flow and its ability to generate positive cash flows from its investments. The main indicators related to CFI are:

  - **Investment cash flow** : It represents the cash flows generated by the company's investments. If the investment cash flow is positive, it means that the investments have generated cash inflows greater than outflows, and vice versa.
  - **Return on investment** : It measures the return on investments made by the company. It is the ratio between the net profit generated by the investments and their cost. A high ROI means that the investments have been profitable for the company.
  - **Payback period** : It measures the time required to recover the invested amount. The shorter this period, the more profitable the investment is considered.

2.2.2.3 Cash Flow from Financing Activities

Cash flow from financing activities (CFF) corresponds to the cash inflows and outflows related to the financing activities of the company. These include borrowing, debt repayments, issuances of stocks and bonds, as well as dividends paid to shareholders. CFF is therefore related to the company's financing choices and determines its financial structure.

- **Inflows from Financing Activities:**

  Inflows from financing activities correspond to the amounts received from borrowings, issuances of stocks and bonds, as well as the sale of debt securities. They can also come from capital contributions by shareholders.

- **Outflows from Financing Activities:**

  Outflows from financing activities correspond to debt repayments, share buybacks, and dividend payments to shareholders. They may also include issuance costs for stocks and bonds.

- **Key Performance Indicators of CFF:**

  Key performance indicators of CFF allow measuring the impact of financing choices on the company's cash flow and its ability to generate positive cash flows from its financing activities. The main indicators to monitor for CFF are:

  - **Financing cash flow cash flow** : It represents the cash flows generated by the company's financing activities. If the financing cash flow is positive, it means that financing activities have generated cash inflows greater than outflows, and vice versa.
  - **Debt ratio** : It measures the proportion of debt in the company's financial structure. The higher this ratio, the more indebted and vulnerable the company is to fluctuations in interest rates.
  - **Weighted average cost of capital (WACC)** : It measures the total cost of the company's financing, taking into account the cost of debt and equity capital. A high WACC indicates that the company has significant financing costs[26].

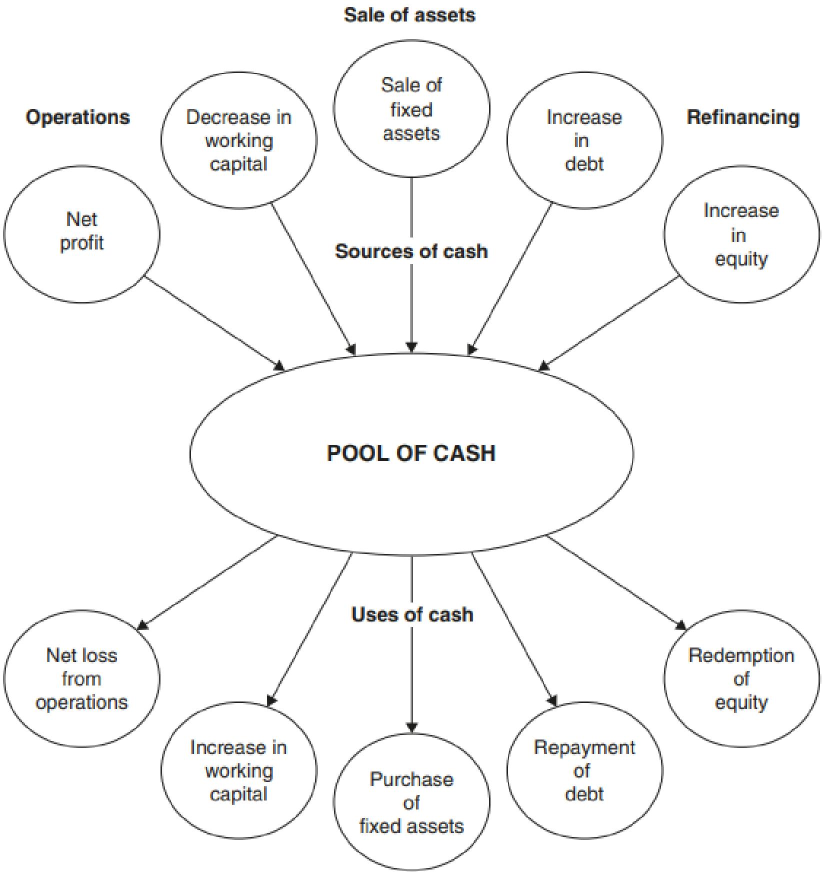


Figure 2.1: Sources and uses of cash (inflows and outflows) [26]

2.2.3 Cash flow forecasting

Cash flow forecasting is an indispensable tool in financial management that estimates a company's future cash positions and overall financial situation. Essential for both large corporations and smaller businesses, this forecasting process is based on anticipated payments, receivables, and a wide array of financial transactions. A comprehensive definition of cash flow forecasting involves predicting not only the operational inflows and outflows but also those related to financing and investing activities. This detailed forecasting includes:

- Estimating revenue from ongoing business operations.
- Anticipating expenses related to operational activities, like operating costs, employee salaries, and other fixed costs.
- Predicting cash inflows from financing activities, such as obtaining loans or issuing new shares.
- Forecasting cash outflows for financing activities, like debt repayment or dividend distribution.
- Estimating cash inflows from investing activities, such as the sale of assets.
- Anticipating cash outflows for investing activities, including purchases of assets or investments in other businesses.
- Analyzing these predictions helps a business assess if it will have sufficient funds to meet all its financial obligations and avoid insolvency, thus playing a critical role in the strategic financial planning and management of the company.

Regarding the methods used in cash flow forecasting, various approaches can be employed, such as the receipts and disbursements method, bank data approach, and statistical modeling approach.

2.3 Accounts Receivable Management

2.3.1 Importance of Accounts Receivable Management

In the dynamic landscape of modern business, managing the finances of a growing company presents substantial challenges, particularly in balancing liquidity with opportunities for growth. Liquidity, the ability of a company to meet its short-term obligations such as rent, salaries, and debts, is essential for operational stability. Businesses must maintain a ready supply of cash or liquid assets—those assets quickly convertible to cash—to meet both expected and unforeseen financial demands. However, excessively maintaining liquid assets may lead to suboptimal financial returns compared to investments in fixed assets like property and machinery, which are not readily liquid but are crucial for long-term growth. Thus, achieving an ideal balance between maintaining necessary liquid assets and maximizing investment becomes crucial.

Integral to this balance is the management of accounts receivables—the dues owed by customers for goods or services rendered in the ordinary course of business. These receivables, which represent future but uncertain cash flows, play a pivotal role in liquidity management. The practice of selling on credit is commonplace in modern commerce, necessitated by the competitive landscape where businesses extend trade credit to safeguard their market share and attract customers under favorable terms. However, while credit sales are vital for expanding business, they introduce risks associated with delayed payments and the potential depreciation of money over time. Given that receivables typically constitute a significant portion of current assets in many industrial sectors, their management is imperative for sustaining cash flow and liquidity.

Therefore, effective accounts receivable management and strategic cash flow forecasting are interdependent. By optimizing the management of credit terms and procedures, businesses can enhance their liquidity and ensure that sufficient funds are available to cover operational needs without compromising on growth investments.

Credit management is risky and it is known as riding on a double-edged sword. If credit is not given sales will not increase, which is allowed as a chance of bad debts. Hence, every firm has to be careful in credit sales and credit extension. As such a prudential financial manager has to be optimum in deciding the quantum of credit, standards and procedures as well as terms of credit. The impact of credit business on the wealth of the firm is shown in this figure [28]:

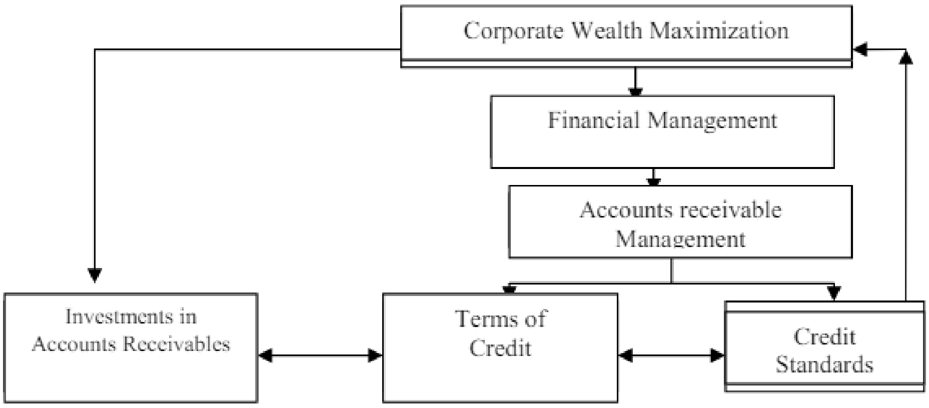

Corporate Wealth Maximization
Financial Management
Accounts receivable Management
Investments in Accounts Receivables
Terms of Credit
Credit Standards

Figure 2.2: Credit impact on wealth Maximization of shareholders [28]

2.3.2 Objectives of Receivables Management

The main aim of credit management is not to maximize the sales, nor to minimize risk of bad debts, but it is to manage its credit in such a way that sales are expanded to such an extent to which risk remains within an acceptable limit. In order to attain the maximize the value of the firm, it should manage its trade credit to:

- obtain the optimum volumes of sales for which the efficient and effective credit management helps the firm to retain the old customers and attract new customer.
- Control the cost of credit and keep it at a minimum. These costs are associated with trade credit in the form of administrative expenses, bad debts losses, and the opportunity cost of funds tied up in receivables.
- Maintain investment in debtors at an optimum level. By extending liberal credit, sales and profits increase, but increased investment in debtors also results in increased costs. Therefore, a trade-off between costs and benefits must be made.

2.3.3 Issues of Receivable Management

The management of receivables is a very critical area in the total working capital management as it can be very costly and time-consuming activity. The management of receivables can be divided into:

2.3.3.1 Credit policy

A credit policy outlines terms of credit, credit limits, and cash discounts. While not required to follow competitors' policies, understanding them helps determine an optimal strategy. A firm's credit policy impacts sales volume and profitability, making it essential for efficient cash flow, clear objectives, strong customer relations, and employee empowerment. The main goals of a firm's credit policy are to serve as a marketing tool to increase sales or retain customers in a competitive market, to maximize sales through a lenient credit policy while balancing costs, and to build long-term relationships and reward customer loyalty, often extending credit due to past practices or customer dependence. An optimum credit policy covers the following aspects:

- **Investment in receivables:** Financial manager has to offer certain sales on credit, which means the credit sales is financed by the firm. Firms if rich in cash, credit extension is desirable. If firms are not strong financially, finance has to be obtained from outside which means inviting interest burden that goes to reduce profitability of the firm. So, financial manager has to reduce the capital tied up on credit sales.
- **Terms of credit:** If credit terms are not competitive it will affect sales and consequently the shareholders' wealth. Here terms refers to what is the price if sold for cash, otherwise, what is the credit period and cash discount, how much percentage for how many days are the issues. Like wise the financial manager has to decide as and when situation arises
- **Credit standard:** Credit standards have a bearing on sales of the firm. These standards refer to minimum requirements for the evaluation of credit worthiness of a customer. The company may be liberal or strict in defining the requirement in getting credit. The standards imposed by the company are to assess the credit worthiness of customers. As long as company's profitability is higher, it can lower credit standards, which it would adversely, affect the sales.

2.3.3.2 Credit Analysis

After establishing the credit policy, the firm should conduct the credit analysis for evaluating the capabilities of the customers. The credit analysis would broadly divided into two steps, i.e., obtaining credit information, and analysis of credit information. It is on the basis of credit analysis that the decision to grant credit to a customer as well as the quantum of credit would be taken. The credit information may provide some insights about the creditworthiness of the customer with respect to the character, capacity, capital, condition, cost and collateral.

Besides establishing a credit policy, a firm should develop procedures for evaluating credit applicants. The first step in the credit analysis is obtaining the credit information. The sources of information broadly divided into internal and external. The internal source of information is derived from the records of the firms contemplating an extension of credit. On the other hand the information available from external sources are financial statements of the customer, bank references, trade references credit bureau reports, etc. In nutshell, the following are the various steps involved the credit analysis of the customers:

- Get the financial data and analyze them
- Obtain the bank and trade references
- Refer the past records of the business
- Take the opinion of sales personnel
- Get the credit assessment of the CRISIL, ICRA, etc.
- Ask customers to supply information substantiating his credit worthiness
- Determine the credit worthiness of the customers
- Take a decision to grant or not to grant credit to them
- Send goods on trial basis before establishing market relations

The company willing to grant credit would enquire about the 'prospective debtor' in the market and know about the inventions and plans from the speeches of the Chairman. With the help of the credit analysis, the customers are selected. The following are the 5 elements that go into credit analysis in identifying a sound customer:

Capital: The customer's repayment capacity largely depends on their capital adequacy. This can be assessed by checking various financial ratios, especially liquidity and turnover ratios, as well as conducting a funds flow analysis. These exercises help to reveal the customer's capital sufficiency and overall financial position.

Character: The character of a customer is crucial in determining their repayment capacity. It involves their cooperation and timeliness in settling debts. It is important to note that some firms may not cooperate despite having funds, while others might wish to pay promptly but are unable due to financial constraints.

Collateral: Collateral refers to the assets a customer offers as security. The amount of secured financing a customer has can indicate their creditworthiness. If a customer fails to meet their credit obligations, creditors can recover the owed amounts from the proceeds of these collateral assets.

Capacity: This refers to the personnel, technology, and entrepreneurial abilities of a firm. A firm's capacity to settle debts is a reflection of its recognition and standing within the market or industry.

Past Experience: In addition to the current financial and operational capabilities, it is crucial to consider a customer's history of payments. This includes checking old records and any previous legal issues or troubles caused to other businesses.

2.3.3.3 Collection Policy

The third area involved in the receivable management is collection policies. The firm should follow a well laid down collection policy and procedure to collect dues from its customers. The collection policies cover two aspects, i.e., the degree of effort to collect the over dues, and type of collection efforts. The collection policies may be classified into strict and liberal. The effects of tightening the collection policy would be to decline in sales, debts, collection period, interest costs and increase in collection costs and whereas, the effects of a lenient policy would be exactly the opposite.

Firms should be practical in their approach in collecting credit sales through regular correspondence, personal calls, telephone contacts, etc. The sudden reminders will not make the collection programs effective unless they take a follow up action and maintain personal relations. If the collection policy is not effective the company will incur large expenses and fail to be 'fund - rich'. So firms should trade –off between cost of collection and bad debt losses. A stitch in right time will save lot. The following diagram shows the relationship between losses due to bad debts and collection expenses [28].

2.3.4 Credit Terms

Another noteworthy aspect of accounts receivables management is deciding credit terms, which include:

- **Credit period:** It is the period allowed by the seller to the customer to pay the bills. The customer can take advantage and pay conveniently his bills. Here the customers are interested in getting more credit period. But the firm has to decide optimally the period even if sales increases proportionately, the relaxation would cost nearly the firm, as the funds will be blocked.
- **Credit discount:** Cash discounts are in the form of discount rate and discount period. It is an incentive to the customer who pays early which many customers take advantage of cash discounts. Rarely some firms do not utilize the opportunity due to their funds tied up and not able to take advantage of as they have no cash balance. Of course this policy would result in loss of revenue of it. Therefore the management has to balance the benefits and costs before arriving a decision on cash discount.
- **Credit limit:** Credit sales decision cannot easily be made. While taking credit decision, besides character and capacity of the customer, the supplier has to decide several other things such as extent of credit and credit period. Some times, the supplier is asked to extend credit amount or credit period, for which the customer will not oblige either extra price or interest rate. Under such circumstances, the supplier has to weigh the profit out of extra sales against costs on account of such deal. As long as he makes reasonable gain in the deal he will say yes to extend extra credit period or credit extension.

2.3.5 Accounts Receivables Collection Management

The importance of improving the Order-to-Cash process (O2C) is argued to be decisive in order also to improve cash flow management. The O2C process is characterized by all the actions on the company level ranging from when the potential customer is checked for credit until the payment is received. The account receivables management can be located in a sub-process inside the O2C, Invoice-to-Cash (I2C). This process will only start once the invoice is created and sent to the customer and its duration depend on the collection effectiveness.

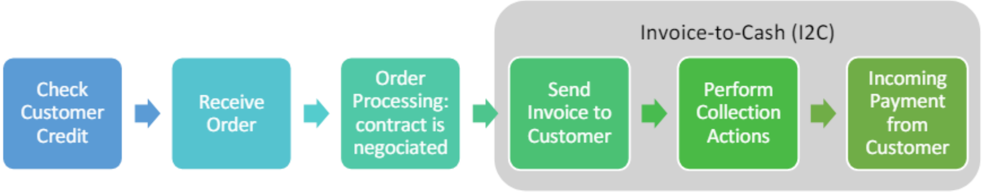

Invoice-to-Cash (I2C)
Check Customer Credit
Receive Order
Order Processing: contract is negociated
Send Invoice to Customer
Perform Collection Actions
Incoming Payment from Customer

Figure 2.3: The Order-to-Cash (O2C) and Invoice-to-Cash (I2C) processes

Before moving on, some critical concepts in collection management are reviewed: in a transaction contract, **the payment term** is simply the negotiated number of days for the invoice to be paid (from invoice created date to due date), usually 30, 45 or 60 days, and **the due date** is the maximum date when the invoice should be paid. For instance, if an invoice is created on the 1st of June and the payment term is 30 days, then the due date is July 1st; **an overdue (or late) invoice** is a bill that is late in their payment, i.e., the invoice's due date has passed, and the customer has not yet made the payment. In contrast, **outstanding invoices** are not paid yet but are not late either, i.e., their due dates lie in the future. Invoices can also be paid in advance – in other words earlier than the agreed due date. These advance payments usually attribute more risk to the customer who can pay for the service or good but not receive it at the end. Some choose to pay in advance for several reasons, namely, to have a good reputation and better relationship with the company.

In a typical company, collection management is usually the responsibility of cash collectors. They monitor company's account receivables, typically with manual steps and techniques that do not guarantee efficiency and are prone to error. They tend to overlook smaller customers favoring larger customers or more risky payments, and prioritize invoices with larger values. They take interventive actions which habitually will be corrective rather than preventive due to time scarcity and lack of better systems. Different actions are taken based on the severity of the lateness of the payment. Soft actions like notices of late invoices by email and calls are usually done on the first days of the late invoice. More severe actions like letters of demand are sent a couple of months after the invoice due date has passed. In very severe cases, the debt can be considered lost, and the company will ensure that no more transactions with the same customer are made, sometimes even proceeding to legal action. The timeline of these actions varies by company, industry and even sometimes customers. Even though less common in most companies, some preventive actions are also taken, such as offering discounts to accelerate payment and negotiating penalty fees for the late payments when the contract is stipulated.

An important tool in collection management are the A/R Aging reports. They are structured tables where the collectors can periodically monitor account receivables; they are useful to keep track of delinquent invoices for an extended period. Usually, ageing buckets are created, for instance, outstanding (not late), 0-30 days overdue, 31-60 days overdue, 61-90 days overdue, 90 + days overdue [29].

2.4 Accounts Payable Management

2.4.1 The significance of Payables

Payables constitute a current or short term liability representing the buyer's obligation to pay a certain amount on a date in the near future for value of goods or services received. They are short term deferments of cash payments that the buyer of goods and services is allowed by the seller. Trade credit is extended in connection with goods purchased for resale or for processing and resale, and hence excludes consumer credit provided to individuals for purchasing goods for ultimate use and instalment credit provided for purchase of equipment for production purposes. Trade credits or payables serve as non-interest bearing source of funds in most cases. They provide a spontaneous source of capital that flows in naturally in the course of business in keeping with established commercial practices or formal understandings.

2.4.2 Types of Payables

Trade Credits or Payables could be of three types: Open Accounts, Promissory Notes and Bills Payable.

- Open Account or open credit operates as an informal arrangement wherein the supplier, after satisfying himself about the credit-worthiness of the buyer, despatches the goods as required by the buyer and sends the invoice with particulars of quantity despatched, the rate and total price payable and the payment terms. The buyer records his liability to the supplier in his books of accounts and this is shown as payables on open account. The buyer is then expected to meet his obligation on the due date.
- The Promissory note is a formal document signed by the buyer promising to pay the amount to the seller at a fixed or determinable future time. Where the client fails to meet his obligation as per open credit on the due date, the supplier may require a formal acknowledgement of debt and a commitment of payment by a fixed date. The promissory note is thus an instrument of acknowledgement of debt and a promise to pay. The supplier may even stipulate an interest payment for the delay involved in payment.
- Bills Payable or Commercial Drafts are instruments drawn by the seller and accepted by the buyer for payment on the expiry of the specified duration. The bill or draft will indicate the banker to whom the amount is to be paid on the due date, and the goods will be delivered to the buyer against acceptance of the bill. The seller may either retain the bill and present it for payment on the due date or may raise funds immediately thereon by discounting it with the banker. The buyer will then pay the amount of the bill to the banker on the due date.

2.4.3 Determinants of Trade Credit

- **Size of the Firm**: Smaller firms have an increasing dependence on trade credit as they find it difficult to obtain alternative sources of finance compared to medium or large-sized firms. Larger firms, being less vulnerable to adverse business conditions, can command prompt credit facilities from suppliers. In contrast, smaller firms may struggle to maintain creditworthiness during financial strains and may have reduced access to credit due to a weaker financial position.

- **Industrial Categories**: Different industries or commercial enterprises show varying degrees of dependence on trade credit. In some businesses, prevailing commercial practices may require purchases against payment. Monopoly firms may insist on cash on delivery. In instances where a firm's inventory turns over every fortnight but enjoys thirty days of credit from suppliers, trade credit can finance the inventory and provide additional working capital.

- **Nature of Product**: Products with faster sales or higher turnover may need shorter-term credit, while products with slower turnover, taking longer to generate cash flows, will need extended credit terms.

- **Financial Position of Seller**: The seller's financial position influences the quantities and period of credit extended. Financially weak suppliers need to be strict and operate on higher credit terms. Financially stronger suppliers can dictate stringent credit terms but may prefer to extend liberal credit if the benefits exceed the costs. They can afford to extend credit to smaller firms and assume higher risks. Suppliers with working capital constraints may offer higher cash discounts to encourage early payments.

- **Financial Position of the Buyer**: The buyer's creditworthiness is crucial in determining the credit quantum and period. Large buyers may not insist on extended credit terms from small suppliers with weak bargaining power. When goods are supplied on a consignment basis, the supplier provides extra finance for the merchandise and pays a commission to the consignee for the goods sold, enabling small retailers to carry larger stock levels than they could finance themselves. Slow-paying or delinquent accounts may face stricter credit terms or higher product prices to cover the risk [25].

2.4.4 Effective Management of Payables

The salient points to be noted on effective management of payables are:

- Negotiate and obtain the most favourable credit terms consistent with the prevailing commercial practice pertaining to the concerned product line.

- Where cash discount is offered for prompt payment, take advantage of the offer and derive the savings therefrom.

- Where cash discount is not provided, settle the payable on its date of maturity and not earlier. It pays to avail the full credit term.

- Do not stretch payables beyond the due date, except in inescapable situations, as such delays in meeting obligations have adverse effects on the buyer's credibility and may result in more stringent credit terms, denial of credit, or higher prices on goods and services procured.

- Sustain a healthy financial status and a good track record of past dealings with the supplier to maintain their confidence. The quantum and terms of credit are mainly influenced by the supplier's assessment of the buyer's financial health and ability to meet maturing obligations promptly.

2.5 Critiques of Actual Accounts Receivable and Payable Management

In AR, the collection processes often face delays due to inefficient policies, adversely affecting a company's liquidity. Cash collectors may inadvertently prioritize larger or more risky payments, neglecting smaller customers and favoring invoices with larger values. This approach tends to lead to corrective actions rather than preventive due to a lack of advanced systems and time constraints. Such inefficiencies can strain a company's cash reserves, forcing them to operate reactively rather than proactively.

Similarly, in AP management, improper scheduling of payments can lead to missed opportunities for early payment discounts or, conversely, incur late payment penalties. Both scenarios negatively impact cash flow, reflecting poor financial planning and execution which can also affect a company's ability to invest in growth or maintain operational stability.

Moreover, the management approaches in both AR and AP can significantly impact business relationships. Aggressive collection practices or inconsistent communication in AR can strain relationships with customers, potentially leading to loss of business and damage to reputation. In the realm of AP, delays in resolving disputes or processing payments can similarly strain relationships with suppliers, jeopardizing the reliability of the supply chain and negatively impacting terms of trade.

Another critical issue is the lack of predictive insights in traditional AR and AP methods. Without the ability to forecast and anticipate future challenges accurately, companies are often stuck with reactive measures. This shortfall not only leads to missed opportunities but also escalates operational costs, as businesses are unable to strategically plan their financial or operational moves ahead of time.

2.6 Improvements for Enhanced Cashflow Management

Incorporating predictive analytics and forecasting into accounts receivable (AR) and accounts payable (AP) management significantly enhances the strategic management of cash flow within these key financial functions. For AR, adopting predictive modeling to enhance invoice collection processes can accurately forecast payment timings and identify potential delays. This capability allows companies to take proactive measures to mitigate delays, improving the efficiency of collections and directly enhancing overall cash flow management.

In terms of accounts payable, utilizing predictive forecasts to anticipate future liabilities and their timing, based on historical trends, facilitates more strategic cash flow planning. This approach ensures that sufficient funds are allocated to cover upcoming expenses, effectively preventing cash shortages that could disrupt operations. By proactively managing these financial commitments, businesses not only maintain operational stability but also optimize their capacity for strategic investments, aligning expenditures with available financial resources for improved financial health.

2.7 Problem Statement

After a thorough examination of the current cash flow management processes, focusing particularly on operating cash flow—which includes accounts payable and accounts receivable management—it has become evident that significant enhancements are necessary. Both areas are crucial for maintaining liquidity and funding ongoing operations but often lack the precision and foresight provided by modern analytical tools. The introduction of predictive analytical modeling could revolutionize these processes by providing insights into future financial states, thereby improving decision-making and operational efficiency.

Despite the fundamental role that accounts receivable and accounts payable play in the management of operating cash flow, our company struggles to predict future financial outcomes accurately. This challenge is primarily due to traditional management practices that do not leverage advanced predictive technologies. These limitations can lead to suboptimal financial planning, missed opportunities for growth, and inefficient resource allocation. Integrating advanced statistical modeling and machine learning techniques presents a promising solution that could dramatically enhance our predictive capabilities, enabling more strategic decision-making and efficient financial management.

So, how can statistical methods and machine learning be effectively integrated into the components of cash flow management, such as accounts receivable and accounts payable, to enhance predictive accuracy and improve overall financial operations?

In the following two chapters, we will explore the various methods that will be employed to address this problem and achieve the objectives of the study.

Chapter 3

Techniques and Approaches in Machine Learning

3.1 Introduction

In recent years, Machine Learning (ML) has significantly impacted various sectors, including finance. It has enabled the automation of complex decision-making processes, risk assessment, customer service enhancements, and notably, accurate financial predictions such as cash flow estimation. Machine learning's ability to handle vast volumes of data and uncover patterns makes it invaluable for predicting financial trends and making informed decisions. This chapter delves into the foundational concepts and diverse techniques of machine learning.

3.2 Machine Learning

3.2.1 Definition

Machine learning, is a subfield of Artificial Intelligence that utilizes statistical techniques to create computer models capable of learning from data. Its aim is to model and understand complex structures or phenomena using data, which can represent a range of elements from a concept to a specific attribute or outcome. The process of machine learning seeks to simulate human learning by using mathematical optimization algorithms that progressively improve as they are exposed to more data. These algorithms learn, much like humans, to make predictions or decisions without being explicitly programmed to perform the task. The primary objective of machine learning algorithms is to estimate a function f that minimizes the difference between predicted values and values actually observed in the data. This function can take many forms, depending on the type of machine learning algorithm used and the nature of the data.

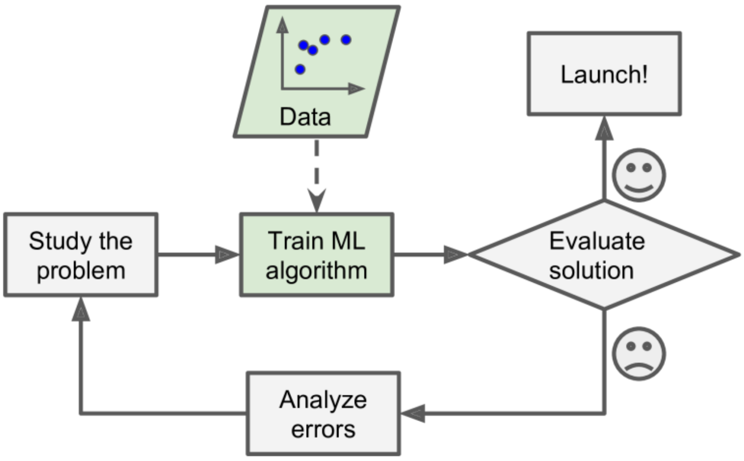


Figure 3.1: Machine learning approach [?]

3.2.2 End-to-End Machine Learning Project

In any concrete machine learning project, there are essential steps that must be followed to ensure success and relevance of the results. These steps, when executed well, enable the successful completion of an end-to-end machine learning project. Whether it's solving complex problems, optimizing business processes, or making data-driven decisions, the following steps form the foundation of any machine learning project

- **Understanding the Business:** At the beginning of the project, it is crucial to thoroughly understand the business objectives and underlying business stakes. This step helps to clearly define what is aimed to be achieved through machine learning. It is also important to formulate a precise problem statement related to data exploration in order to meet the specific needs of the business. Simultaneously, a preliminary plan is established to guide the subsequent steps of the project.

- **Understanding the Data:** This phase is dedicated to collecting data from various relevant sources. It is essential to deeply explore these data to understand their structure, format, and the information they contain. This step also includes describing the data by identifying key features and assessing their quality. Verifying the quality of the data ensures their reliability and suitability for building a machine learning model.

- **Preparing the Data:** Data preparation is a crucial step in which the collected data are prepared for use in the final model. This includes cleaning the data, where missing or outlier values are appropriately addressed. Additionally, creating new features from existing data can be performed to enhance the model's performance. The transformation of data, such as normalization or encoding, is also carried out to make them suitable for machine learning.

- **Modeling:** In this phase, several machine learning models are developed and tested to find the one that best meets the project's objectives. Different techniques and algorithms are explored, adjusting the parameters of each model to improve their performance. This stage often involves an iterative process of building, evaluating, and tuning the models to achieve the most optimal results.

- **Evaluation:** Once the models have been trained and tested, it is crucial to evaluate them thoroughly. The evaluation is not limited to technical metrics but must also consider the business objectives of the project. It is important to analyze how the models practically translate in the business context, assessing their relevance, reliability, and ability to address the identified issues.

- **Deployment:** The creation of the model does not mark the end of the project, as it needs to be deployed to be used by end-users. This step involves setting up appropriate infrastructure to host the model, as well as integrating it into existing systems. It is also crucial to provide a user-friendly interface allowing users to easily interact with the model and fully leverage its capabilities, while ensuring its regular maintenance and updates.

3.2.3 Types of Machine Learning Systems

There are so many different types of Machine Learning systems that it is useful to classify them in broad categories based on:

- Whether or not they are trained with human supervision (supervised, unsuper-vised, semi supervised, and Reinforcement Learning)
- Whether or not they can learn incrementally on the fly (online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do (instance-based versus model-based learning) [?].

3.2.3.1 Supervised Learning

Supervised Learning is a machine learning paradigm for acquiring the input-output relationship information of a system based on a given set of paired input-output training samples. As the output is regarded as the label of the input data or the supervision, an input-output training sample is also called labeled training data, or supervised data.

The goal of supervised learning is to build an artificial system that can learn the mapping between the input and the output, and can predict the output of the system given new inputs. If the output takes a finite set of discrete values that indicate the class labels of the input, the learned mapping leads to the classification of the input data. If the output takes continuous values, it leads to a regression of the input.

The input-output relationship information is frequently represented with learning-model parameters. When these parameters are not directly available from training samples, a learning system needs to go through an estimation process to obtain these parameters. So, the training data for Supervised Learning need supervised or labeled information.

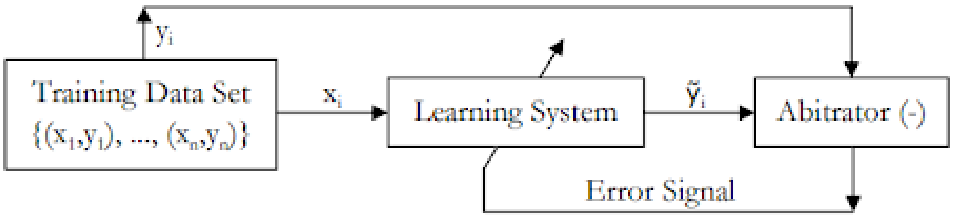


Figure 3.2: Block diagram that illustrates the form of Supervised Learning [14]

This figure shows a block diagram that illustrates the form of Supervised Learning. In this diagram, (x_i, y_i) is a supervised training sample, where 'x' represents system input, 'y' represents the system output (i.e., the supervision or labeling of the input x), and 'i' is the index of the training sample. During a Supervised Learning process, a training input x_i is fed to the Learning System, and the Learning System generates an output $\tilde{y}_i$. The Learning System output $\tilde{y}_i$ is then compared with the ground truth labeling y_i by an arbitrator that computes the difference between them. The difference, termed Error Signal in this diagram, is then sent to the Learning System for adjusting the parameters of the learner. The goal of this learning process is to obtain a set of optimal Learning System parameters that can minimize the differences between $\tilde{y}_i$ and y_i for all i, i.e., minimizing the total error over the entire training data set[12].

- **Regression:**

  Regression is a task aimed at predicting a numerical target value: the endogenous (dependent or output) variable, y, thus constitutes a vector containing numerical values. The learned model enables the prediction of a y_i for a new input X. An example would be predicting the price of a car based on a set of features (mileage, age, brand, etc.) known as predictors. To train the system, numerous examples of cars should be provided, including both their predictors and their labels (prices)[?].

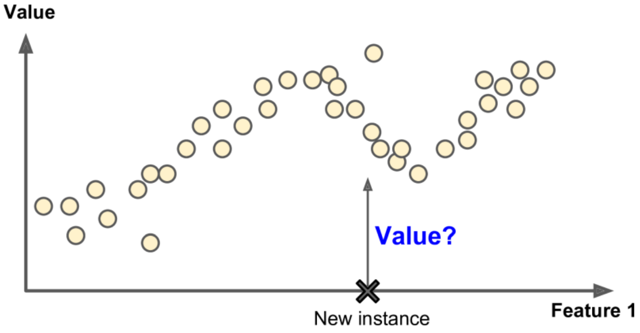


Figure 3.3: Regression [?]

- **Classification:**

  In this type of problem, the objective is to build a model capable of assigning each data input X, drawn from the set of input $\mathcal{X}$, to a class within a group of classes defined by Ω. Here, the variable Y is categorical (either nominal or ordinal) and takes its values in the set Ω. The spam filter is a good example of this: it is trained with numerous examples of emails accompanied by their class (spam or non-spam), and it must learn how to classify new emails. In this case, the classification is binary, as Ω consists of two possible outcomes: $\{\text{spam} = 1, \text{non-spam} = 0\}$. It is also possible to encounter problems requiring multi-class classification [?].

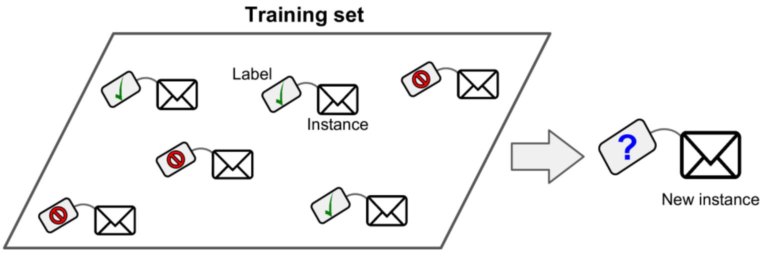


Figure 3.4: Spam classification [?]

3.2.3.2 Unsupervised Learning

Unsupervised learning, a category within machine learning, is dedicated to uncovering underlying patterns within data without pre-existing labels and with minimal human intervention. Unlike supervised learning, which relies on labeled data provided by humans, unsupervised learning operates under the premise of self-organization, aiming to model probability densities across inputs. Within unsupervised learning, four prominent methods are commonly employed: Clustering, Dimensionality Reduction, Association Rules, and Anomaly Detection.

Here is a table summarizing these different types of unsupervised learning:

Types	**Clustering**	**Dimensionality Reduction**	**Association Rules**	**Anomaly Detection**
Principe	The principle is to group a set of objects (data points) in such a way that objects in the same group, called a cluster, are more similar to each other than to those in other clusters. This similarity is typically based on features like distance, density, or connectivity.	The core principle is to simplify the dataset by preserving only the most important variables or features. This is achieved by identifying and retaining the components that capture the most variance or information in the data, thus transforming the data into a lower-dimensional space.	They are based on discovering interesting relationships between variables in large datasets, particularly in transactional databases. The fundamental principle is to find patterns of items or features that frequently occur together in these datasets.	The principle is to identify data points, events, or observations that deviate significantly from the majority of the data. The core idea is to detect outliers or anomalies that are not consistent with the normal behavior or expected pattern in a dataset.
Algorithms	K-means, Hierarchical Clustering, DBSCAN, Mean Shift Clustering, Spectral Clustering, OPTICS.	Principal Component Analysis (PCA), Singular Value Decomposition (SVD), t-Distributed Stochastic Neighbor Embedding (t-SNE), Linear Discriminant Analysis (LDA), Autoencoders.	Apriori, Eclat, FP-Growth (Frequent Pattern Growth).	Isolation Forest, One-Class SVM (Support Vector Machine), Autoencoders.
Application examples	market segmentation, social network analysis, image segmentation.	image and speech recognition, reducing noise in data, visualization of high-dimensional data, data compression.	market basket analysis to understand products frequently bought together, recommendation systems, fraud detection.	fraud detection, intrusion detection in network security, fault detection in mechanical systems.

Table 3.1: Unsupervised Learning Types

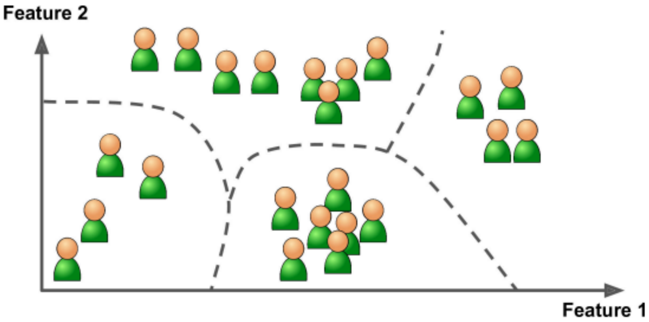


Figure 3.5: Clustering [?]

3.2.3.3 Semi-supervised learning

Semi-supervised learning is a type of machine learning where a model is trained on a dataset that contains both labeled and unlabeled data. Unlike supervised learning, which relies solely on labeled data, and unsupervised learning, which operates exclusively with unlabeled data, semi-supervised learning leverages the combined information from both labeled and unlabeled data to improve model performance. This approach is particularly useful in scenarios where obtaining labeled data is expensive or time-consuming, as it allows the model to learn from a larger pool of data while still benefiting from the supervision provided by labeled examples. Semi-supervised learning algorithms aim to effectively utilize the unlabeled data to enhance the model's ability to generalize and make accurate predictions on new, unseen data points.

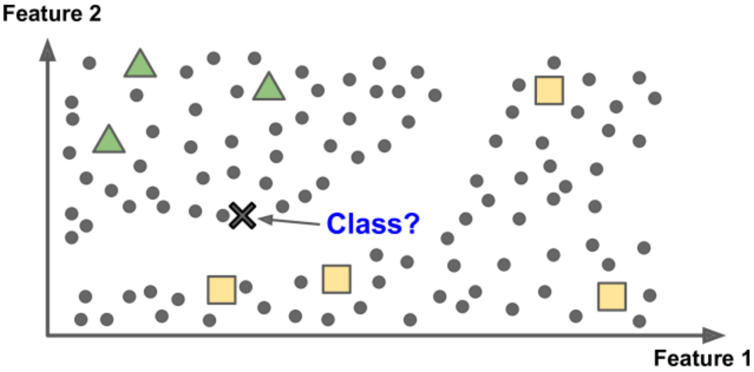


Figure 3.6: Semisupervised learning [?]

Most semi supervised learning algorithms are combinations of unsupervised and supervised algorithms. For example, deep belief networks (DBNs) are based on unsu- pervised components called restricted Boltzmann machines (RBMs) stacked on top of one another. RBMs are trained sequentially in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques.

3.2.3.4 Reinforcement learning

Reinforcement Learning is a very different beast. The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return (or penalties in the form of negative rewards). It must then learn by itself what is the best strategy, called a policy, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.

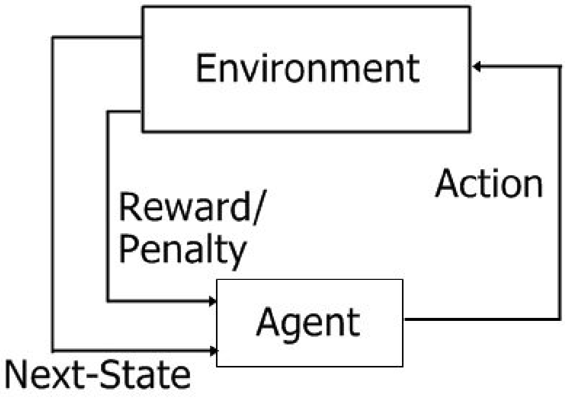


Figure 3.7: Reinforcement learning

3.3 Deep Learning

3.3.1 Definition

Deep learning is a set of learning methods attempting to model data with complex architectures combining different non-linear transformations. The elementary bricks of deep learning are the neural networks, that are combined to form the deep neural networks. These techniques have enabled significant progress in the fields of sound and image processing, including facial recognition, speech recognition, computer vision, automated language processing, text classification, time series forecasting..

There exist several types of architectures for neural networks :

- The multilayer perceptrons, that are the oldest and simplest ones
- The Convolutional Neural Networks (CNN), particularly adapted for image processing
- The recurrent neural networks, used for sequential data such as text or times series.

They are based on deep cascade of layers. They need clever stochastic optimization algorithms, and initialization, and also a clever choice of the structure. They lead to very impressive results, although very few theoretical fondations are available till now.

3.3.2 Neural networks

An artificial neural network is an application, nonlinear with respect to its parameters θ, that associates an input x with an output $y = f(x, \theta)$. For simplicity, we assume that y is unidimensional, although it could also be multidimensional. This application f has a particular form that will be specified later.

Neural networks can be used for both regression and classification tasks. As usual in statistical learning, the parameters θ are estimated from a learning sample. The function to minimize is not convex, leading to local minimizers. The success of this method stems from a universal approximation theorem, attributed to Cybenko (1989) and Hornik (1991). Furthermore, Le Cun (1986) proposed an efficient method to compute the gradient of a neural network, known as *backpropagation of the gradient*, that allows obtaining a local minimizer of the quadratic criterion easily.

3.3.2.1 Artificial Neuron

An artificial neuron is a function f_j of the input $\mathbf{x} = (x_1, \dots, x_d)$, weighted by a vector of connection weights $\mathbf{w}_j = (w_{j,1}, \dots, w_{j,d})$, completed by a neuron bias b_j, and associated with an activation function ϕ. The output y_j of the neuron can be described by the following equation:

$$y_j = f_j(\mathbf{x}) = \phi\left(\sum_{i=1}^{d} w_{j,i}x_i + b_j\right). \tag{3.1}$$

Several activation functions can be considered, such as the sigmoid, hyperbolic tangent, or ReLU functions, each serving different properties and uses in neural network architectures.

- **Identity Function:**

$$\phi(x) = x. \tag{3.2}$$

- **Sigmoid Function (or Logistic):**

$$\phi(x) = \frac{1}{1+\exp(-x)}. \tag{3.3}$$

- **Hyperbolic Tangent Function (tanh):**

$$\phi(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} = \frac{\exp(2x) - 1}{\exp(2x) + 1}. \tag{3.4}$$

- **Hard Threshold Function:**

$$\phi_\beta(x) = \begin{cases} 1 & \text{if } x \geq \beta, \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

- **Rectified Linear Unit (ReLU):**

$$\phi(x) = \max(0, x). \tag{3.6}$$

Here is the schematic representation of an artificial neuron where Σ represents the sum of the weighted inputs and the bias term:

$$y = \phi\left(\sum w_j x_i + b\right)$$

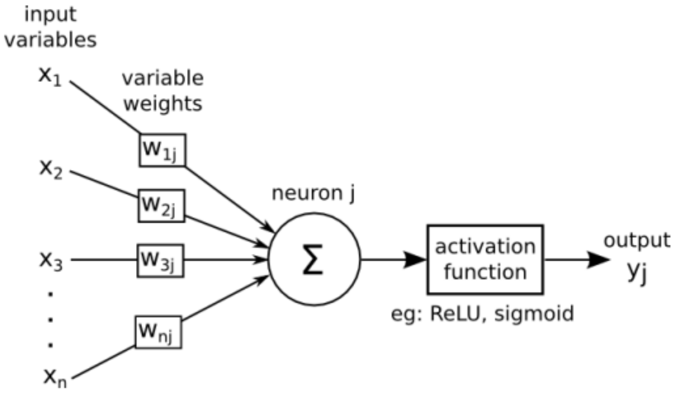


Figure 3.8: Schematic representation of an artificial neuron [13]

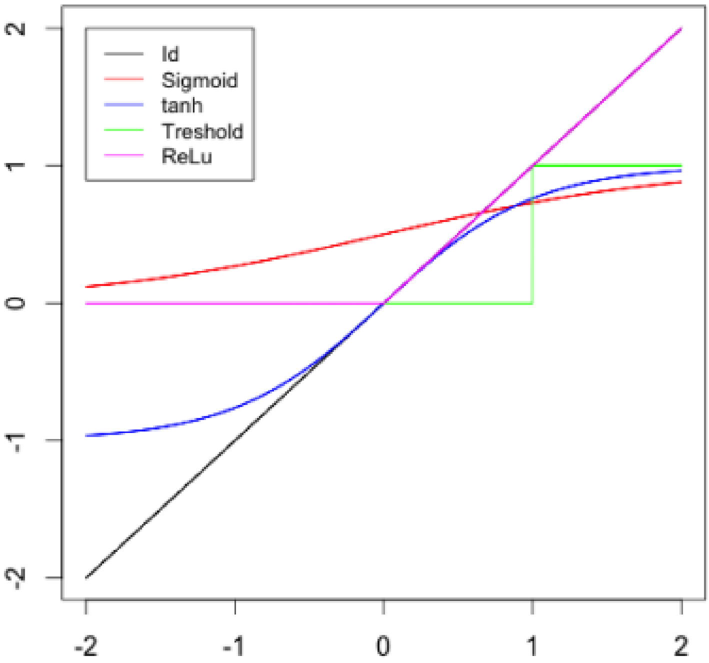


Figure 3.9: Activation functions [11]

Historically, the sigmoid was the mostly used activation function since it is differentiable and allows to keep values in the interval $[0, 1]$. Nevertheless, it is problematic since its gradient is very close to 0 when $|x|$ is not close to 0.

With neural networks with a high number of layers (which is the case for deep learning), this causes troubles for the backpropagation algorithm to estimate the parameter . This is why the sigmoid function was supplanted by the rectified linear function. This function is not differentiable in 0 but in practice this is not really a problem since the probability to have an entry equal to 0 is generally null [13].

3.3.2.2 Multilayer perceptron

A multilayer perceptron (or neural network) is a structure composed by several hidden layers of neurons where the output of a neuron of a layer becomes the input of a neuron of the next layer. Moreover, the output of a neuron can also be the input of a neuron of the same layer or of neuron of previous layers (this is the case for recurrent neural networks). On last layer, called output layer, we may apply a different activation function as for the hidden layers depending on the type of problems we have at hand : regression or classification.

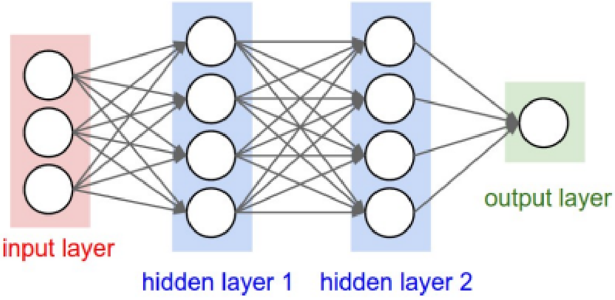


Figure 3.10: A basic neural network [13]

Multilayers perceptrons have a basic architecture since each unit (or neuron) of a layer is linked to all the units of the next layer but has no link with the neurons of the same layer. The parameters of the architecture are the number of hidden layers and of neurons in each layer. The activation functions are also to choose by the user. For the output layer, the activation function is generally different from the one used on the hidden layers. In the case of regression, we apply no activation function on the output layer.

3.3.2.3 Estimation of the parameters

Once the architecture of the network has been chosen, the parameters (the weights w_j and biases b_j) have to be estimated from a learning sample. These parameters are fundamental as they determine how well the network learns from the training data and performs on unseen data.

- **Parameter Initialization** Before the actual training begins, the parameters of the neural network are typically initialized. Initial values might be set to small random numbers, as starting with zero or the same values can lead to symmetrical problems where neurons learn the same features during training. More sophisticated methods like He initialization or Glorot initialization are often used depending on the activation functions in the network.

- **Loss Function Selection** The estimation process is primarily driven by the selection of an appropriate loss function. The loss function quantifies the difference between the predicted outputs of the neural network and the actual data. Common choices for loss functions include Mean Squared Error (MSE) for regression tasks and Cross-Entropy Loss for classification tasks. The choice of loss function significantly impacts the training process and the network's ability to generalize from training data to unseen data.

- **Gradient Descent Optimization** The core of the training process is the optimization algorithm used to minimize the loss function. Gradient descent is the most commonly used approach, where the gradient (or approximate gradient) of the loss function with respect to the model parameters is computed to make iterative adjustments to the weights and biases.

- **Backpropagation** Backpropagation is employed to compute the gradient of the loss function efficiently. This method involves a forward pass where input data is passed through the network to generate predictions. Following this, a backward pass systematically adjusts the weights by propagating the error back through the network, allowing for efficient computation of the gradient.

- **Regularization Techniques** To improve the generalization of the model, regularization techniques such as L1 and L2 regularization, dropout, and early stopping may be employed during training. These techniques help prevent the model from overfitting to the noise in the training data.
- **Evaluation and Tuning** Finally, the network's performance must be evaluated using a separate validation set not seen by the model during training. This evaluation helps in tuning hyperparameters like the learning rate, batch size, and the number of epochs, which are crucial for the training performance and outcomes [13].

3.4 Identified Machine Learning Algorithms

In this section, we will introduce the machine learning algorithms identified for our project. These algorithms are used to solve classification as well as regression problems.

3.4.1 Linear Models

Linear Regression

Linear regression models the linear relationship between the dependent and independent variables by fitting a line, or a hyperplane in the case of multiple variables, that minimizes the sum of the squared residuals. The mathematical expression for simple linear regression is as follows:

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{3.7}$$

where:

- Y is the dependent variable we want to predict.
- X is the independent variable used for prediction.
- β_0 is the intercept of the line (y-intercept).
- β_1 is the slope of the line.
- ϵ is the error term, representing the difference between the actual and predicted values.

The goal is to find values of β_0 and β_1 that minimize the sum of the squared differences between the predicted and actual values of the dependent variable, a method known as the least squares method.

Linear regression is based on the following assumptions:

- **Linearity**: The relationship between the independent and dependent variables is linear.

- **Independence**: The residuals are independent, meaning the residuals from one prediction have no effect on the residuals from another.
- **Homoscedasticity**: The variance of the errors is constant across all levels of the independent variables.
- **Normality**: For any fixed value of the independent variables, the dependent variable is normally distributed.

Violation of these assumptions can lead to biased and inefficient parameter estimates and incorrect inference.

Strengths and Limitations of Linear Regression:

Strengths:

- **Simplicity**: Linear regression is straightforward to understand and explain.
- **Efficiency**: It is computationally efficient, suitable for large datasets or many features.
- **Predictive Performance**: It can provide strong predictive performance with sufficient and relevant features.
- **Flexibility**: Can model non-linear relationships with polynomial or interaction terms.

Limitations:

- **Linearity Assumption**: Assumes a linear relationship, which might not always hold.
- **Sensitive to Outliers**: Outliers can significantly impact the model.
- **Multicollinearity**: Does not handle multicollinearity well, making estimates less reliable.
- **Overfitting and Underfitting**: Can overfit with many features and underfit if the relationship is complex.
- **Lack of Fit Tests**: Challenging to define model complexity optimally without trial and error.

Lasso Regression

Lasso Regression, or Least Absolute Shrinkage and Selection Operator, is a modification of linear regression that incorporates a regularization technique to enhance the model's predictability and interpretability. Unlike standard linear regression, Lasso adds a penalty term to the cost function to control the magnitude of the regression coefficients, promoting sparsity and potentially reducing overfitting. The cost function for Lasso regression is given by:

$$\text{Minimize: } \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{3.8}$$

where:

- y_i is the i-th observed value of the dependent variable.
- β_0 is the y-intercept.
- β_j are the coefficients for the predictor variables x_{ij}.
- λ is the regularization parameter controlling the strength of the penalty imposed on the size of the coefficients.

Lasso regression helps prevent overfitting by including a penalty term, which is the sum of the absolute values of the coefficients. This penalty discourages the model from fitting the training data too closely. Lasso can also set some coefficient values to zero, thereby performing feature selection, which simplifies the model and may reveal the most influential features.

Strengths:

- **Feature Selection**: Lasso regression is effective at reducing the number of features in the model by setting coefficients to zero for less important variables.
- **Prevention of Overfitting**: By penalizing large coefficients, Lasso helps ensure the model does not overfit the training data.
- **Handling Multicollinearity**: Lasso can address multicollinearity by selecting one feature from a group of highly correlated features and shrinking the others to zero.

Limitations:

- **Selection of Regularization Parameter**: The success of Lasso regression heavily depends on the choice of λ. Improper selection can lead to underfitting or overlooking significant variables.
- **Feature Selection Limitations**: While useful, the feature selection mechanism can sometimes be arbitrary, particularly among highly correlated variables.
- **Difficulty Handling Complex Relationships**: Lasso may not capture complex, nonlinear relationships as effectively as other models, such as those including interaction terms or polynomial features [15].

Ridge Regression

Ridge Regression is a variant of linear regression that incorporates a regularization term to improve the model's accuracy and interpretability. Unlike Lasso Regression, which uses the absolute values of the coefficients as the penalty, Ridge Regression uses the squares of the coefficients. This distinction affects how the model handles the coefficients. The cost function for Ridge Regression is defined as:

$$\text{Minimize: } \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3.9}$$

where:

- y_i is the i-th observed value of the dependent variable.
- β_0 is the y-intercept.
- β_j are the coefficients for the predictor variables x_{ij}.
- λ is the regularization parameter that controls the extent of the penalty.

This penalty term, being a sum of the squares of the coefficients, discourages large values but unlike Lasso, it does not set coefficients to zero. This means that features are less likely to be entirely excluded from the model.

Multicollinearity, where two or more predictors are highly correlated, can destabilize a regression model. Ridge Regression addresses this issue by imposing a penalty that "shrinks" the coefficients, thus distributing the influence of correlated predictors more evenly across them. This bias introduced by the penalty helps to reduce variance and improve model generalizability.

Strengths:

- **Prevention of Overfitting**: By using a penalty term, Ridge helps to keep the model complexity in check, thereby preventing overfitting.
- **Handling Multicollinearity**: It effectively manages multicollinearity among predictors, leading to a more robust model.
- **Performance with Many Features**: Ridge Regression performs well in scenarios with a high number of features, including cases where the number of features exceeds the number of observations.

Limitations:

- **Selection of Regularization Parameter**: The effectiveness of Ridge Regression heavily relies on the correct choice of λ. Finding the optimal value often requires cross-validation.
- **Lack of Feature Selection**: Unlike Lasso, Ridge does not reduce coefficients to zero, which means it does not perform explicit feature selection and can result in a model that is less interpretable.
- **Bias Introduction**: The regularization term introduces bias into the model, which could potentially lead to underfitting if the value of λ is too high [15].

3.4.2 Random Forest Ensemble

Random Forest is an ensemble of decision tree algorithms. It is an extension of bootstrap aggregation (bagging) of decision trees and can be used for classification and regression problems.

In bagging, multiple decision trees are constructed where each tree is created from a different bootstrap sample of the training dataset. A bootstrap sample is a sample of the training dataset where an example may appear more than once in the sample, a process known as “sampling with replacement”.

Bagging is an effective ensemble algorithm as each decision tree is fit on a slightly different training dataset, and in turn, has a slightly different performance. Unlike normal decision tree models, such as classification and regression trees (CART), trees used in the ensemble are unpruned, making them slightly overfit to the training dataset. This overfitting is desirable as it helps to make each tree more different and have less correlated predictions or prediction errors.

Predictions from the trees are averaged across all decision trees, resulting in better performance than any single tree in the model. A prediction on a regression problem is the average of the predictions across the trees in the ensemble, mathematically represented as:

$$\hat{y}(x) = \frac{1}{B} \sum_{i=1}^{B} T_i(x) \tag{3.10}$$

where:

- $\hat{y}(x)$ is the predicted output for input x.
- B is the number of trees in the forest.
- $T_i(x)$ is the prediction of the i-th tree.

A prediction on a classification problem is the majority vote for the class label across the trees in the ensemble:

$$\hat{y}(x) = \text{mode}\{T_1(x), T_2(x), \ldots, T_B(x)\} \tag{3.11}$$

Random Forest involves constructing a large number of decision trees from bootstrap samples from the training dataset, like bagging.

Unlike bagging, Random Forest also involves selecting a subset of input features (columns or variables) at each split point in the construction of the trees. Typically, constructing a decision tree involves evaluating the value for each input variable in the data in order to select a split point. By reducing the features to a random subset that may be considered at each split point, it forces each decision tree in the ensemble to be more different. The effect is that the predictions, and in turn, prediction errors, made by each tree in the ensemble are more different or less correlated. When the predictions from these less correlated trees are averaged to make a prediction, it often results in better performance than bagged decision trees [10].

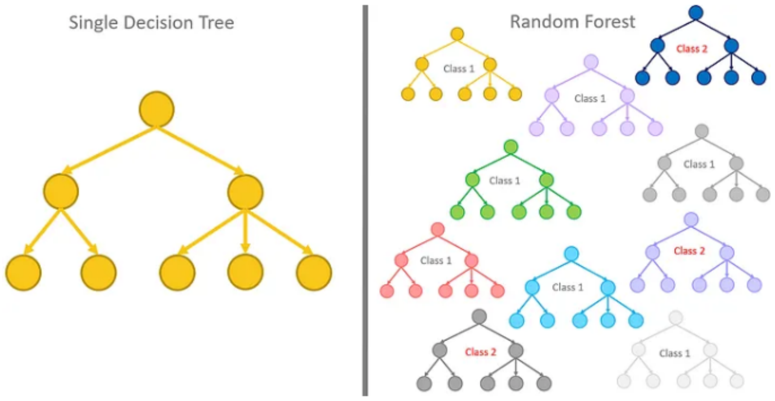


Figure 3.11: Random Forest vs Single Decision Tree [10]

3.4.3 Gradient Boosted Trees

The Gradient Boosted Trees (GBDT) algorithm is another powerful algorithm based on the ensemble learning approach for regression. It trains a set of decision trees sequentially, focusing on the residuals (error gradient) of each tree. This sequential approach allows for continuous error reduction.

The GBDT algorithm operates according to the following steps:

1. For the first iteration, the residuals (r_i) are initialized with the actual values (y_i) for all data points.
2. A number B of trees is chosen to be trained.
3. For each iteration b from 1 to B:
   a. A tree $\hat{f}_b$ with cuts ($d + 1$ leaves) is trained.
   b. The model is updated by adding this tree multiplied by a coefficient $\lambda < 1$:
   $$\hat{f}(x) = \hat{f}(x) + \lambda \hat{f}_b(x) \tag{1.16}$$
   c. The residuals are updated:
   $$r_i = r_i - \lambda \hat{f}_b(x_i) \tag{1.17}$$
4. The final model is calculated by combining the predictions of all trees:
   $$\hat{f}(x) = \sum(\lambda \hat{f}_b(x)), \text{ for } b \text{ ranging from 1 to } B.$$

The GBDT algorithm is used to progressively improve predictions by focusing on residual errors. Trees are trained iteratively to capture the residual relationships not explained by previous trees. The coefficients λ control the importance of each tree in the final model.

To visually represent the tree training process and the improvement of predictions, you can use graphs showing the successive predictions at each step of the algorithm, along with the updated residuals. These graphs can be generated using graphical libraries such as Matplotlib, plotting the predicted values by the model at each stage.

It is important to note that the detailed implementation of the GBDT algorithm may vary depending on the library or programming language used. You should refer to the specific documentation of the GBDT library you are using for concrete examples and detailed instructions on visualizing the ensemble learning process of the trees.

XGBoost (eXtreme Gradient Boosting)

XGBoost, or eXtreme Gradient Boosting, is an advanced implementation of gradient boosting designed to be highly efficient, flexible, and portable. It is an open-source library that provides a machine learning algorithm under the Gradient Boosting framework. XGBoost provides a scalable and highly efficient implementation of gradient boosted decision trees designed for speed and performance.

The key features of XGBoost that distinguish it from traditional gradient boosting include:

- **Regularization:** XGBoost introduces regularization terms in the objective function, which help in reducing overfitting and improving model performance. This makes XGBoost more robust than standard gradient boosted trees.

- **Parallel Processing:** It utilizes the power of multi-core computers to speed up tree construction. This makes XGBoost significantly faster than traditional gradient boosting.

- **High Flexibility:** XGBoost allows users to define custom optimization objectives and evaluation criteria, adding a layer of flexibility not found in many other traditional algorithms.

- **Handling Missing Values:** XGBoost has an in-built routine to handle missing values. When XGBoost encounters a missing value at a node, it learns the direction to assign to missing values based on what will result in the best training continuation.

- **Tree Pruning:** The tree pruning in XGBoost is depth-first rather than breadth-first. Unlike traditional gradient boosting, which stops splitting a node as soon as it encounters a negative loss, XGBoost will split up to the max_depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.

- **Built-in Cross-Validation:** XGBoost allows a user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting rounds for a given model.

- **Effective Handling of Sparse Data:** XGBoost is designed to be efficient with sparse data and works well with matrices that have a large number of zeros [9].

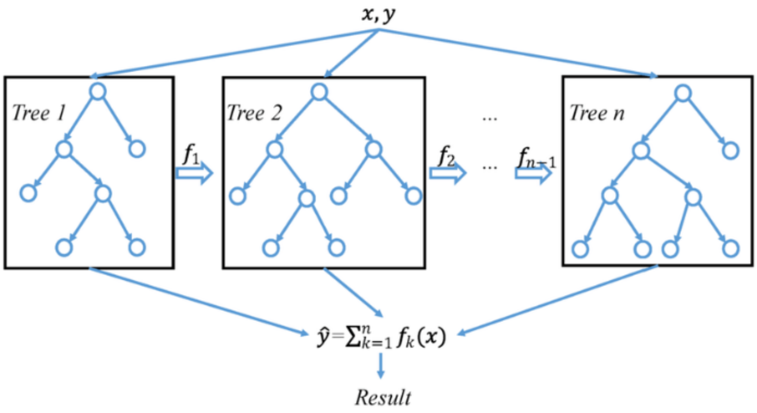


Figure 3.12: A general architecture of XGBoost [9]

3.4.4 SVM (Support Vector machines)

Support Vector Machines (SVM) are based on solid mathematical concepts. The goal of SVM is to find a hyperplane that maximizes the margin between the classes in a multidimensional space. Mathematically, this involves solving a quadratic optimization problem. The hyperplane is defined by a linear decision function, which can be expressed as:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b\right)$$

where α_i are the Lagrange multipliers, y_i are the class labels, x_i are the support vectors, and $K(x, x_i)$ is a kernel function that measures the similarity between data points. Commonly used kernel functions include the linear kernel, the polynomial kernel, and the RBF (Radial Basis Function) kernel.

In terms of regression, the SVM model can be used to model nonlinear relationships by introducing the kernel trick. The idea is to replace x_i with a nonlinear function $\phi(x_i)$ and to use the dual of the optimization program. The quadratic minimization problem can be formulated as follows:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \tag{3.12}$$

$$\text{s.t.} \quad |y_i - \mathbf{w}^T\phi(x_i)| \leq \epsilon + \xi_i, \quad \xi_i \geq 0 \tag{3.13}$$

where $\mathbf{w}$ is the weight vector, C is a tolerance coefficient, ϵ is the error margin, and ξ_i are slack variables. This mathematical program can be solved using the method of Lagrange or other optimization methods [16].

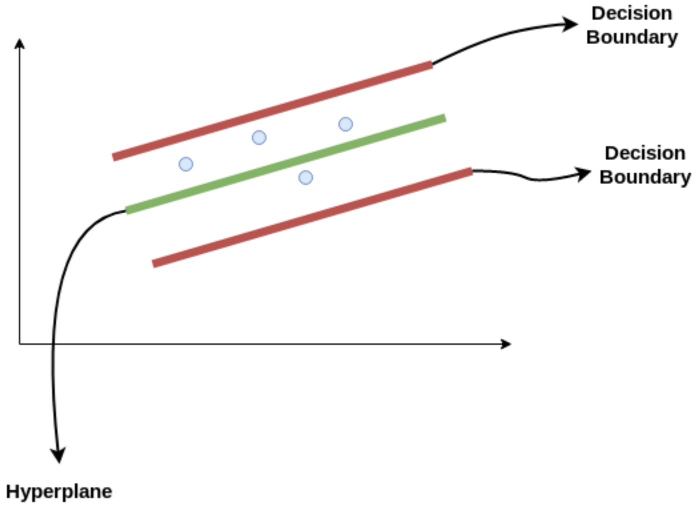


Figure 3.13: Support Vector Regression [7]

If we consider these two red lines as the decision boundary and the green line as the hyperplane. Our objective, when we are moving on with SVR, is to basically consider the points that are within the decision boundary line. Our best fit line is the hyperplane that has a maximum number of points.

3.5 Models Evaluation Metrics

To evaluate and compare the performances of prediction models, several error metrics have been chosen to provide a complete view of their performance, taking into account the specific characteristics of the data and the predictive models:

- **Mean Absolute Error (MAE)**: Measures the average magnitude of the errors in a set of predictions, without considering their direction.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3.14}$$

- **Mean Squared Error (MSE)**: Measures the average of the squares of the errors—that is, the average squared difference between the estimated values and what is estimated.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.15}$$

- **Root Mean Squared Error (RMSE)**: The square root of the average of squared differences between prediction and actual observation.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.16}$$

- **R-squared Score (R² Score)**: A statistical measure of how close the data are to the fitted regression line.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{3.17}$$

3.6 Conclusion

This chapter provided a comprehensive overview of various machine learning techniques and approaches, emphasizing their applications and significance in different contexts. We began with a general introduction to machine learning, defining its core concepts and outlining the essential steps in an end-to-end machine learning project. We then classified machine learning systems into supervised, unsupervised, semi-supervised, and reinforcement learning categories, detailing the specific methods and algorithms used in each. Furthermore, we explored deep learning, highlighting neural network architectures and their applications. Lastly, we discussed key machine learning algorithms, including linear models, random forests, gradient-boosted trees, and support vector machines, and concluded with an examination of evaluation metrics to assess model performance. Through this chapter, we established a foundational understanding of machine learning's diverse techniques and their practical implementations.

Chapter 4

Time Series Analysis and Forecasting Methods

4.1 Introduction

In this chapter, we delve into the essentials of time series analysis. We'll explore the nature of time series data, and dissect its key components like trend, seasonality, cycles, and residuals. The chapter also discusses stationarity, a critical concept in time series, and go through different methods for decomposing and forecasting time series.

4.2 Overview of time series data and its characteristics

4.2.1 Time Series

A time series is a collection of data points gathered over a period of time and ordered chronologically. The primary characteristic of a time series is that it's indexed or listed in time order, which is a critical distinction from other types of data sets. If you were to plot the points of time series data on a graph, one of your axes would always be time. The time series data can be represented by a mathematical function in the following manner:

$$X(t) = \{X_1, X_2, \ldots, X_n\} \tag{4.1}$$

Here, $X(t)$ denotes the series of values of the time series at various time points t, and $X_1, X_2, \ldots, X_n$ represent the numeric values of the time series at each specific time instance. The notation { } signifies that these values are arranged chronologically over time. Thus, $X(t)$ is a function that, when evaluated at different time instances, yields the corresponding values of the time series.

Time series metrics refer to a piece of data that is tracked at an increment in time. For instance, a metric could refer to how much inventory was sold in a store from one day to the next.

4.2.2 Time Series Components

real-world time series are often governed by a (deterministic) trend and they might have (deterministic) cyclical or seasonal components in addition to the irregular/remainder (stationary process) component:

- **Trend** $X(t)$**:** The trend component captures long-term movements or changes in the mean of the time series. Trends reflect underlying patterns or tendencies in the data, such as overall growth or decline over time.
- **Seasonal Effects** $S(t)$**:** Seasonal effects represent cyclical fluctuations that occur regularly within a specific time period, such as daily, weekly, monthly, or annually. These fluctuations are often related to calendar events, holidays, or seasonal factors and exhibit a repetitive pattern.
- **Cycles** $C(t)$**:** Cycles represent additional cyclical fluctuations in the time series data that are not accounted for by seasonal effects. Unlike seasonal patterns, cycles may occur irregularly and can vary in duration. These fluctuations may be influenced by economic or business cycles, technological advancements, or other external factors.
- **Residuals** $R(t)$**:** Residuals, or irregular components, also known as errors or noise, capture the remaining random or systematic fluctuations in the data that cannot be explained by the trend, seasonal effects, or cycles. These residuals represent unmodeled variation or uncertainty in the time series [22].

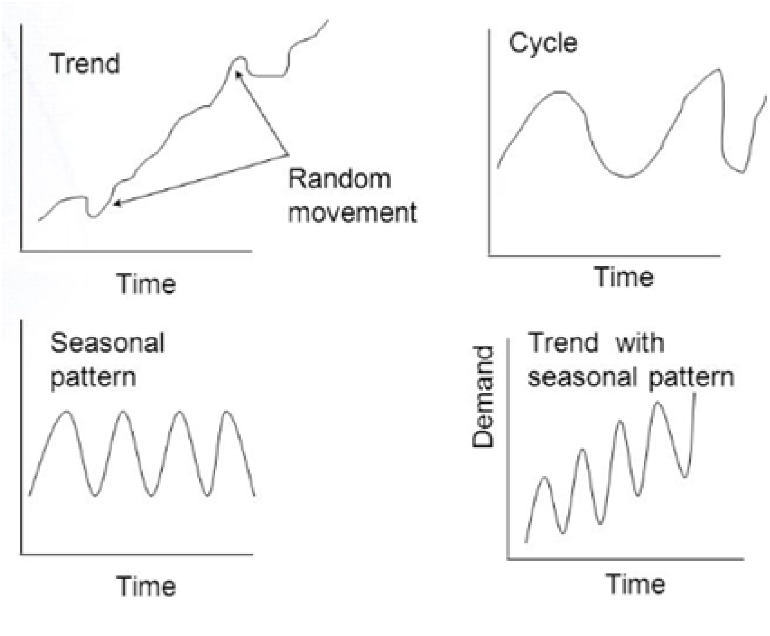


Figure 4.1: Time series components [22]

4.2.3 Time Series Decomposition

The general mathematical representation of the decomposition approach is given by:

$$X(t) = f(T(t), S(t), E(t)) \tag{4.2}$$

where:

- $X(t)$:is the time series value (actual data) at period t;
- $T(t)$:is a deterministic trend-cycle or general movement component;
- $S(t)$:is a deterministic seasonal component;
- $E(t)$:is the irregular (remainder or residual) (stationary) component.

The exact functional form of $f(\cdot)$ depends on the decomposition method used. A common approach is to assume that the equation has an additive form:

$$X(t) = T(t) + S(t) + E(t) \tag{4.3}$$

where trend, seasonal and irregular components are simply added together to give the observed series.

Alternatively, the multiplicative decomposition has the form:

$$X(t) = T(t) \cdot S(t) \cdot E(t) \tag{4.4}$$

where trend, seasonal and irregular components are multiplied together to give the observed series.

In both additive and multiplicative cases, the series $X(t)$ is called a trend stationary (TS) series.

The classical decomposition method aims to create separate models for each of these four elements and then combine them to reconstruct the original time series. This can be done either additively, by summing the individual components, or multiplicatively, by multiplying them together [21].

4.3 The Stationarity Concept

In a fundamental sense, stationarity denotes that the statistical characteristics of a process generating a time series remain consistent over time. It does not imply that the series remains constant, but rather that the manner in which it changes remains stable. This stability is akin to that of a linear function, where the rate of change remains constant as the variable x progresses.

The importance of stationarity lies in its facilitation of analysis. As a subset within the broader spectrum of possible models of reality, stationary processes offer a more manageable framework for modeling and investigation. Moreover, their predictable nature suggests potential for accurate prediction, enhancing their utility in practical applications.

Stationarity serves as a fundamental assumption underpinning various practices and methodologies in time series analysis, including trend estimation, forecasting, and causal inference. Its pervasive presence underscores the necessity of understanding, detecting, and modeling stationarity for the effective application of prominent analytical tools and procedures.

Consequently, proficiency in discerning whether data conforms to a stationary process and the ability to transform it accordingly are essential skills in navigating many scenarios involving time series analysis.

Stationarity, in any manifestation, constitutes a characteristic inherent to a stochastic process rather than to any finite or infinite realization thereof. In essence, it denotes the stability of statistical properties across time for the process generating a time series. This stability encompasses consistency in parameters such as mean, variance, and autocovariance, indicating that these properties remain constant over time intervals within the process [17].

4.3.1 Strong stationarity

Strong stationarity requires the shift-invariance (in time) of the finite-dimensional distributions of a stochastic process. This means that the distribution of a finite sub-sequence of random variables of the stochastic process remains the same as we shift it along the time index axis.

Formally, the discrete stochastic process $X = \{x_i; i \in \mathbb{Z}\}$ is stationary if:

$$F_X(x_{t_1+\tau}, \ldots, x_{t_n+\tau}) = F_X(x_{t_1}, \ldots, x_{t_n}) \tag{4.5}$$

for $T \subset \mathbb{Z}$ with $n \in \mathbb{N}$ and any $\tau \in \mathbb{Z}$. For continuous stochastic processes the condition is similar, with $T \subset \mathbb{R}$, $n \in \mathbb{N}$ and any $\tau \in \mathbb{R}$ instead.

This is the most common definition of stationarity, and it is commonly referred to simply as stationarity. It is sometimes also referred to as strict-sense stationarity or strong-sense stationarity.

4.3.2 Weak stationarity

Weak stationarity only requires the shift-invariance (in time) of the first moment and the cross moment (the auto-covariance). This means the process has the same mean at all time points, and that the covariance between the values at any two time points, t and $t-k$, depend only on k, the difference between the two times, and not on the location of the points along the time axis.

Formally, the process $X = \{x_i; i \in \mathbb{Z}\}$ is weakly stationary if:

- The first moment of x_i is constant; i.e. $\forall t,\ \mathbb{E}[x_i] = \mu$
- The second moment of x_i is finite for all t; i.e. $\forall t,\ \mathbb{E}[x_i^2] < \infty$ (which also implies of course $\mathbb{E}[(x_i - \mu)^2] < \infty$; i.e., that the variance is finite for all t.
- The cross moment; i.e., the auto-covariance depends only on the difference $u - v$; i.e. $\forall u, v, a,\ \mathrm{cov}(x_u, x_v) = \mathrm{cov}(x_{u+a}, x_{v+a})$.

  This third condition implies that every lag $\tau \in \mathbb{N}$ has a constant covariance value associated with it:

  $$\mathrm{cov}(X_{t_1}, X_{t_2}) = K_{XX}(t_2 - t_1, 0) = K_{XX}(\tau) \tag{4.6}$$

  Note that this directly implies that the variance of the process is also constant, since we get that for all $t \in \mathbb{N}$:

  $$\mathrm{Var}(X_t) = \mathrm{cov}(X_t, X_t) = K_{XX}(t, t) = K_{XX}(0) = d \tag{4.7}$$

This paints a specific picture of weakly stationary processes as those with constant mean and variance. Their properties are contrasted nicely with those of their counterparts in the figure below.

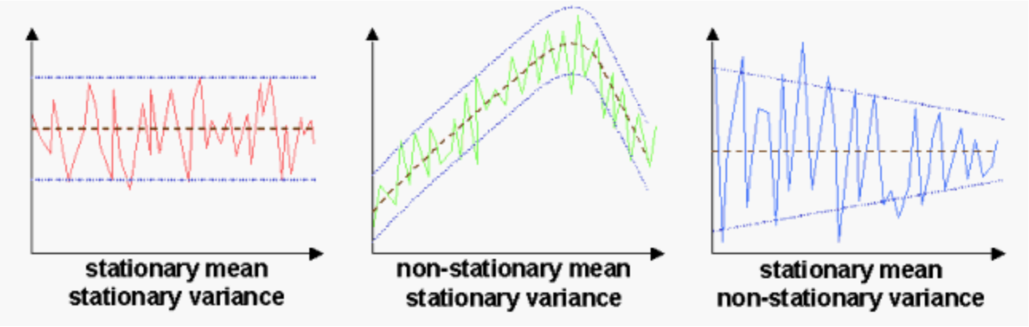


Figure 4.2: Constancy in mean and variance [21]

Other common names for weak stationarity are wide-sense stationarity, weak-sense stationarity, covariance stationarity and second order stationarity.

4.3.3 Methods to Check Stationarity

There are several methods used to check for stationarity in a time series. The most common approaches include:

Visual Inspection:

- Plotting the time series data can provide insights into its stationarity. Stationary series typically do not have trends or seasonal effects, and their variances appear constant over time.
- Rolling statistics plots, such as rolling mean and rolling standard deviation, can also be used to visually assess stationarity. A constant rolling mean and a constant rolling standard deviation over time suggest stationarity.

Statistical Tests:

- **Augmented Dickey-Fuller (ADF) Test**: This is one of the most widely used statistical tests. It tests the null hypothesis that a unit root is present in the time series sample. A lower p-value (< 0.05) typically suggests rejecting the null hypothesis, indicating stationarity.
- **Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test**: Contrary to the ADF test, the null hypothesis for the KPSS test assumes that the series is stationary. A high p-value suggests the series is stationary, whereas a low p-value (typically < 0.05) would lead to rejecting the null hypothesis, indicating non-stationarity.
- **Phillips-Perron (PP) Test**: This test is used to detect a unit root in a time series without specifying the order of the autoregressive model, accounting for autocorrelation and heteroscedasticity in the error terms [20].

4.3.4 Converting Non-Stationary Into Stationary

Various methods are employed to achieve stationarity, each addressing different characteristics of non-stationarity:

Differencing:

Differencing is performed by computing the difference between consecutive observations.

Mathematically, if X_t represents the original time series, the first difference ΔX_t is given by:

$$\Delta X_t = X_t - X_{t-1} \tag{4.8}$$

For seasonal differencing, where the series has a seasonal period of S, the seasonal difference is:

$$\Delta_S X_t = X_t - X_{t-S}$$

This process can help eliminate the trend and reduce seasonality in the data.

Transformation:

Common transformations include logarithmic, square root, and power (e.g., Box-Cox) transformations.

For a logarithmic transformation:

$$Y_t = \log(X_t) \tag{4.9}$$

A square root transformation is expressed as:

$$Y_t = \sqrt{X_t} \tag{4.10}$$

Box-Cox transformation, which is more general, can be represented as:

$$Y_t(\lambda) = \begin{cases} \frac{X_t^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \log(X_t) & \text{if } \lambda = 0. \end{cases} \tag{4.11}$$

These transformations aim to stabilize variance in the series.

Detrending:

Detrending involves removing the trend component from the time series. If T_t represents the trend component, the detrended series D_t can be obtained by subtracting the trend from the original series:

$$D_t = X_t - T_t \tag{4.12}$$

The trend T_t can often be estimated using linear regression or other fitting techniques.

4.4 Time Series analysis

Time series analysis is the systematic examination and interpretation of patterns, trends, and fluctuations present in time series data. It involves applying statistical methods, mathematical models, and analytical techniques to analyze and extract meaningful insights from the sequential data points collected over a period of time. Time series analysis aims to understand the underlying structure and behavior of the time series, identify influential factors, and make predictions or forecasts about future values. By analyzing the temporal dependencies and relationships within the data, time series analysis enables practitioners to uncover patterns, detect anomalies, and derive actionable insights for decision-making across various domains and industries.

The study of time series aims primarily to understand temporal trends and data behaviors over time. The main objectives of time series analysis include:

- **Description**: Describing the main characteristics of the time series, such as trends, seasonal fluctuations, random variations, cycles, etc.
- **Modeling**: Developing statistical models that quantify temporal trends and forecast future values of the time series.
- **Forecasting**: Using statistical models to predict future values of the time series and to assess the uncertainty associated with these forecasts.
- **Intervention**: Evaluating the impact of interventions or events on the time series and determining if these interventions have had a significant effect on trends or data behaviors.
- **Control**: Using statistical models to control and monitor the performance of a process over time, detecting anomalies and deviations from expected values.

However, one of the most important objectives of time series analysis is forecasting.

4.5 Time Series Forecasting

Time series forecasting refers to the process of predicting future values or trends of a time series based on past observations. It involves using historical data points to develop mathematical models or algorithms that can project future values of the time series. The objective of time series forecasting is to make accurate predictions about future behavior, allowing businesses and organizations to anticipate trends, plan resources, and make informed decisions.

4.5.1 Time Series Forecasting Methods

4.5.1.1 Statistical Methods

- **Auto-Regressive Models**:

  Autoregressive (AR) models are a foundational tool in time series analysis and forecasting, utilizing the concept that current observations in a dataset can be expressed as a linear combination of past values. The essence of an AR model is its dependence on previous observations to predict future values, making it particularly useful for univariate time series data where the relationship of interest is within the series itself.

  The order of an AR model, denoted as p, determines the number of lagged observations included in the model, with these past values serving as predictors for the current value. The model parameters, or coefficients, are typically estimated through a process that minimizes the error between the model's predictions and the actual observed values.

  A key strength of AR models lies in their ability to capture and elucidate the dynamics of time series, especially when data exhibits clear trends or patterns. However, they operate under the assumption of linearity between past and current observations and often require a substantial amount of data to yield accurate predictions. Despite this, the models incorporate a term for random noise, acknowledging the potential for unpredictable variation and signaling that refinements to the model may be necessary for improved accuracy.

In mathematical terms, an AR model of order p can be represented as:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \epsilon_t \tag{4.13}$$

where y_t is the current value being modeled, $\phi_1, \phi_2, \ldots, \phi_p$ are the coefficients that weight the influence of each lagged value, and ϵ_t represents the random error term at time t. This formulation underlines that each outcome y_t is predicted from a linear regression of its own previous values, with no other independent variables aside from the series' past results.

- **Moving Average Models**:

  Moving Average (MA) models are key instruments in time series analysis, particularly for forecasting and understanding the stochastic nature of a series. MA models postulate that the current value of a time series can be characterized as a linear combination of past error terms, which are the residuals from previous forecasts. These models are primarily used when the interest lies in smoothing out random fluctuations and detecting underlying patterns.

  The order of an MA model, often denoted by q, specifies the number of past error terms included in the prediction of the current value. The estimation of the model involves determining the weights, or coefficients, that best combine these error terms to match the observed values, effectively capturing the 'moving average' of the series.

  An MA model's main advantage is its capability to model the random shocks that affect a time series, thus providing a clearer picture of the intrinsic data patterns without long-term trends or seasonality. However, like AR models, MA models also assume linearity in the relationship between current values and past errors and may necessitate ample data for precise parameter estimation.

  Mathematically, an MA model of order q is typically written as:

$$X_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} \tag{4.14}$$

  where:

  - X_t is the current value of the series,
  - μ is the mean of the series,
  - ϵ_t is the error term at time t,
  - $\theta_1, \theta_2, \ldots, \theta_q$ are the coefficients of the model, representing the impact of the error terms on the current value.

  This equation encapsulates the essence of the MA model, where each value X_t is the outcome influenced by a series of past forecast errors, offering insights into the volatility and noise components of the time series.

- **Autoregressive Moving Average Models**:

  Autoregressive Moving Average (ARMA) models are robust tools for analyzing and forecasting stationary time series data, synthesizing the features of Autoregressive (AR) and Moving Average (MA) models. An ARMA model is adept at capturing the dependencies on both past observations and past forecast errors.

The ARMA model is specified by two parameters, p and q, where p represents the order of the AR part and q denotes the order of the MA part. The AR component p identifies the momentum or mean reversion from past values, and the MA component q captures the influence of past shocks or random disturbances.

The mathematical representation of an ARMA model with orders p and q is given by:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t \tag{4.15}$$

where:

- X_t is the current value of the series,
- $\phi_1, \phi_2, \ldots, \phi_p$ are the coefficients of the AR terms,
- $\theta_1, \theta_2, \ldots, \theta_q$ are the coefficients of the MA terms,
- ϵ_t represents the error term at time t, generally assumed to be white noise.

By integrating both AR and MA components, ARMA models offer a sophisticated approach to fitting a broad range of time series behaviors, providing a versatile framework for accurate forecasting.

- **Autoregressive Moving Integrated Average Models**:

  The Autoregressive Integrated Moving Average (ARIMA) model is an extension that is specifically designed to analyze and forecast non-stationary time series data. ARIMA models incorporate the techniques of differencing to convert non-stationary data into a stationary form, making it amenable to the methods used in ARMA models.

  An ARIMA model is characterized by three parameters: p, d, and q, where:

  - p is the order of the Autoregressive (AR) part, representing the number of lagged terms of the series,
  - d is the degree of differencing required to make the series stationary,
  - q is the order of the Moving Average (MA) part, indicating the number of lagged forecast errors in the prediction equation.

  The ARIMA model can be mathematically represented as:

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)(1 - B)^d X_t = (1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q)\varepsilon_t \tag{4.16}$$

  where B is the backshift operator, X_t represents the time series, $\phi_1, \phi_2, \ldots, \phi_p$ are the coefficients of the AR terms, $\theta_1, \theta_2, \ldots, \theta_q$ are the coefficients of the MA terms, and ε_t is the error term [19].

- **Seasonal Autoregressive Integrated Moving Average Models**:

  The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is an advanced form of the ARIMA model designed specifically to handle and forecast time series data exhibiting seasonality. SARIMA models extend ARIMA by incorporating both non-seasonal and seasonal factors in a unified framework.

  A SARIMA model is characterized by its parameters: (p, d, q) for the non-seasonal components and $(P, D, Q)_s$ for the seasonal components:

  - p is the order of the non-seasonal Autoregressive (AR) part, representing the number of lagged terms of the series.
  - d is the degree of non-seasonal differencing required to render the series stationary.

- q is the order of the non-seasonal Moving Average (MA) part, indicating the number of lagged forecast errors in the prediction equation.
- P is the order of the seasonal Autoregressive part, analogous to p but for the seasonal data.
- D is the degree of seasonal differencing, addressing seasonality in the series.
- Q is the order of the seasonal Moving Average part, similar to q but for seasonal forecast errors.
- s represents the length of the seasonal cycle (e.g., 12 for monthly data with an annual cycle).

The SARIMA model can be mathematically represented as follows:

$$\begin{aligned}(1-\phi_1 B-\phi_2 B^2-\ldots-\phi_p B^p)(1-B)^d(1-\Phi_1 B^s-\Phi_2 B^{2s}-\ldots-\Phi_P B^{Ps})(1-B^s)^D X_t\\ =(1+\theta_1 B+\theta_2 B^2+\ldots+\theta_q B^q)(1+\Theta_1 B^s+\Theta_2 B^{2s}+\ldots+\Theta_Q B^{Qs})\varepsilon_t\end{aligned} \tag{4.17}$$

where:

- B is the backshift operator.
- X_t represents the time series data.
- $\phi_1, \phi_2, \ldots, \phi_p$ are the coefficients of the non-seasonal AR terms.
- $\theta_1, \theta_2, \ldots, \theta_q$ are the coefficients of the non-seasonal MA terms.
- $\Phi_1, \Phi_2, \ldots, \Phi_P$ are the coefficients of the seasonal AR terms.
- $\Theta_1, \Theta_2, \ldots, \Theta_Q$ are the coefficients of the seasonal MA terms.
- ε_t is the error term.

- **Variations in Statistical Models for Time Series**:

  Several variations on statistical models can be used for analyzing and forecasting time series data, but the most important ones are:

  - **Exponential Smoothing Models:** These models are used for forecasting time series data that exhibit trends and/or seasonality. They are based on weighting recent observations more heavily than older ones.
  - **State Space Models:** These models are used for modeling dynamic systems with unobserved variables. They are particularly useful for modeling time series data with complex patterns.
  - **GARCH Models:** These models are used for modeling time series data with volatility that changes over time. They are commonly used in finance for modeling stock returns and other financial time series data.
  - **Structural Time Series Models:** This model is used for modeling time series with an underlying structure and a stochastic component. They are used to model the underlying causal factors of a time series [19].

4.5.1.2 FB Prophet

Prophet is an open source library published by Facebook that is based on decomposable (trend+seasonality+holidays) models. It provides us with the ability to make time series predictions with good accuracy using simple intuitive parameters and has support for including impact of custom seasonality and holidays.

The Prophet Forecasting Model uses a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{4.18}$$

Where:

- **$g(t)$:** piece-wise linear or logistic growth curve for modeling non-periodic changes in time series.
- **$s(t)$:** periodic changes (e.g. weekly/yearly seasonality).
- **$h(t)$:** effects of holidays (user provided) with irregular schedules.
- **ϵ_t:** error term accounts for any unusual changes not accommodated by the model

Components:

- **Trend:** In Prophet, trend modeling involves fitting a piece-wise linear curve to the non-periodic component of the time series data. This essentially means that rather than assuming a single linear trend over the entire time period, Prophet breaks the time series into smaller intervals and fits a separate linear trend to each interval.
- **Seasonality:** To fit and forecast the effects of seasonality, prophet relies on Fourier series to provide a flexible model. Seasonal effects $s(t)$ are approximated by the following function:

  $$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right)\right) \tag{4.19}$$

  P is the period (365.25 for yearly data and 7 for weekly data)

  The Fourier order N that defines whether high frequency changes are allowed to be modeled is an important parameter to set here. For a time series, if the user believes the high frequency components are just noise and should not be considered for modeling, he could set the values of N from to a lower value. If not, N can be tuned to a higher value and set using the forecast accuracy.
- **Holidays and events:** Holidays and events incur predictable shocks to a time series. For instance, Diwali in India occurs on a different day each year and a large portion of the population buy a lot of new items during this period.

  Prophet allows the analyst to provide a custom list of past and future events. A window around such days are considered separately and additional parameters are fitted to model the effect of holidays and events [23].

4.5.1.3 Machine Learning algorithms

In addition to traditional statistical time series techniques, there is a broad spectrum of machine learning algorithms well-suited for time series forecasting. These include Random Forests and Gradient Boosting Trees as well as Support Vector Machines. The intricate architecture and practical applications of these algorithms is extensively explored and elaborated upon in the previous chapter

Chapter 5

Conception of the solution

5.1 Introduction

In addressing the critical issue of cash flow management within our client's firm, I propose a solution that focuses on two key components: forecasting Accounts Payable (AP) and predicting the timing of Accounts Receivable (AR) payments. This chapter delves into the detailed steps of implementing the proposed solution, structured into two main parts.

- The first section focuses on the prediction of Accounts Receivable (AR) payments. This involves employing advanced machine learning algorithms to analyze payment history and patterns from customers, allowing us to estimate the timing and amount of future cash inflows. The methodology includes data collection, preprocessing, model selection, training, and validation to ensure accuracy and reliability in the predictions.
- The second section addresses the forecasting of Accounts Payable (AP). Here, time series modeling is utilized to forecast future cash outflows related to significant purchases and operational expenses. This segment covers the systematic approach to model building, including data gathering, trend analysis, model fitting, and validation. By accurately predicting future expenditures, the firm can better manage its financial obligations, aligning outflows with the anticipated inflows detailed in the first paragraph.

Together, these two analytical components form a unified strategy for effective cash flow management, essential for supporting and sustaining the firm's operations. This approach ensures robust financial oversight and informed decision-making, crucial for managing business expansion and maintaining operational stability.

5.2 Accounts Receivable Payment Prediction

5.2.1 Data Collection and Consolidation

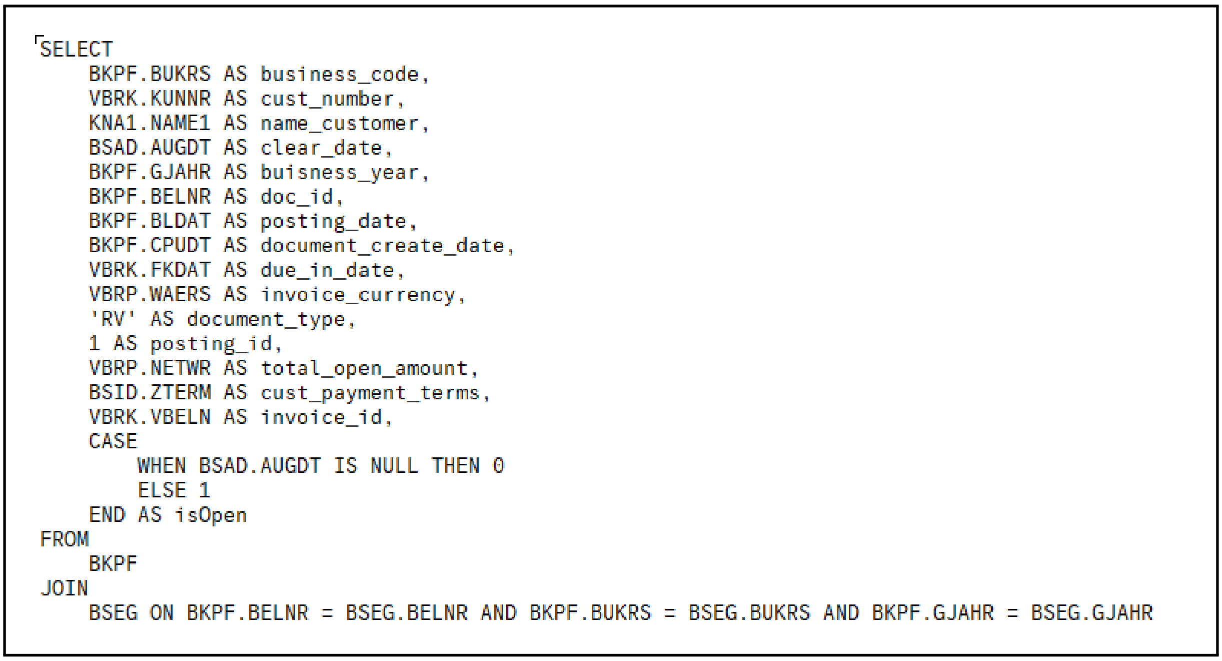
The primary dataset was sourced from an in-memory, column-oriented and relational database – the SAP HANA Data lake. SAP HANA is acknowledged as a good service mainly for transactional query processing in several real-world problems and applications.

In this case, our project's focus is upon the payment outcome of an invoice emitted by firm to a customer, so the level of detail extracted was on the invoice level. each row will correspond to a transaction or invoice, which naturally has as its main characteristics its value, and the date it was created, the customer that incurred in the cost and the date it is due to be payed.

The years 2022 and 2023 were specifically selected for analysis to capture recent trends and shifts in client payment behaviors. Analyzing more recent transactions is crucial, as it reflects the current economic environment and client financial practices, which are more relevant for making accurate predictions and strategic decisions.

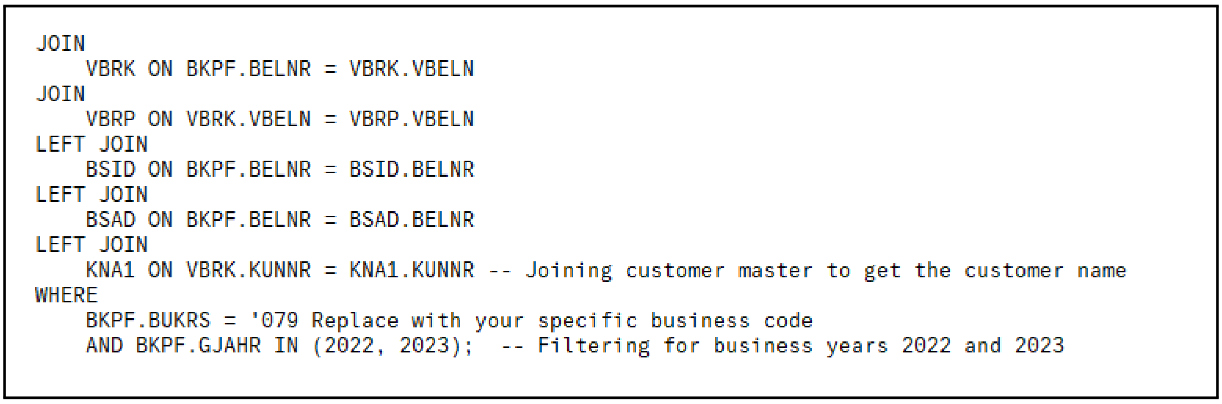
The main tables from which data was extracted include:

- **BKPF (Accounting Document Header):** This table stores header data for accounting documents and includes fields like the document number, company code, fiscal year, and document type. It provides a top-level view of each transaction, crucial for understanding the context and scope of financial entries.
- **BSEG (Accounting Document Segment):** This table contains line item data for each accounting document recorded in BKPF. It is vital for detailed financial analysis as it includes information about amounts, currencies, and accounts involved in transactions.
- **VBRK (Billing Document: Header Data) and VBRP (Billing Document: Item Data):** These tables are central to managing billing information in SAP. VBRK holds header-level data about billing documents such as the billing document number and date, while VBRP contains line-item level data such as product and pricing details. These tables are crucial for analyzing the billing aspect of Accounts Receivable.
- **BSID (Customer Open Items) and BSAD (Customer Cleared Items):** BSID tracks open accounts receivable items, providing insights into unpaid customer invoices. BSAD complements this by recording details once invoices are settled, allowing for a complete cycle analysis of receivables.
- **KNA1 (Customer Master):** This table was used to enrich the transactional data with customer-specific information, offering insights into customer demographics and segmentation, which are essential for personalized analysis and credit management.



```
SELECT
    BKPF.BUKRS AS business_code,
    VBRK.KUNNR AS cust_number,
    KNA1.NAME1 AS name_customer,
    BSAD.AUGDT AS clear_date,
    BKPF.GJAHR AS buisness_year,
    BKPF.BELNR AS doc_id,
    BKPF.BLDAT AS posting_date,
    BKPF.CPUDT AS document_create_date,
    VBRK.FKDAT AS due_in_date,
    VBRP.WAERS AS invoice_currency,
    'RV' AS document_type,
    1 AS posting_id,
    VBRP.NETWR AS total_open_amount,
    BSID.ZTERM AS cust_payment_terms,
    VBRK.VBELN AS invoice_id,
    CASE
        WHEN BSAD.AUGDT IS NULL THEN 0
        ELSE 1
    END AS isOpen
FROM
    BKPF
JOIN
    BSEG ON BKPF.BELNR = BSEG.BELNR AND BKPF.BUKRS = BSEG.BUKRS AND BKPF.GJAHR = BSEG.GJAHR
```

Figure 5.1: SQL Query for Retrieving Relevant Invoice Data from the SAP HANA Database



```
JOIN
    VBRK ON BKPF.BELNR = VBRK.VBELN
JOIN
    VBRP ON VBRK.VBELN = VBRP.VBELN
LEFT JOIN
    BSID ON BKPF.BELNR = BSID.BELNR
LEFT JOIN
    BSAD ON BKPF.BELNR = BSAD.BELNR
LEFT JOIN
    KNA1 ON VBRK.KUNNR = KNA1.KUNNR -- Joining customer master to get the customer name
WHERE
    BKPF.BUKRS = '079 Replace with your specific business code
    AND BKPF.GJAHR IN (2022, 2023);  -- Filtering for business years 2022 and 2023
```

Figure 5.2: SQL Query for Retrieving Relevant Invoice Data from the SAP HANA Database-2

The query filters data for the fiscal years 2022 and 2023 and confines results to transactions within a predefined date range and the firm client's specific company code, thus tailoring the output to the relevant analysis timeframe and organizational context. By incorporating the CASE statement, it also distinguishes between open and cleared transactions, marking them as open or closed based on the presence of a clearing date in BSAD.AUGDT.

The result is a structured output that includes fields like the document type, posting ID, total open amount, and custom payment terms, The table below presents all the pertinent fields related to invoices, along with their descriptions:

Field	**Description**
business_code	This code identifies the business entity within a larger corporation or conglomerate. Each code is unique to a particular business unit.
cust_number	A unique identifier assigned to each customer that facilitates tracking and managing customer-specific transactions within the financial system.
name_customer	The full legal or recognized name of the customer as registered in the company's database, crucial for accurate record-keeping and customer relationship management.
clear_date	The actual date on which the invoice was fully paid and settled, marking the closure of that specific financial transaction.
business_year	The fiscal year during which the transaction was recorded, important for financial reporting, auditing, and analysis.
doc_id	A unique identifier for each document (invoice), used extensively for tracking and retrieval of specific invoice records in the database.
posting_date	The date on which the invoice was officially posted to the company's accounting books, pivotal for financial timeliness and accuracy.
document_create_date	The initial date when the invoice was generated and entered into the system, essential for tracking the inception of the transaction.
due_in_date	The predetermined date by which the invoice is expected to be paid by the customer, based on agreed payment terms.
invoice_currency	Specifies the currency in which the invoice amounts are denoted, important for financial operations in a global business environment involving multiple currencies.
document type	Classifies the type of document.
posting_id	A system-generated identifier that confirms the posting of a transaction, typically used to ensure data integrity and uniqueness in financial postings.
total_open_amount	The total amount billed to the customer that remains unpaid as of the data extraction, critical for managing accounts receivable.
cust_payment_terms	Describes the payment conditions agreed upon with the customer, such as net 30 days, crucial for managing cash flow and credit terms.
invoice_id	Another identifier for the invoice, often used for systems that require a separate identification system from doc_id.
isOpen	A flag indicating whether the invoice is open (1) or closed (0), used to quickly ascertain the status of financial items in accounts receivable.

Table 5.1: Detailed Descriptions of Invoice-Related Fields

Target Setting:

For this problem, the target variable, 'delay' is defined as the difference between the actual payment date and the scheduled due date of an invoice. This variable can assume various values: a negative value indicates an early payment, zero signifies on-time payment, and a positive value indicates a delay or late payment.

5.2.2 Exploratory Data Analysis

The object of the dataset used in this project is the Accounts Receivable from our client company for the fiscal years 2022 and 2023. Before proceeding with any data processing, conducting an exploratory data analysis (EDA) is crucial. This initial step allows us to understand underlying patterns, identify any inconsistencies, and assess the overall quality of the data. This understanding is essential for accurate analysis and effective decision-making in later stages.

For this purpose, I have conducted several visualizations to gain a deeper insight into the financial dynamics of our client interactions:

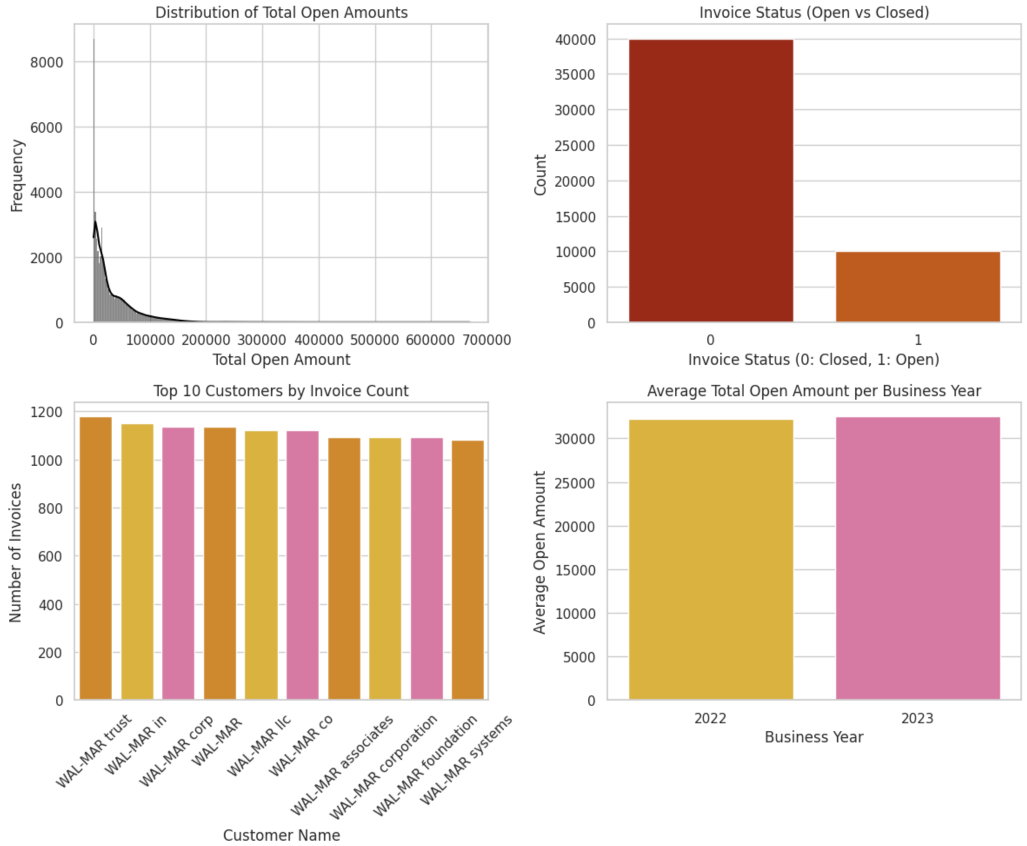


Figure 5.3: Invoice Distributions, Customer Activity, and Yearly Trends

Distribution of Total Open Amounts: This histogram shows the distribution of total open amounts across invoices, with the overlay of a kernel density estimate (KDE) providing a smooth curve of the distribution. The graph is heavily right-skewed, indicating that the majority of invoices are for smaller amounts, with a few invoices having very high amounts. This suggests that while the company deals with a high volume of transactions, most of these are for relatively small values. Given the right-skewness, applying a logarithmic transformation to the total open amounts could be beneficial.

Invoice Status (Open vs Closed): The bar chart displays the count of invoices based on their status, differentiating between closed (0) and open (1). There are significantly more closed invoices than open ones. For the purposes of data analysis, these open or still not paid invoices are eliminated from the training.

Top 10 Customers by Invoice Count: This bar chart ranks the top 10 customers by the number of invoices issued to each. The plot shows a relatively balanced distribution among the top 10 customers, although some variation in invoice count is evident.

Average Total Open Amount per Business Year: The bar chart shows the average open amount per invoice for the business years 2022 and 2023. There is an increase in the average invoice amount from 2022 to 2023.

Here is a scatter plot that visualizes the days difference between when invoices were due and when they were actually cleared:

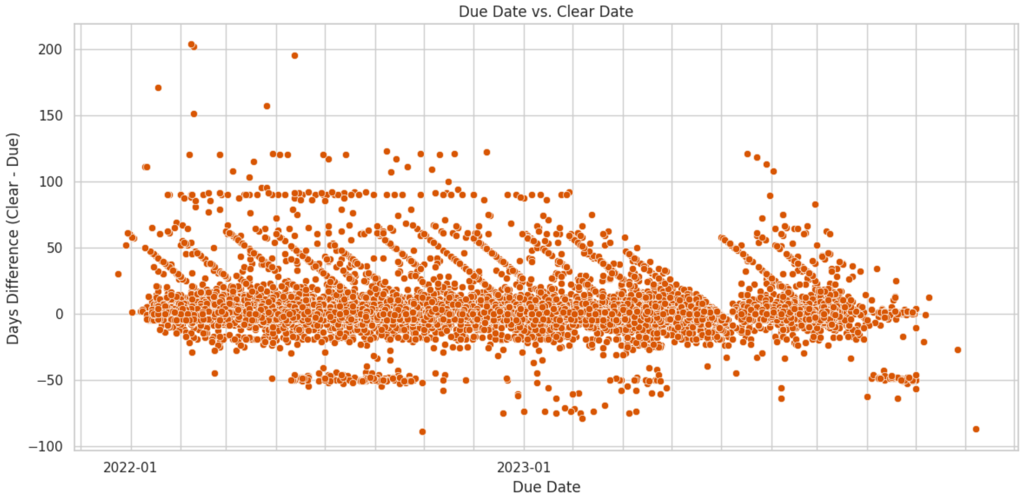


Figure 5.4: Due Date vs. Clear Date

Most data points are above the zero line, indicating that a majority of invoices were paid after their due dates, which suggests potential issues with late payments or extended credit terms. The cluster of points close to the zero line signifies that many invoices were paid close to their due dates, showing a degree of adherence to payment terms.

The goal of this project is to create a model that predicts whether invoices will be paid early, on time, or late, providing future insights on cash inflows.

5.2.3 Data Preparation

Preparing the data for the task at hand is one of the most important steps in any machine-learning project. It is also the step that required me the most time and effort.

When the dataset was first extracted, there was limited care when selecting which columns would be included. The aim was to include the biggest number of columns that could be interesting to the problem at first glance of their description and then analyze them one by one. Consequently, it was expected that some columns extracted would be redundant, other would be included merely for reporting purposes and some could be ill-maintained and thus wouldn't be interesting to the predictive model.

High Cardinality and Insignificant Features:

Features such as `posting_date`, `name_customer`, and `document type` were excluded due to their high cardinality, which complicates model training with many unique levels that do not substantially influence model decisions.

Handling Null Values and None Features: The `area_business` column was entirely composed of null values, as evidenced by the bar chart analysis, demonstrating a complete lack of usable data. Consequently, this column was removed from the dataset to streamline the data structure and enhance processing efficiency.

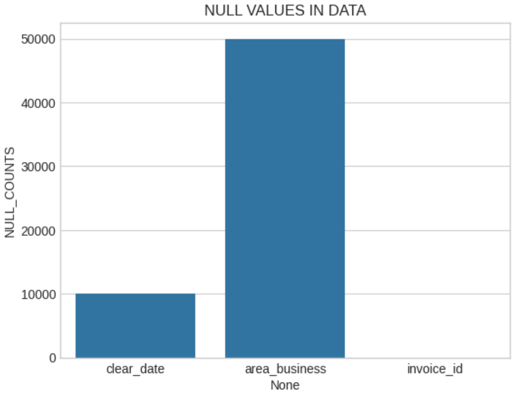


Figure 5.5: Null Values Bar Chart

The `clear_date` field, which indicates when an invoice was cleared, also contained some null values,they were segregated from the main training dataset.

Constant or near-constant features: Like `posting_id`, `business_year` and `isOpen` were also removed to enhance the model's performance. These are features that have the same value in nearly all rows, providing little to no informational value. The VarianceThreshold method was employed to identify such features. Columns with a variance below 0.01 were dropped from both datasets.

Duplicate rows were identified and removed from both the training and testing datasets.

Label Encoding for Categorical Variables: To handle categorical variables effectively, a custom label encoder class (LabelEncoderExt) was implemented. The following categorical columns were encoded: `buisness_year`, `cust_number`, `business_code`, `cust_payment_terms`

Feature Engineering and Generation

The last step of the data preparation stage is feature engineering. In machine-learning, models learn by finding patterns in their input data. That input is what practitioners call the features, an informative representation of the data in numerical form.

Several steps were taken to enhance the dataset by creating new features and refining existing ones to improve the predictive modeling process. The following outlines each step:

- **Date Conversion and Calculation:**

  The date columns (`clear_date`, `baseline_create_date`, and `due_in_date`) were converted to datetime format to facilitate date manipulations and calculations.

  New features were calculated to capture essential time intervals:

  ∗ `payment_term`: The number of days between `clear_date` and `baseline_create_date`.

  - `due_term`: The number of days between `due_in_date` and `baseline_create_date`.
  - `delay`: The difference between `payment_term` and `due_term`, representing the delay in payment and which is going to be our target variable.

- **Date Component Extraction:**
  - **Date Components:** For `baseline_create_date` and `due_in_date`, additional features were created to capture the day, month, and year components, as well as the day of the week.

- **Aggregation by Customer:**
  - **Sum and Mean Calculations:** The dataset was grouped by `cust_number` to calculate the sum and mean of numeric columns (`total_open_amount`, `due_term`), resulting in features such as `Sum_base_amount`, `Sum_due_term`, `mean_base_amount`, and `mean_due_term`.
  - **Invoice Count and Average Delay:** The number of invoices per customer and the average payment delay were also computed.

- **Feature Generation:** generating new features is crucial in machine learning model development because it significantly enhances model performance by providing additional information that captures more nuances in the data, thus improving predictive power and reducing bias and variance. New features can reveal hidden patterns and relationships within the data, leveraging domain-specific knowledge to make the model more informative and interpretable.

  New features were derived by calculating ratios such as:

  - **amount/mean_amount**: this feature is the ratio of the `total_open_amount` to the `mean_base_amount`. It measures how the amount of a specific invoice compares to the average amount of all invoices for the same customer.
  - **amount-/mean_amount**: this feature is the normalized difference between the `total_open_amount` and the `mean_base_amount`, divided by the `mean_base_amount`. It measures the deviation of the invoice amount from the customer's average invoice amount, relative to that average.
  - **due_term/amount**: this feature is the ratio of the `due_term` (the number of days between the baseline create date and the due date) to the `total_open_amount`. It measures the time allowed for payment relative to the invoice amount.
  - **mean_due_term/amount**: this feature is the ratio of the `mean_due_term` (the average due term for a customer) to the `total_open_amount`. It compares the average time allowed for payment across all invoices of a customer to the specific invoice amount.
  - **mean_due_term/Sum_base_amount**: this feature is the ratio of the `mean_due_term` to the `Sum_base_amount` (total sum of invoice amounts for a customer). It measures the average due term relative to the total invoiced amount for a customer.
  - **cust_count/mean_amount**: this feature is the ratio of the number of invoices (`cust_count`) for a customer to the `mean_base_amount`. It measures the frequency of invoicing relative to the average invoice amount.
  - **cust_count*due_term/amount**: this feature is the product of the `cust_count` and the `due_term`, divided by the `total_open_amount`. It measures the cumulative credit period for a customer relative to the invoice amount.

  Features representing the count of transactions per customer (`cust_count`) and the ratio of `cust_count` to `mean_base_amount` were added.

- **Normalizing and Scaling Data:**

Log Transformation: The `total_open_amount` was log-transformed to normalize its distribution because its distribution was heavy-tailed and so, not so convenient to provide information to most models.

$$v' = \log(v) \tag{5.1}$$

Scaling: Numerical features are scaled using `MinMaxScaler` to ensure all features are on the same scale. Scaling is important in this problem because some algorithms that use weights or distance measures such as regression are adversely affected by differences in variable ranges and can be skewed towards features with greater ranges.

$$v' = \frac{v - \text{mean}}{\text{variance}} \tag{5.2}$$

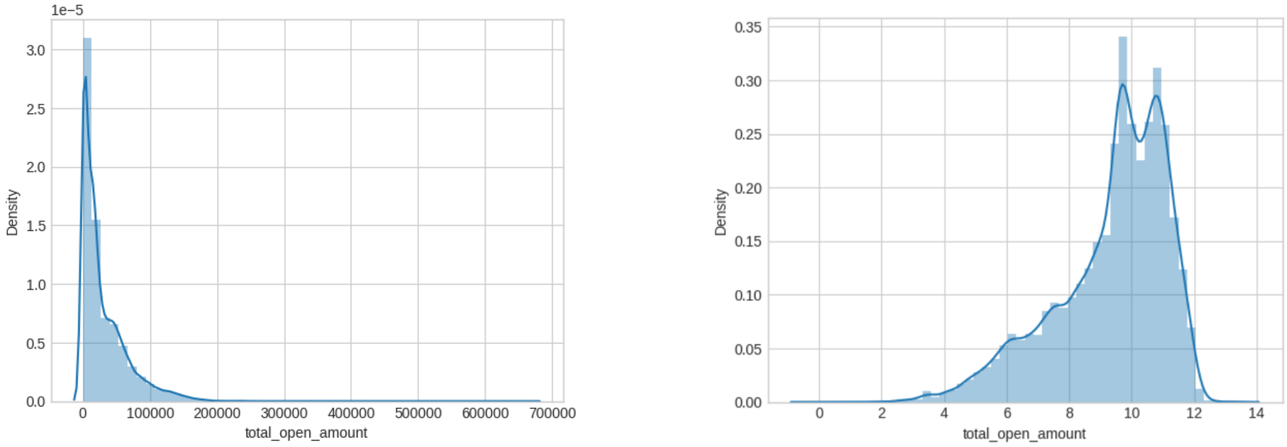


Figure 5.6: Distribution of the Total Open Amount before and after the Log Transformation

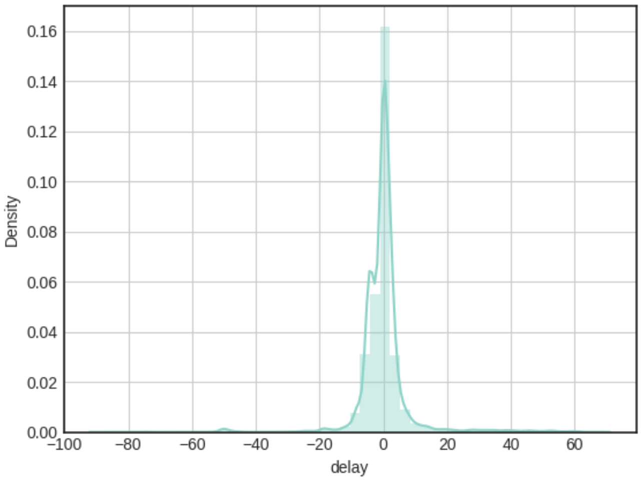


Figure 5.7: Distribution of the target variable 'delay'

The distribution of the 'delay' variable, as depicted in the plot, primarily centers around zero and spans from -15 to 20 days, indicating that the majority of invoices are settled shortly before or just after their due date. This concentration suggests a general adherence to payment terms with most transactions occurring within a reasonably tight timeframe around the scheduled dates. The presence of values outside this range, although significantly fewer, highlights occasional deviations where some invoices are settled much earlier or later than expected. This pattern underscores the variability in payment behaviors, with most customers demonstrating timely compliance, while a minority contribute to the tails of the distribution, reflecting early settlements or delayed payments.

5.2.4 Modeling

5.2.4.1 Train-Test Split:

After an extensive data cleaning process, which included the transformation of variables and the generation of relevant features, we conducted scaling on the numerical features and applied one-hot encoding to the categorical variables. This preprocessing resulted in a refined dataset split into two distinct sets:

- **Training Dataset:** Used for learning the model.
- **Testing Dataset:** Evaluates the final model performance against completely unseen data to estimate real-world performance.

Table 5.2: Summary of Train-Validation-Test Split

Dataset	Number of Rows (Invoices)	Number of Columns	Percentage of Total
train	27,171	92	75.0%
test	9,059	92	25.0%

5.2.4.2 Modeling using Machine Learning Algorithms

The task at hand involves predicting the `delay` variable associated with invoice payments. The `delay` is quantified as the number of days an invoice is settled relative to its *due date*. A positive value indicates a delay where the invoice is paid after the due date, a negative value signifies early payment before the due date, and a value of zero means the invoice is paid exactly on the due date.

Accurate prediction of the `delay` is crucial as it directly informs the calculation of the *invoice clear date*. This date is determined by adding the predicted `delay` to the original *due date* of the invoice.

The following supervised learning algorithms were utilized to solve this regression problem:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Support Vector Regression
- Random Forest Regression
- Decision Tree Regression
- XGBoost Regression

Justification for the Choice of Algorithms:

The selection of algorithms for predicting the `delay` variable in invoice payments was strategically made to cover a broad spectrum of modeling techniques, each bringing unique strengths and capabilities for addressing different aspects of regression analysis.

- **Linear Regression** was chosen for its simplicity and high interpretability, serving as a baseline model that evaluates linear relationships between the features and the target variable. Extensions of this model, **Ridge Regression** and **Lasso Regression**, incorporate regularization to penalize large coefficients, thus reducing the risk of overfitting and enhancing the model's ability to generalize. Ridge is particularly effective in datasets with multicollinearity, whereas Lasso aids in feature selection by shrinking less important coefficients to zero.

- For capturing complex, non-linear patterns and interactions between variables, **Random Forest Regression** and **Decision Tree Regression** are utilized. These models do not require feature scaling and offer intuitive mechanics. Random Forest, as an ensemble of decision trees, generally delivers higher accuracy and stability by averaging multiple deep decision trees, thereby minimizing variance.
- Lastly, **XGBoost Regression** leverages the power of gradient boosting frameworks known for their predictive efficiency and processing speed, especially in structured data scenarios. XGBoost offers systematic treatment of missing values, includes regularization to deter overfitting, and is highly scalable, making it an optimal choice for large datasets.

Collectively, these algorithms enable a comprehensive exploration of the dataset from multiple perspectives, enhancing the robustness and reliability of the predictions made for payment delays.

5.2.4.3 Models Tuning

Fine-tuning hyperparameters is a critical step that can significantly impact performance. For this project, I employed two distinct methods to optimize the models hyperparameters: Grid Search and Bayesian Optimization.

Grid Search Method

Grid Search was utilized for most of the models, including Linear Regression, Lasso Regression, and Decision Tree Regression. This method involves defining a grid of hyperparameter values and evaluating every combination of these parameters to determine which configuration yields the best performance based on a predefined scoring metric.

Bayesian Optimization

For the Random Forest and XGBoost algorithms, Bayesian Optimization was chosen due to the complexity and the large number of hyperparameters involved, such as `n_estimators` (the number of trees in the forest), `max_depth` (maximum depth of each tree), `min_samples_split` (minimum number of samples required to split an internal node), and `min_samples_leaf` (minimum number of samples required to be at a leaf node). These parameters critically influence the behavior and performance of the Random Forest, making the model highly sensitive to their settings.

Bayesian Optimization seeks to minimize the number of evaluations needed by constructing a probabilistic model mapping hyperparameters to a probability of a score on the objective function. This method then uses this model to make informed decisions about which hyperparameters are likely to improve model performance, focusing computational resources on testing those that offer the most promise.

5.2.4.4 Results Analysis

After deploying the different regression models on our dataset, I utilized the Mean Squared Error (MSE), RMSE and the Coefficient of Determination (R^2 Score) to assess the efficacy of each model. Here are the computed values for each respective model:

Table 5.3: Evaluation Results of Regression Models Performance

Algorithm	MSE Score	RMSE Score	R2 Score
LR	0.0012	0.0346	0.5781
Ridge	0.0012	0.0347	0.5758
Lasso	0.0013	0.0360	0.5437
RF	0.0007	0.0263	0.7568
XGB	0.0016	0.0395	0.5960
SVR	0.0026	0.0506	0.1010
DT	0.0020	0.0446	0.4863

The following scatter plots show the actual vs. predicted values for each regression model, providing a visual representation of their performance.

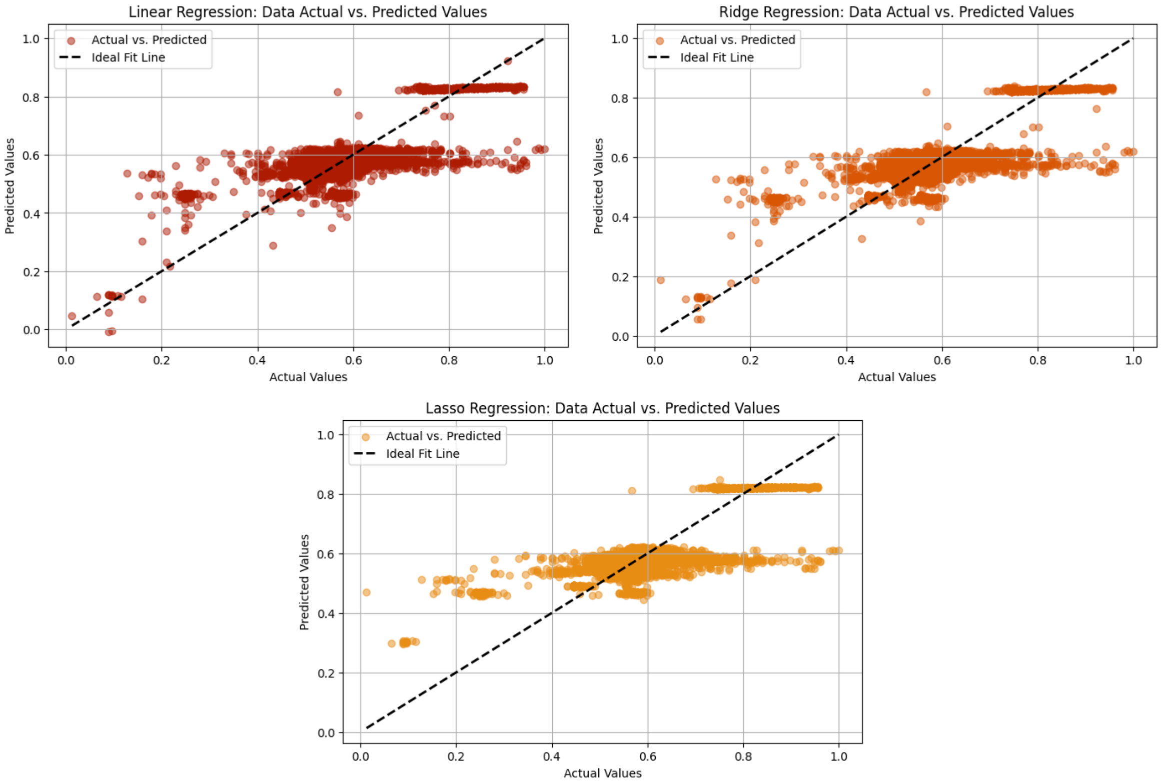


Figure 5.8: Actual vs. Predicted Values: Linear, Ridge, Lasso Regression

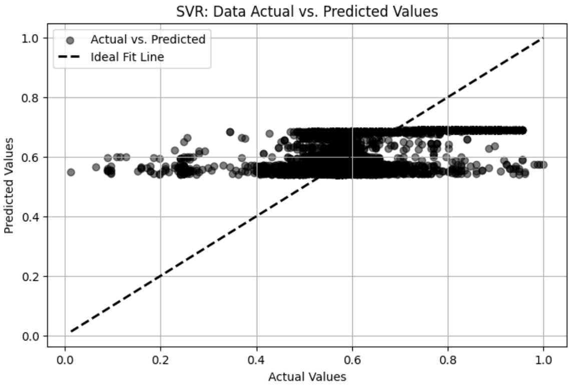


Figure 5.9: Actual vs. Predicted Values for SVR

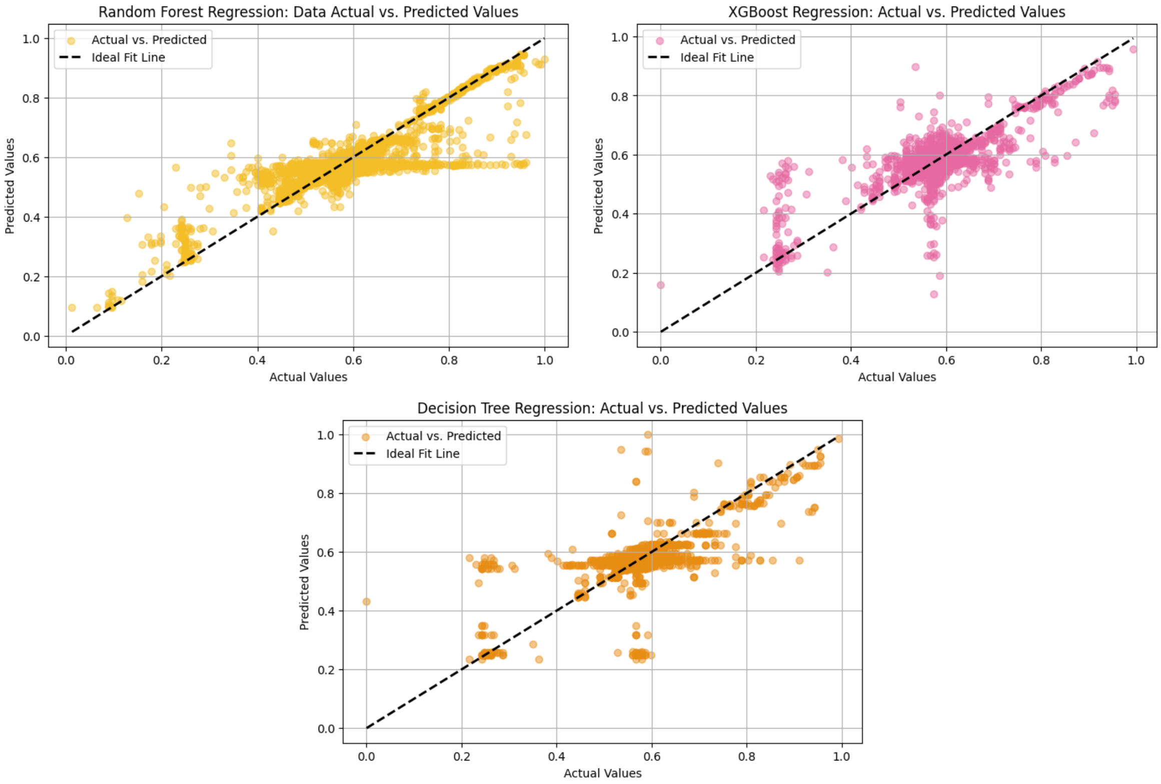


Figure 5.10: Actual vs. Predicted Values: Random Forest, XGBoost, Decision Tree Regression

The evaluation results indicate that Random Forest Regression performs the best among all models, achieving the lowest MSE of 0.0007, and the highest R2 score of 0.7568, signifying a strong ability to explain the variance in the data. Linear Regression, Ridge Regression, and Lasso Regression exhibit similar performances with moderate MSE and RMSE values, and R2 scores around 0.54-0.58, indicating a decent fit. XGBoost Regression also shows good performance, with slightly higher MSE and RMSE than RF but a respectable R2 score of 0.5960. Support Vector Regression and Decision Tree Regression perform poorly, with SVR showing the highest MSE (0.0026), and a very low R2 score of 0.1010, indicating it explains only 10% of the variance. Decision Tree Regression has a moderate performance with an MSE of 0.0020, and an R2 score of 0.4863.

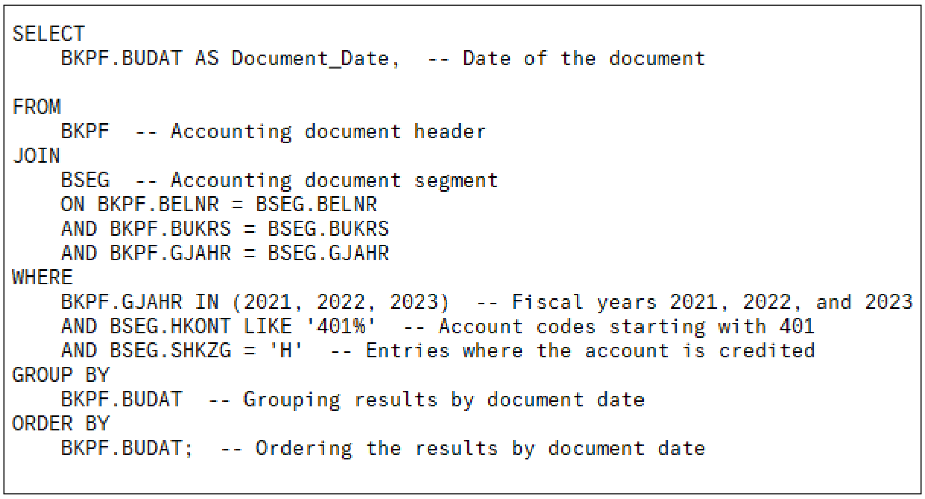
Overall, ensemble methods like Random Forest and XGBoost outperform the other models due to their ability to combine multiple weak learners to form a robust model that captures complex patterns and interactions in the data. Random Forest leverages the power of averaging multiple decision trees to reduce variance and overfitting, while XGBoost uses gradient boosting to iteratively improve model performance by focusing on the errors of previous iterations. These methods are particularly effective in handling non-linearity and high-dimensional data, leading to better predictive accuracy and reliability. In contrast, linear models and support vector regression, which assume a linear relationship between features and the target variable, are less capable of capturing intricate patterns, resulting in weaker performance.

5.3 Accounts Payable Forecasting

5.3.1 Data Collection

In the data collection process from the SAP HANA database, I employed SQL queries to retrieve all accounting entries for the years 2021 through 2023. I joined necessary tables to access comprehensive data on financial transactions. From the extracted fields, I retained only the *Document_Date* (the date when the Account Payable was recorded) and the *Accumulated_Accounts_Payable* for final analysis.

I specifically filtered the Account Codes to include only those beginning with “401,” which correspond to supplier accounts in French accounting standards, reflecting that supplier accounts are typically credited in accounts payable transactions. Although the Account Code was crucial for filtering the data, it was not included in the final aggregated output. I then aggregated the data by the *Document Date*, and summed the amounts of accounts payable for the same date to provide a consolidated view of liabilities incurred on each date.



```
SELECT
    BKPF.BUDAT AS Document_Date,  -- Date of the document

FROM
    BKPF  -- Accounting document header
JOIN
    BSEG  -- Accounting document segment
    ON BKPF.BELNR = BSEG.BELNR
    AND BKPF.BUKRS = BSEG.BUKRS
    AND BKPF.GJAHR = BSEG.GJAHR
WHERE
    BKPF.GJAHR IN (2021, 2022, 2023)  -- Fiscal years 2021, 2022, and 2023
    AND BSEG.HKONT LIKE '401%'  -- Account codes starting with 401
    AND BSEG.SHKZG = 'H'  -- Entries where the account is credited
GROUP BY
    BKPF.BUDAT  -- Grouping results by document date
ORDER BY
    BKPF.BUDAT;  -- Ordering the results by document date
```

Figure 5.11: SQL Query for Retrieving Accounts Payable Data from SAP HANA

where:

- The `BKPF` and `BSEG` tables are utilized, where `BKPF` contains the document header data, and `BSEG` provides line item details.
- Fields such as `BELNR`, `BUKRS`, and `GJAHR` link both tables and represent the document number, company code, and fiscal year, respectively.
- The field `BSEG.WRBTR` represents the amount in the document currency.
- The condition `BSEG.SHKZG = ’H’` ensures that only credit transactions are included, aligning with how accounts payable are typically recorded by crediting the supplier’s account.
- The query aggregates the credit amounts per document date, providing a daily summarized view of the financial obligations towards suppliers.

At the end of the data collection process, the resulting dataset is meticulously organized into two primary columns, spanning the years 2021, 2022, and 2023. These columns include:

- **Document Date:** This column has been strategically configured as the index using the `set_index` function. This arrangement facilitates an efficient time-series analysis by aligning data points in accordance with the respective transaction dates, enhancing the accessibility and readability of the time-related financial data.
- **Accumulated Accounts Payable:** This column aggregates the summed amounts of accounts payable, compiled by each document date. It offers a consolidated and comprehensive view of the financial liabilities accrued on each specific day, providing a clear depiction of the company's financial obligations over time.

This structured dataset serves as the critical foundation for our subsequent time-series modeling endeavors. Utilizing this dataset, we aim to analyze and predict financial trends based on the historical payment behaviors captured within the data, employing sophisticated modeling techniques to forecast future financial liabilities effectively.

5.3.2 Data Understanding

Following the data collection process, I obtained a dataset comprising two columns: "Document Date" and "Accumulated Accounts Payable" amounts by date. To facilitate a preliminary analysis of the time series data and to understand its underlying components more effectively, I first converted the "Document Date" column into a datetime format. This step was crucial for time series analysis, allowing me to set this column as the index of the dataset. Subsequently, I utilized the `Matplotlib` library to visualize the time series, providing an initial graphical representation of the data. This visualization is instrumental in revealing patterns, trends, and cyclical behavior within the Accounts Payable data over time, offering valuable insights even before deep diving into more complex analyses.

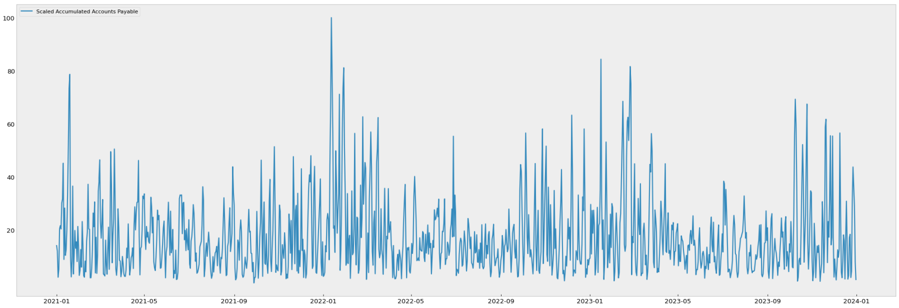


Figure 5.12: Accumulated Accounts Payable Distribution(2021-2023)

This plot offers several insights into the financial obligations and expenditure patterns of the firm:

Volatility and Seasonality: The time series exhibits significant volatility, with frequent and pronounced spikes in accounts payable amounts. This suggests periods of heavy expenditure, likely correlated with operational cycles and strategic purchasing decisions. The repeated patterns of peaks suggest a potential seasonality in the accounts payable, which may align with specific times of the year where purchasing activity is intensified, such as seasonal inventory restocking and large scale procurement for projects.

Annual Trends: Each year shows variability in the total amounts and the distribution of accounts payable. For instance, high peaks often occur around the same time each year, indicating annual cycles in business spending. The beginning of each year (January) and mid-year typically show higher payable amounts, which could align with the financial closing of quarters and half-year financial planning cycles.

This analysis indicates that the firm has a dynamic and active approach to managing its payables, with clear cyclic patterns that likely reflect its operational and strategic financial rhythms.

5.3.2.1 Time Series Decomposition

To further understand the components of the accumulated accounts payable data, I performed a time series decomposition. This technique separates the series into trend, seasonal, and residual components, providing a clearer view of the underlying patterns.

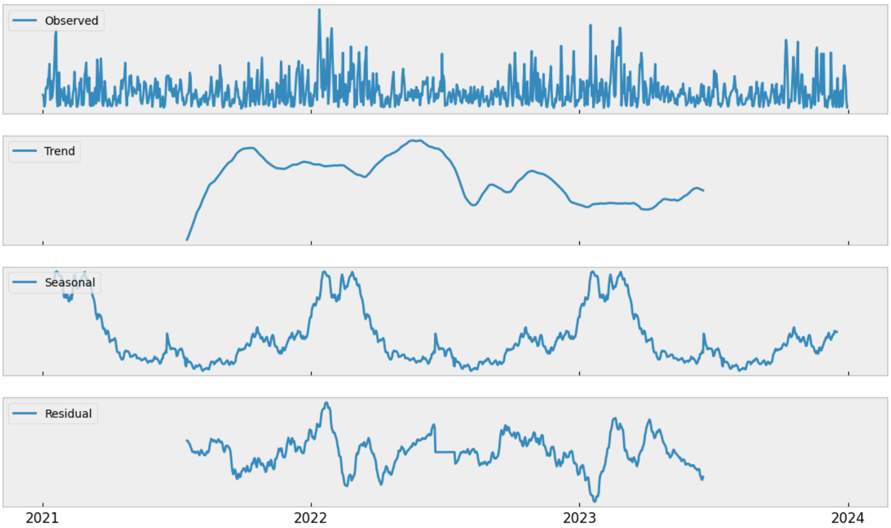


Figure 5.13: Time Series Decomposition

The presence of significant seasonal patterns suggests that the firm's Accounts Payable amounts follow a predictable cycle, likely linked to its operational and financial planning. The peaks observed in the seasonal component may correspond to strategic periods where the firm engages in heavy purchasing activities, such as preparing for high-demand seasons or executing large-scale projects.

The trend component highlights a period of increasing payables, followed by a stabilization phase.

Understanding these patterns allows the firm to better anticipate its cash outflow requirements. By recognizing the seasonal peaks and planning accordingly, the firm can ensure sufficient liquidity during high expenditure periods. The trend analysis also provides insight into long-term financial obligations, aiding in strategic financial planning.

The residuals indicate that while most of our firm's payables can be explained by the identified trends and seasonality, there is still some level of unpredictability.

5.3.2.2 White Noise Check

White noise is a key concept in time series analysis, representing a random signal having equal intensity at different frequencies, which gives it a constant power spectrum. A time series that is white noise would not have any predictable patterns or structures, making predictions or further analyses unreliable.

Before proceeding with further analysis and forecasting on the time series, I initially applied the Ljung-Box test. This test specifically examines whether any of the autocorrelations in a group within the time series differ from zero. Essentially, it tests the null hypothesis that the data points are distributed independently, displaying no serial correlation—a key trait typically associated with white noise.

The Ljung-Box test results yield a test statistic (`lb_stat`) of 698.272762 and a p-value (`lb_pvalue`) of approximately $1.474875 \times 10^{-143}$. These values carry significant implications for our time series analysis.

Interpretation of the Test Statistic and P-Value:

- **Test Statistic:** The value of 698.272762 is a measure of the overall evidence against the null hypothesis that suggests no autocorrelations are present in the time series. A larger test statistic indicates stronger evidence against the null hypothesis.
- **P-Value:** The p-value is extremely small, practically zero for all intents and purposes. This indicates that the probability of observing such a strong test statistic under the null hypothesis (that the time series data points are independently distributed with no serial correlation) is extremely low.

Given the very low p-value, we decisively reject the null hypothesis. This rejection suggests that there is substantial statistical evidence that the time series exhibits significant autocorrelation at one or more of the first 10 lags. Therefore, the time series data are not random ’white noise’; instead, they display patterns or relationships that persist over time.

5.3.2.3 Stationarity Check

A stationary time series is one whose properties do not depend on the time at which the series is observed. This means that the mean, variance, and covariance of the series do not change over time. Stationarity is important because most time series models assume that the underlying data is stationary. Non-stationary data can lead to misleading statistics and unreliable results.

Verifying that the time series is stationary is critical before applying statistical models because models like ARIMA require the input data to be stationary. If the data is non-stationary, it needs to be transformed, often by differencing, to make it stationary.

When analyzing the stationarity of this time series, the Augmented Dickey-Fuller (ADF) test is employed. For our dataset, the ADF test provided the following results:

Table 5.4: ADF Test Results

Metric	Value
ADF Statistic	-10.1167
p-value	9.6809×10^{-18}

Interpretation of ADF Test Results:

- **ADF Statistic:** The ADF statistic, being a highly negative number, provides strong evidence against the null hypothesis, which states that the series has a unit root and is non-stationary.
- **p-value:** The extremely small p-value indicates a very low probability that such a strong stationary result would be seen if the null hypothesis were true.

Given the highly negative ADF statistic and the very small p-value, we decisively reject the null hypothesis of the presence of a unit root. This rejection indicates that the time series is stationary. Establishing stationarity is vital as many time series forecasting methods require the data to be stationary to ensure reliable predictions.

5.3.3 Forecasting Models Application

After confirming the stationarity of the dataset, we are well-positioned to apply various statistical and machine learning models for forecasting. Initially, the dataset was divided into training and testing subsets to facilitate effective model validation. For each forecasting model, appropriate model fitting and visualization techniques were employed to ensure accurate representation and analysis of the data's future behavior.

The test dataset comprises 20 percent of the entire dataset, which encompasses daily data spanning three years. Given this setup, the test dataset includes the final 219 days of the year 2023, representing the last portion of the data for that year.

5.3.3.1 ARIMA

The ARIMA model is characterized by three parameters: p, d, and q. These parameters are crucial for the model's structure:

- p: Represents the number of lag observations included in the model, also known as the lag order. This parameter helps capture the autocorrelation in the AR (autoregressive) part of the model.
- d: Denotes the degree of differencing required to make the series stationary. Since our data has been verified to be stationary, we do not need to apply differencing; thus, $d = 0$.
- q: Indicates the size of the moving average window, which is the order of the MA (moving average) part of the model. This parameter helps capture the lingering effects of past forecast errors in the prediction equation.

Parameter Determination Using PACF and ACF:

To accurately select the values for p and q, we utilize the PACF (Partial Autocorrelation Function) and ACF (Autocorrelation Function) respectively:

PACF for determining p:

The PACF plot is used to identify the extent of direct effect past data points have on the current data. To determine the appropriate p value, we look for the point at which the PACF plot cuts off (i.e., where the partial autocorrelations become statistically insignificant beyond a certain lag). This cutoff point suggests the optimal number of lags that should be used in the AR portion of the model.

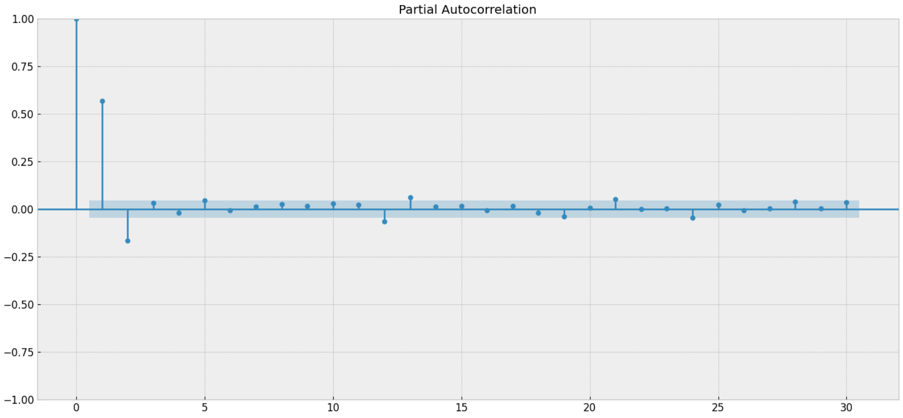


Figure 5.14: PACF Plot

Based on the PACF plot analysis, the value of p should be set to 1 for the ARIMA model parameterization. This determination is grounded in the significant autocorrelation at the first lag, which suggests that including one past term in the autoregressive part of the model will sufficiently capture the autocorrelation structure of the data.

ACF for determining q:

The ACF plot shows the correlation between the series and its lags. For identifying the q value, we examine the ACF plot to find out where the autocorrelations drop off sharply, which can be indicated by the bars becoming small and insignificant after a certain lag. This indicates the maximum lag after which the MA parameters can effectively capture the autocorrelation in the residuals.

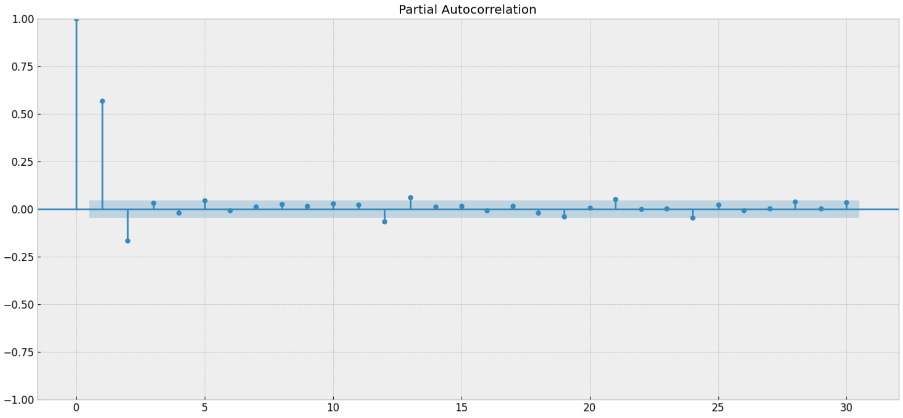


Figure 5.15: ACF Plot

Given the sharp drop in autocorrelation after the first lag and considering that the autocorrelation at lag 1 is distinctly positive and above the usual significance line, the optimal q value for the ARIMA model would be set to 1.

After determining the appropriate values for the parameters p, q, and d, I proceeded to train the ARIMA model using the training dataset. Below is a visualization that compares the predicted data from the model against the actual test data.

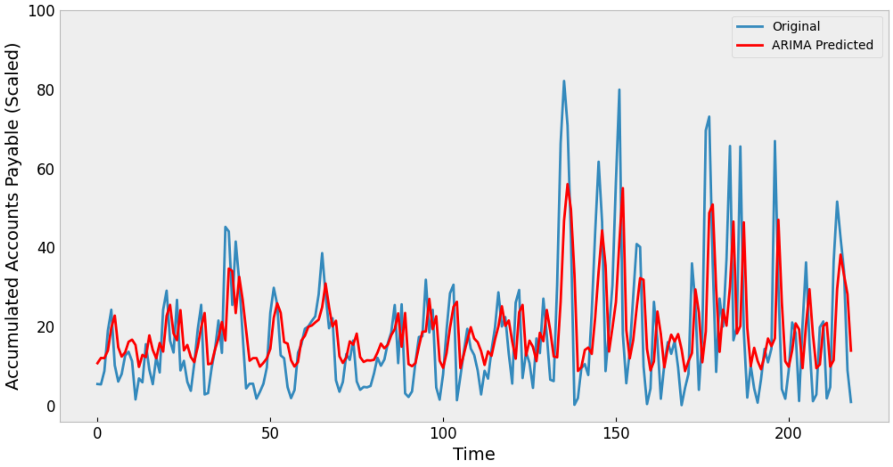


Figure 5.16: Comparison of Original and ARIMA Predicted Values

The ARIMA model appears to follow the general trend of the original data closely, suggesting it has captured the underlying trend effectively.

The model seems to somewhat capture the seasonality, as indicated by the alignment of peaks and troughs in both the predicted and original lines. However, it may not capture all seasonal variations perfectly, as there are instances where the predicted values diverge from actual spikes or drops.

5.3.3.2 SARIMA

Unlike ARIMA model, which is primarily focused on identifying and modeling trend components, the SARIMA model adds another layer of sophistication with four crucial seasonal parameters. These parameters are designed to capture seasonal variations more precisely, thereby significantly enhancing the model's predictive accuracy for time series data characterized by distinct periodic patterns. The optimal configuration of these seasonal parameters—namely P, D, Q, and m—was rigorously established through a methodical process of cross-validation. This systematic approach helps ensure that the SARIMA model is exquisitely tuned to accurately reflect both the underlying trends and the seasonal fluctuations present in the dataset, providing a solid foundation for reliable and robust forecasting.

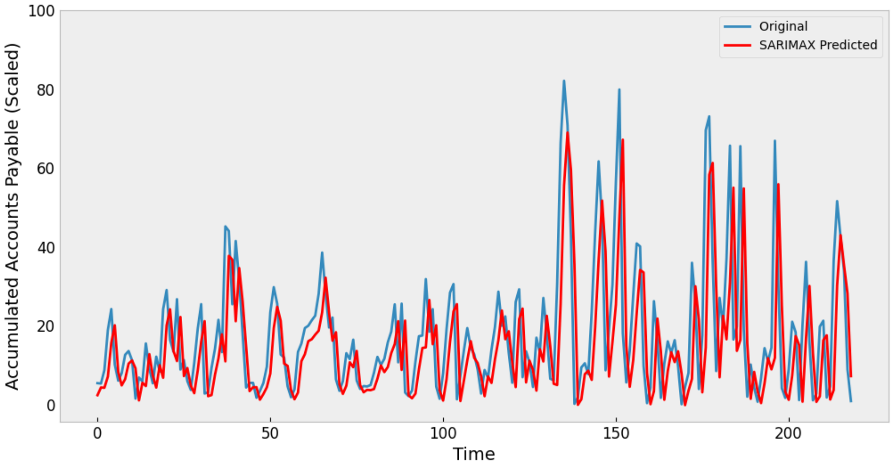


Figure 5.17: Comparison of Original and SARIMA Predicted Values

SARIMA model appears to generally follow the trend and seasonal patterns of the original data quite closely. This indicates that the model has successfully captured the primary dynamics of the dataset.

The model effectively predicts the direction and timing of peaks and troughs, although there are variances in amplitude. This suggests that while the model can forecast general trends and cyclic behavior accurately, it may slightly under or overestimate the actual values at times.

The close tracking between the predicted and actual values for most of the series indicates a strong model fit. This suggests that the SARIMA model parameters are well-tuned for capturing the essential characteristics of this time series.

5.3.3.3 Prophet

In setting up the Prophet model for forecasting, several key configuration steps were undertaken to tailor the model specifically to the nuances of the dataset. First, the data was reformatted to meet Prophet's requirements, with the time index reassigned to a column named 'ds' and the target variable 'Accumulated Accounts Payable' renamed to 'y'. This prepares the data for processing by Prophet, which expects these specific column names.

The Prophet model was then initialized with a linear growth assumption and a multiplicative approach to modeling seasonal variations. This setup suggests an expectation of steady growth over time with seasonal fluctuations that scale proportionally with the trend. Various custom seasonalities were defined explicitly, including daily, weekly, monthly, quarterly, and yearly patterns, each with specified periods and Fourier orders. This detailed setup allows the model to capture complex seasonal behaviors accurately.

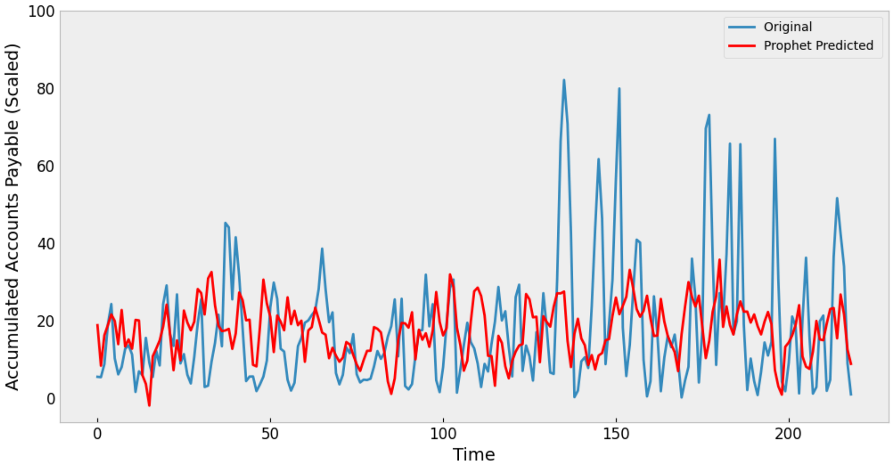


Figure 5.18: Comparison of Original and Prophet Predicted Values

While the Prophet model effectively captures general trends and seasonal patterns in the "Accumulated Accounts Payable" data, it appears to underperform compared to SARIMA and ARIMA models, particularly in accurately forecasting sharp peaks and sudden fluctuations. This limitation likely stems from Prophet's simpler handling of trend and seasonality, which might not capture the complex, non-linear dynamics as effectively as ARIMA-type models with their intricate differencing and error correction mechanisms. Additionally, Prophet's predictions occasionally overshoot or undershoot critical values.

5.3.3.4 XGBoost

Using XGBoost for the time series forecasting requires transforming our univariate time series into a multivariate format to effectively leverage the model's capabilities. XGBoost, is not inherently designed to handle the sequential data typically found in time series. This transformation is crucial for capturing the inherent patterns within the data effectively.

I began by engineering features from the time series data to create a rich dataset that could reveal underlying temporal dynamics:

- **Date Decomposition:** Components such as day of the week, quarter, month, and year were extracted from the datetime index of the dataframe. These components help capture intra-day, weekly, quarterly, monthly, and annual patterns in the data.
- **Additional Cyclical Features:** Day of the year, day of the month, and week of the year were included to capture other potentially relevant cyclical behaviors which might influence the time series' characteristics.
- **Trigonometric Features:** Sinusoidal transformations (sine and cosine) of the 'day of year' were calculated to capture the smooth cyclical pattern through the year, aiding in modeling the seasonality effectively.

After extracting these datetime components, the original 'date' column was dropped from the dataset. The remaining features were assembled into a new feature matrix. This matrix now represents the transformed multivariate dataset where each feature provides specific insights into the time dynamics of the series.

Following these preparatory steps, the XGBoost model was applied. The performance of the model can be visualized in the graphs below.

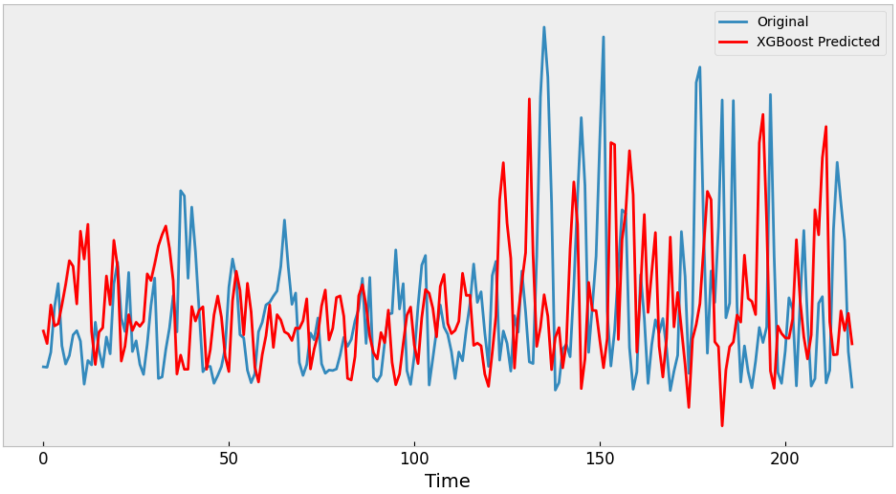


Figure 5.19: Comparison of Original and XGBoost Predicted Values

Despite the model's ability to follow the trend, there are notable deviations where the predicted values either overshoot or undershoot significantly, especially at peaks and troughs.

To gain a deeper understanding of the XGBoost model's performance, let's examine the scatter plot provided below.

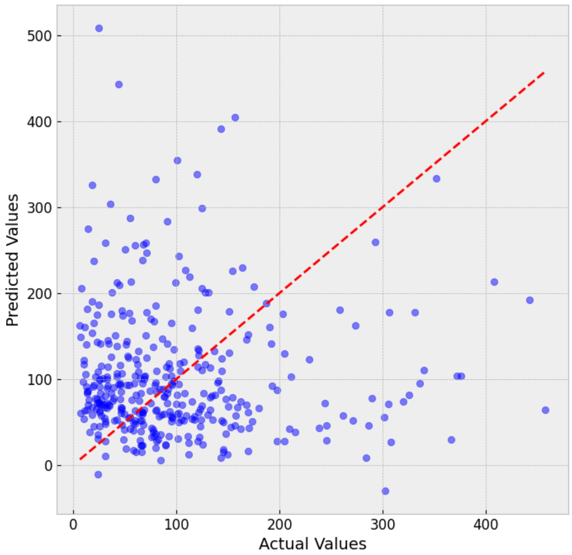


Figure 5.20: XGBoost Model Predictions Versus Actual Data

The plot shows a considerable number of predictions scattered around but not consistently along the red dashed line of perfect prediction. This indicates that while the model can predict general trends, it does not consistently predict accurate values for all data points. The dense clustering of points around the lower end of the scale suggests that the model is more accurate at predicting smaller values. As the actual values increase, the predictions become more dispersed and less accurate.

5.3.4 Models Performance Evaluation

After deploying the four models, I evaluated their performance using the Mean Squared Error (MSE) and the Coefficient of Determination (R^2 Score), and RMSE metrics. The results are as follows:

Table 5.5: Forecasting Models Evaluation Results

Algorithm	MSE Score	RMSE Score	R2 Score
ARIMA	4407.4087	66.3880	0.3356
SARIMA	454.9450	21.3286	0.6930
Prophet	6938.0885	83.2952	-0.0460
XGBoost	1248.1091	35.3286	0.5130

The results indicate that the SARIMA model performed the best in terms of the metrics used, with the highest R2 score and the lowest MSE and RMSE values, followed by the XGBoost model. This suggests that the SARIMA model is particularly well-suited to capturing both the seasonal and non-seasonal patterns present in the data. The XGBoost model, known for its robustness and ability to handle various data patterns, also showed strong performance but didn't quite match the SARIMA model's precision. The ARIMA model, which doesn't account for seasonality, had moderate performance, reflecting its limitations in handling the inherent seasonal variations in the dataset. The Prophet model, while designed to handle time series data, showed the least predictive accuracy, possibly due to its reliance on more generalized assumptions that didn't align as well with the specific characteristics of the dataset used in this analysis.

5.3.5 Predicting Future Accounts Payable Using SARIMA

In this section, we focus on using the SARIMA model to predict future accounts payable for the year 2024. SARIMA was chosen due to its superior performance compared to other models, as demonstrated by its higher R2 score and lower MSE and RMSE values.

The SARIMA model was applied to predict the accounts payable amounts for the year 2024. As seen in the plot below, the model captures the overall trend and seasonal variations of the accounts payable effectively.

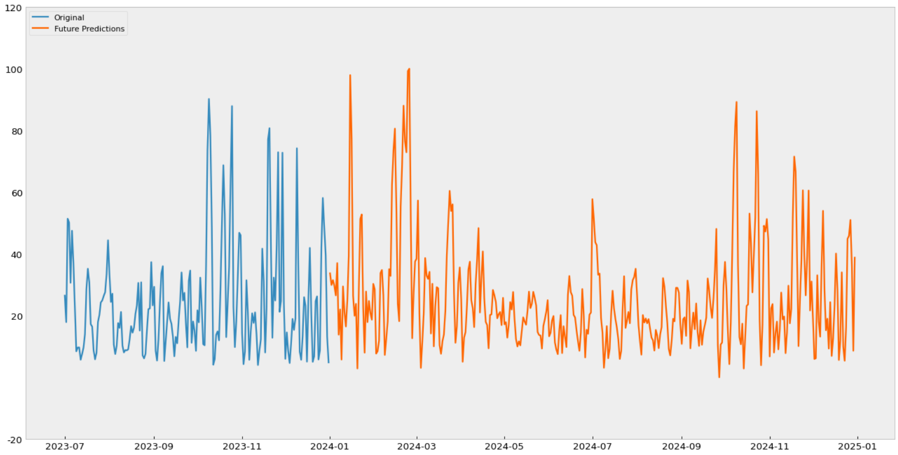


Figure 5.21: SARIMA Future Predictions

The plot demonstrates that the SARIMA model successfully identifies the periodic spikes and fluctuations in the accounts payable amounts, which are influenced by the firm's operational cycles and strategic financial activities.

To better visualize the future trend, we aggregated the predicted accounts payable sums by month and scaled the values to a range of 0 to 100 (for confidentiality issues). The bar plot below represents the monthly sums for 2024:

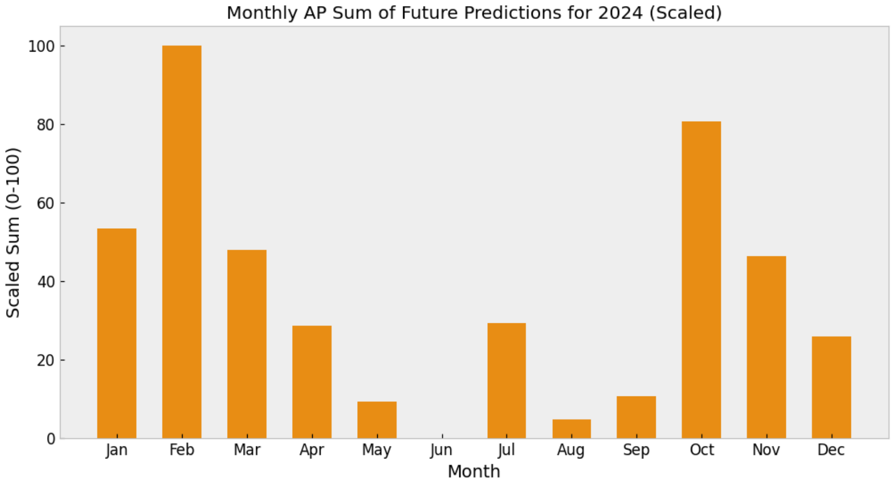


Figure 5.22: Monthly Sum of Future Predictions for 2024 (Scaled)

The bar plot of the scaled monthly sum of future predictions for accounts payable in 2024 indicates distinct peaks in January, February, and October, suggesting significant procurement or financial activities during these months. January and February likely reflect post-holiday restocking or annual contract renewals, while the October peak could be due to preparations for the end-of-year demands. Lower values in May and August might indicate periods of reduced operational activity, possibly due to seasonal lulls or fewer procurement needs. The moderate values observed in March, April, June, and November suggest steady ongoing expenses, ensuring consistent operational functionality.

5.4 Conclusion

In this chapter, I developed in detail the various aspects of my proposed solution, which involves the creation of two predictive models to enhance cash flow management and gain future insights into its trends. Specifically, I constructed a regression model aimed at predicting accounts receivable payment dates. This model enables cash managers to forecast incoming cash flows, anticipate whether customers will pay on time or delay payments, and take proactive measures to prevent late payment penalties. This also facilitates the optimization of the collection processes.

The second component of the proposed solution involves employing time series modeling and forecasting to predict future liabilities and expenditures, specifically accounts payable. This approach aids in effective cash flow management by ensuring the necessary cash reserves are allocated appropriately, thus preventing potential cash shortages. By accurately forecasting these future financial obligations, the company can maintain liquidity, plan better for future expenditures, and ensure operational continuity without the risk of financial shortfalls. This dual-model strategy significantly improves the overall efficiency and reliability of cash flow management processes.

General Conclusion

In conclusion, this thesis highlights how optimizing cash flow management using advanced predictive statistical techniques, including machine learning and time series forecasting models, can provide future insights that enable managers to make informed cash-related strategic decisions.

The first chapter provided a comprehensive overview of the host organization, detailing its various activities and key figures. It also presented the context of this study, which is part of a mission for a PwC client company aiming to enhance its current cash flow management processes through predictive analysis.

In the second chapter, we explored the significance and importance of cash flow, breaking down its components. Given that the primary cash source for our firm is cash flow from operations, the focus was on accounts receivable and accounts payable management. These processes were thoroughly discussed in the third chapter, where areas for improvement were identified, and predictive modeling was proposed as a major enhancement strategy.

The fourth and fifth chapters delved into the theoretical aspects of machine learning and time series forecasting, setting the stage for their application in our proposed solution. The sixth chapter detailed the implementation of our solution, covering data collection, preprocessing, understanding, model selection, and evaluation. This study significantly impacts cash flow management by:

- **Enhancing Cash Flow Forecasting:**

  By utilizing machine learning and time series forecasting models, the firm can accurately predict future cash inflows and outflows. This predictive capability allows for more precise financial planning and reduces the uncertainty associated with cash flow management.

- **Improving Liquidity Management:**

  The ability to forecast accounts receivable and accounts payable ensures that the firm maintains optimal liquidity levels. This helps in avoiding cash shortages and ensures that funds are available to meet operational needs and investment opportunities.

- **Optimizing Working Capital:**

  Predictive analytics enables the firm to optimize its working capital by efficiently managing the timing of cash flows. This leads to better utilization of resources, minimizing idle cash and reducing the cost of capital.

- **Mitigating Financial Risks:**

  By identifying patterns and trends in payment behaviors, the firm can proactively manage credit risk and avoid potential bad debts. This helps in maintaining a healthy balance sheet and reduces the risk of financial distress.

- **Strategic Decision-Making:**

The insights gained from predictive models provide valuable information for strategic decision-making. Managers can make informed decisions regarding investment, financing, and operational activities, leading to improved overall financial performance.

- **Enhancing Collection Processes:**

  Predictive models can forecast which customers are likely to delay payments, allowing the firm to take proactive measures to improve collections. This helps in reducing the days sales outstanding (DSO) and improving cash flow from operations.

To further enhance this solution, future perspectives include:

- **Integrating Investment and Financing Predictions:**

  Expanding the scope of predictive analytics to include investment and financing activities will provide a comprehensive view of the firm's financial landscape, enabling better long-term planning and resource allocation.

- **Incorporating External Economic Indicators:**

  Including external economic data such as market trends, interest rates, and economic forecasts will enhance the accuracy of cash flow predictions and provide a more holistic understanding of the financial environment.

- **Implementing Smart Dashboards:**

  Deploying the predictive models on interactive and user-friendly dashboards will facilitate real-time monitoring and decision-making. These dashboards can provide managers with up-to-date insights, enabling them to respond quickly to changes in cash flow dynamics.

This study demonstrates the significant benefits of integrating advanced predictive analytics into cash flow management, offering a robust framework for enhancing financial stability and strategic planning.

Bibliography

[1] Sergio Garcia. (2023). *What is Cash Forecasting?*. Trovata Blog. Available at https://trovata.io/blog/what-is-cash-forecasting/

[2] Luo Feng. (2022). *Predicting and Improving Invoice-to-Cash Collection Through Machine Learning.* Ph.D. Thesis, Massachusetts Institute of Technology. Available at https://dspace.mit.edu/bitstream/handle/1721.1/99584/925473704-MIT.pdf?sequence=1

[3] BDO Belgium. (2022). *E-Guide: Cash Flow Management.* Available at https://www.bdo.be/getmedia/67ad3c74-8a22-4c15-b330-246e58ffa59f/2022_04_E-Guide-cash-flow-management_EN_def_1.pdf.aspx

[4] K L University. (2023). *Time Series Analysis Forcasting* . Available at https://www.kluniversity.in/arp/uploads/2093.pdf

[5] Alexander J. Smola. *An Introduction to Machine Learning.* Draft Version. Available at https://alex.smola.org/drafts/thebook.pdf

[6] Hayes, A. (2024). *Cash Flow: What It Is, How It Works, and How to Analyze It.* Updated April 17, 2024. Available at: https://www.investopedia.com/terms/c/cashflow.asp

[7] Rastogi, R. (2023). *Support Vector Regression and Its Mathematical Implementation.* Available at: https://medium.com/@rahulrastogi1104/support-vector-regression-and-its-mathematical-implementation-b6377898cd74

[8] Aurélien Géron. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly Media. Available at https://powerunit-ju.com/wp-content/uploads/2021/04/Aurelien-Geron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-OReilly-Media-2019.pdf

[9] Mingtao, L. (2019). *A general architecture of XGBoost.* Available at: https://www.researchgate.net/figure/A-general-architecture-of-XGBoost_fig3_335483097

[10] Silipo, R. (2020). *From a Single Decision Tree to a Random Forest.* Available at: https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147

[11] Abdullah, S. M. (2023). *Activation Functions.* Available at: https://www.researchgate.net/figure/figure-2-explains-the-activation-function-10-This-determines-whether-or-not-the-data_fig2_376170562

[12] Liu, Q., Wu, Y. (2005). *Supervised Learning.* FX Palo Alto Laboratory. Available at: https://www.researchgate.net/publication/229031588_Supervised_Learning

[13] Besse, P., al. (2020). *Deep Learning.* Wikistat. Available at: https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning.pdf

[14] Seel, N. M. (Ed.). (2012). *Encyclopedia of the Sciences of Learning.* Springer. Available at: https://link.springer.com/referencework/10.1007/978-1-4419-1428-6

[15] Jain, A. (2024). *Ridge and Lasso Regression in Python – Complete Guide.* Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/

[16] Analytics Vidhya. (2020). *Support Vector Regression Tutorial for Machine Learning.* Available at: https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/#:~:text=Support%20Vector%20Regression%20(SVR)%20is,while%20minimizing%20the%20prediction%20error.

[17] Dass, S. (2020). *Stationarity in Time Series Analysis.* Towards Data Science. Available at: https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322

[18] Mordor Intelligence. (2024). *Consulting Service Market Size & Share Analysis - Growth Trends & Forecasts (2024 - 2029).* Available at: https://www.mordorintelligence.com/industry-reports/consulting-service-market

[19] Analytics Vidhya. (2023). *Learning Time Series Analysis: Modern Statistical Models.* Available at: https://www.analyticsvidhya.com/blog/2023/01/learning-time-series-analysis-modern-statistical-models/

[20] Analytics Vidhya. (2021). *A Comprehensive Guide to Time Series Analysis.* Available at: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-to-time-series-analysis/

[21] Buteikis, A. (2019). *Lecture 3: Introduction to Time Series Analysis.* Available at: https://web.vu.lt/mif/a.buteikis/wp-content/uploads/2019/02/Lecture_03.pdf

[22] Santra, R. (2023). *What is Time Series and Components of Time Series.* Medium. Available at: https://medium.com/@ritusantra/what-is-time-series-and-components-of-time-series-c80b69ad5cb9

[23] Choudhary, A. (2022). *Generate Accurate Forecasts with Facebook Prophet (with Python R).* Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/

[24] Billy McDevitt. (2023). *How to Forecast Accounts Payable.* Tratta Blog. Available at https://www.tratta.io/blog/how-to-forecast-accounts-payable

[25] Egyankosh. (2024). *Payables Management: Integrating Working Capital and Capital Investment Processes.* Available at: https://egyankosh.ac.in/bitstream/123456789/6216/1/Unit-13.pdf

[26] Regate by Qonto. (2023). *Connaître et analyser les flux de trésorerie.* Updated April 28, 2023. Available at: https://www.regate.io/blog/connaitre-et-analyser-les-flux-de-tresorerie.

[27] Buteikis, A. (2010). *Time series with trend and seasonality components.* Available at: https://web.vu.lt/mif/a.buteikis/wp-content/uploads/2019/02/Lecture_03.pdf

[28] Universidade NOVA de Lisboa. (2024). *Predicting Account Receivables Outcomes with Machine Learning.* Available at: https://run.unl.pt/bitstream/10362/134205/1/TGI0546.pdf

[29] Cronie, G. (2024). *Order-to-Cash Processes.* Head Sales, Payments and Cash Management, ING. University of Tennessee, Knoxville. Available at: https://web.utk.edu/~jwachowi/INGpart2.pdf

Annexes

Annex A

Presentation of SAP HANA Database

SAP HANA is a high-performance in-memory database designed to handle both high transaction rates and complex query processing on the same platform. This database is built to support real-time analytics and applications. Here is a detailed examination of its architecture, features, and capabilities.

1. **In-Memory Technology**
   SAP HANA operates primarily in-memory, storing data in RAM rather than on traditional disk storage. This approach significantly reduces data access times, enabling real-time data processing and analytics.
2. **Column-Based Data Storage**
   Unlike traditional row-oriented databases, SAP HANA uses a column-based storage architecture which improves performance in read-intensive database operations and allows for better compression and parallel processing.
3. **Data Modeling and Processing Capabilities**
   SAP HANA supports complex calculations and transformations directly in the database through its advanced data modeling capabilities. It also supports SQL, and includes tools for processing graph and series data.
4. **Scalability and Multimodel Capabilities**
   Designed to scale both horizontally and vertically, SAP HANA supports a diverse range of data types and models within a unified environment. It can handle structured, semi-structured, and unstructured data efficiently.
5. **Real-Time Replication and Data Integration**
   Features like SAP Landscape Transformation and SAP Replication Server facilitate real-time data replication from various sources, maintaining up-to-date information within the database.
6. **Security Features**
   SAP HANA provides advanced encryption, dynamic data masking, and role-based access control to ensure data security and regulatory compliance.
7. **Deployment Options**
   SAP HANA can be deployed on-premises, in the cloud, or in a hybrid setting, offering flexibility to meet different organizational needs and data governance standards.

Python Libraries

- **Pandas:** Pandas is a powerful and flexible open-source data analysis and manipulation library in Python, specifically designed to work with labeled and relational data effortlessly. It introduces two primary data structures: Series (one-dimensional) and DataFrame (two-dimensional), which can handle a vast range of data types and are equipped with a comprehensive set of operations for indexing, slicing, reshaping, and aggregating data. With tools for reading and writing data between in-memory data structures and different formats: CSV, text files, Microsoft Excel, SQL databases, and the fast HDF5 format, Pandas is ideally suited for data wrangling and preparation. It also features time-series functionality, window and aggregation operations, and missing data handling. Developed by Wes McKinney in 2008, Pandas has become an indispensable tool in quantitative economics, finance, statistics, analytics, and more, thanks to its easy-to-use interface and wide range of functionalities.

- **NumPy:** NumPy, short for Numerical Python, is a fundamental package for scientific computing with Python. It includes support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy arrays provide much more efficient storage and data operations as arrays grow in size than Python's built-in list data structure. The core feature of NumPy is the "ndarray", an N-dimensional array type which is a contiguous block of memory, consisting of two parts: the data buffer which is an actual block of memory storing the values, and the metadata which describes the data buffer including how to locate an element and how to interpret an element. It is the foundation on which virtually all the higher-level tools in this ecosystem are built, particularly Pandas, and is known for its speed and flexibility in data handling.
- **Matplotlib:** Matplotlib is a versatile 2D plotting library for the Python programming language that enables the creation of high-quality graphs, charts, figures, and plots from data. Developed by John D. Hunter in 2002, Matplotlib emulates the plotting capabilities of MATLAB and provides an object-oriented API which can be used to embed plots into applications using Python GUI toolkits such as Tkinter, wxPython, Qt, or GTK. It can produce plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code. For scientists, data analysts, and engineers, Matplotlib is a vital tool for data visualization, providing an essential means of analyzing and sharing data results.
- **Seaborn:** Seaborn is a statistical data visualization library designed to integrate closely with Pandas data structures and is built on top of Matplotlib. It provides a high-level interface for drawing attractive statistical graphics and aims to make visualization a central part of exploring and understanding data. Its plotting functions operate on DataFrames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Seaborn's ability to simplify complex visualizations of large amounts of data and its integration with Pandas structures makes it a powerful tool for exploratory data analysis.
- **Statsmodels:** Statsmodels is a library for fitting many kinds of statistical models, performing statistical tests, and data exploration. The extensive list of descriptive statistics, statistical tests, plotting functions, and result statistics provide a solid foundation for data analysts and statisticians to explore data and model relationships. Supported models include linear regression, generalized linear models, discrete choice models, robust linear models, and many others. Statsmodels also allows users to explore model results through a comprehensive summary output, providing key insights into data.
- **Scikit-learn:** Scikit-learn is a library for machine learning that provides simple and efficient tools for predictive data analysis built on NumPy, SciPy, and Matplotlib. It supports various supervised and unsupervised learning algorithms. The

library includes classification, regression, clustering, and dimensionality reduction, and implements a clean, uniform, and streamlined API. Scikit-learn makes it possible to perform complex data analysis with few lines of code, which democratizes machine learning and makes it accessible to a broader audience of developers and data scientists.

Annex B

Concepts Related to Machine Learning and Time Series Analysis

Hyperparameter Tuning

Hyperparameter tuning, or hyperparameter optimization, is an essential process in machine learning that involves selecting the optimal set of hyperparameters for a learning algorithm to achieve the best possible performance. Unlike model parameters, which are learned during training, hyperparameters are set before the training process begins and control various aspects of the training algorithm and model structure, such as learning rate, batch size, number of layers in a neural network, and regularization strength. Effective hyperparameter tuning can significantly impact a model's ability to generalize from training data to unseen data, thereby improving predictive accuracy and robustness. The process can be performed using various strategies, including grid search, random search, Bayesian optimization, gradient-based optimization, and evolutionary algorithms.

Bayesian Optimization

Bayesian optimization is a powerful strategy for optimizing objective functions that are expensive to evaluate. It is particularly useful for hyperparameter tuning in machine learning models. The core idea of Bayesian optimization is to build a probabilistic model of the objective function and use this model to make decisions about where to evaluate the function next.

Mathematically, Bayesian optimization seeks to find the maximum of an objective function $f(\mathbf{x})$ over a bounded domain $\mathcal{X} \subset \mathbb{R}^d$. The process involves the following steps: **1Surrogate Model**

A surrogate model, typically a Gaussian Process (GP), is used to model the unknown objective function. The GP is specified by a mean function $m(\mathbf{x})$ and a covariance function (kernel) $k(\mathbf{x}, \mathbf{x}')$, and is defined as:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Acquisition Function

An acquisition function $\alpha(\mathbf{x}; \mathcal{D})$ is used to decide the next point to evaluate. It balances exploration (searching new areas) and exploitation (refining known good areas). Common acquisition functions include:

- **Expected Improvement (EI)**:

  $$\alpha_{EI}(\mathbf{x}; \mathcal{D}) = \mathbb{E}[\max(0, f(\mathbf{x}) - f(\mathbf{x}^+))]$$

  where $\mathbf{x}^+$ is the current best point.
- **Probability of Improvement (PI)**:

  $$\alpha_{PI}(\mathbf{x}; \mathcal{D}) = \mathbb{P}(f(\mathbf{x}) > f(\mathbf{x}^+))$$

- **Upper Confidence Bound (UCB)**:

  $$\alpha_{UCB}(\mathbf{x}; \mathcal{D}) = \mu(\mathbf{x}) + \kappa\sigma(\mathbf{x})$$

  where $\mu(\mathbf{x})$ is the predicted mean, $\sigma(\mathbf{x})$ is the predicted standard deviation, and κ is a hyperparameter.

Iterative Optimization

Bayesian optimization iteratively updates the surrogate model and acquisition function:

1. Evaluate the objective function $f(\mathbf{x})$ at the point $\mathbf{x}_{\text{next}} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x};\mathcal{D})$.
2. Augment the data $\mathcal{D}$ with the new observation $(\mathbf{x}_{\text{next}}, f(\mathbf{x}_{\text{next}}))$.
3. Update the surrogate model with the new data.

Mathematical Foundations

The Gaussian Process (GP) provides a distribution over functions and is defined by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

Given a set of observations $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, where $y_i = f(\mathbf{x}_i) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise, the posterior distribution of the GP can be derived using Bayes' theorem.

ACF and PACF

Autocorrelation Function (ACF)

Definition: The Autocorrelation Function (ACF) measures the correlation between a time series and its own past values. It is essentially a measure of how well a time series is related to itself over different time lags. The ACF is calculated for different lag values k, where k represents the number of time steps separating the two points being compared.

Mathematically: For a time series x_t, the ACF at lag k, denoted as $\rho(k)$, is given by:

$$\rho(k) = \frac{\sum_{t=k+1}^{N}(x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^{N}(x_t - \bar{x})^2}$$

where $\bar{x}$ is the mean of the time series, and N is the number of observations.

Utility:

- **Identifying Seasonality:** ACF can help in identifying seasonality in the data. Peaks in the ACF at regular intervals suggest a seasonal pattern.
- **Choosing ARIMA Models:** ACF is used in determining the order of Moving Average (MA) components in ARIMA models. Significant spikes in the ACF plot suggest the need for MA terms.
- **Diagnosing Model Fit:** After fitting a time series model, the ACF of the residuals can be examined to check for any remaining autocorrelation, indicating model inadequacy if present.

Partial Autocorrelation Function (PACF)

Definition: The Partial Autocorrelation Function (PACF) measures the correlation between a time series and its own past values, but with the linear dependence of the intermediate lags removed. Essentially, it quantifies the direct effect of a particular lag on the time series while accounting for the influence of other lags.

Mathematically: For a time series x_t, the PACF at lag k, denoted as $\phi(k)$, is obtained by fitting autoregressive models of different orders and considering the direct effect of the k-th lag.

Utility:

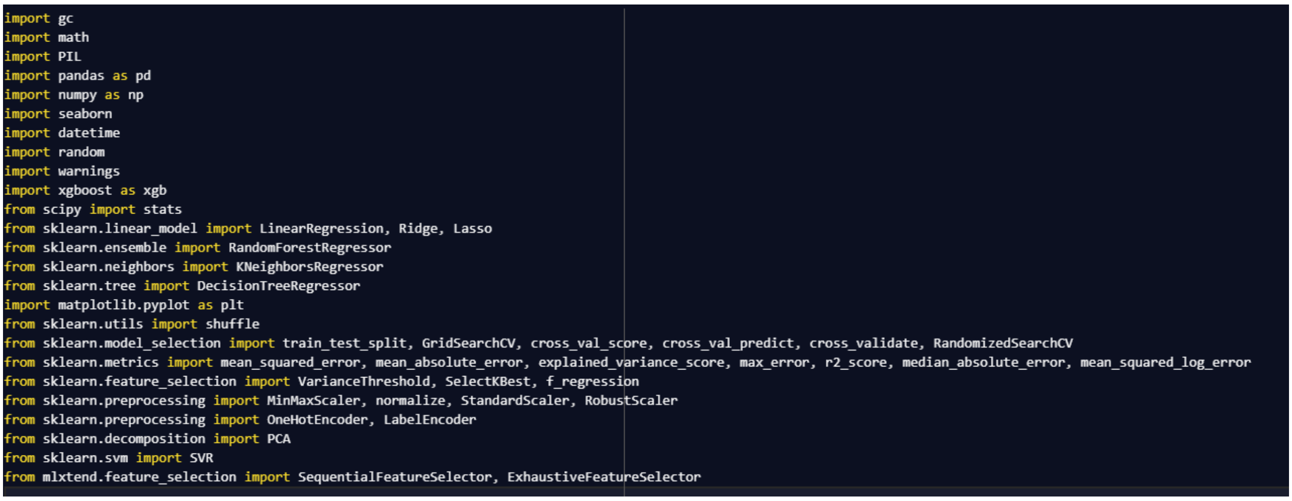
- **Choosing ARIMA Models:** PACF is used in determining the order of Autoregressive (AR) components in ARIMA models. Significant spikes in the PACF plot suggest the need for AR terms.
- **Model Identification:** PACF helps in identifying the appropriate lags for AR models by showing the lags that have a direct influence on the series, rather than indirect effects.
- **Analyzing Relationships:** PACF can help understand the direct relationships between different points in the time series, making it useful for interpreting the underlying processes.

Annex C

Python Code

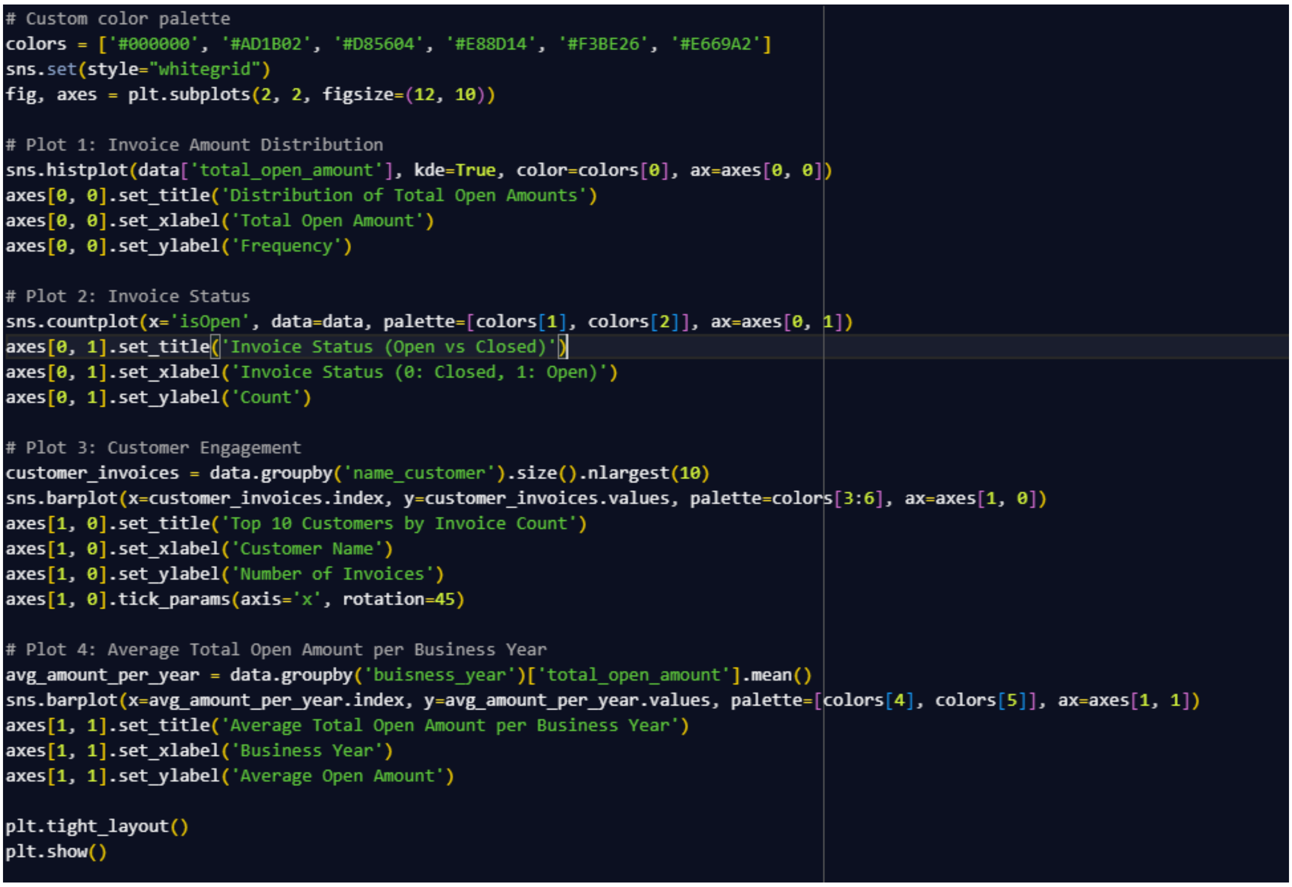
AR Payment Prediction

Libraries Importation:

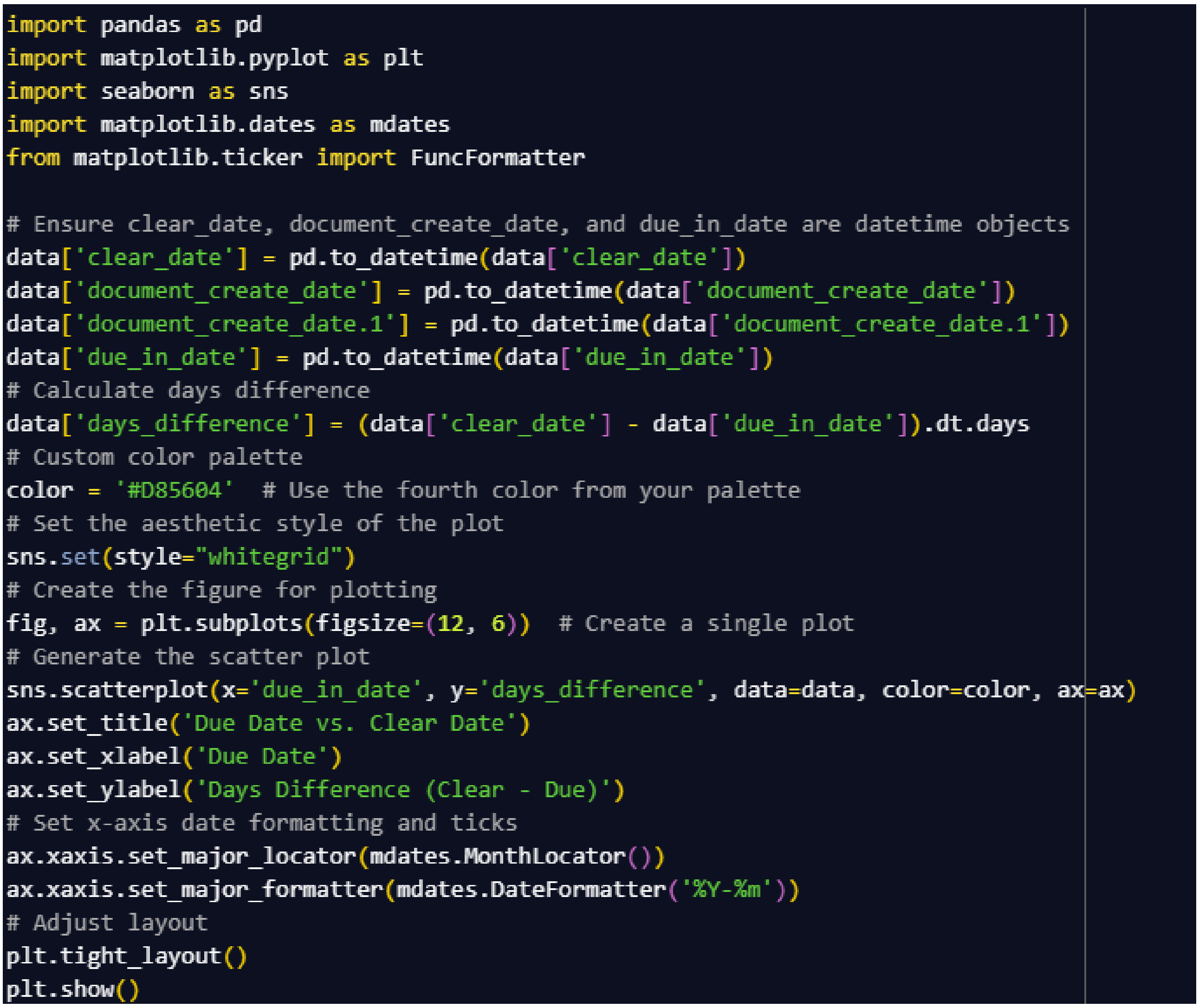


```
import gc
import math
import PIL
import pandas as pd
import numpy as np
import seaborn
import datetime
import random
import warnings
import xgboost as xgb
from scipy import stats
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score, cross_val_predict, cross_validate, RandomizedSearchCV
from sklearn.metrics import mean_squared_error, mean_absolute_error, explained_variance_score, max_error, r2_score, median_absolute_error, mean_squared_log_error
from sklearn.feature_selection import VarianceThreshold, SelectKBest, f_regression
from sklearn.preprocessing import MinMaxScaler, normalize, StandardScaler, RobustScaler
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
from sklearn.decomposition import PCA
from sklearn.svm import SVR
from mlxtend.feature_selection import SequentialFeatureSelector, ExhaustiveFeatureSelector
```

Exploratory Data Analysis (EDA)

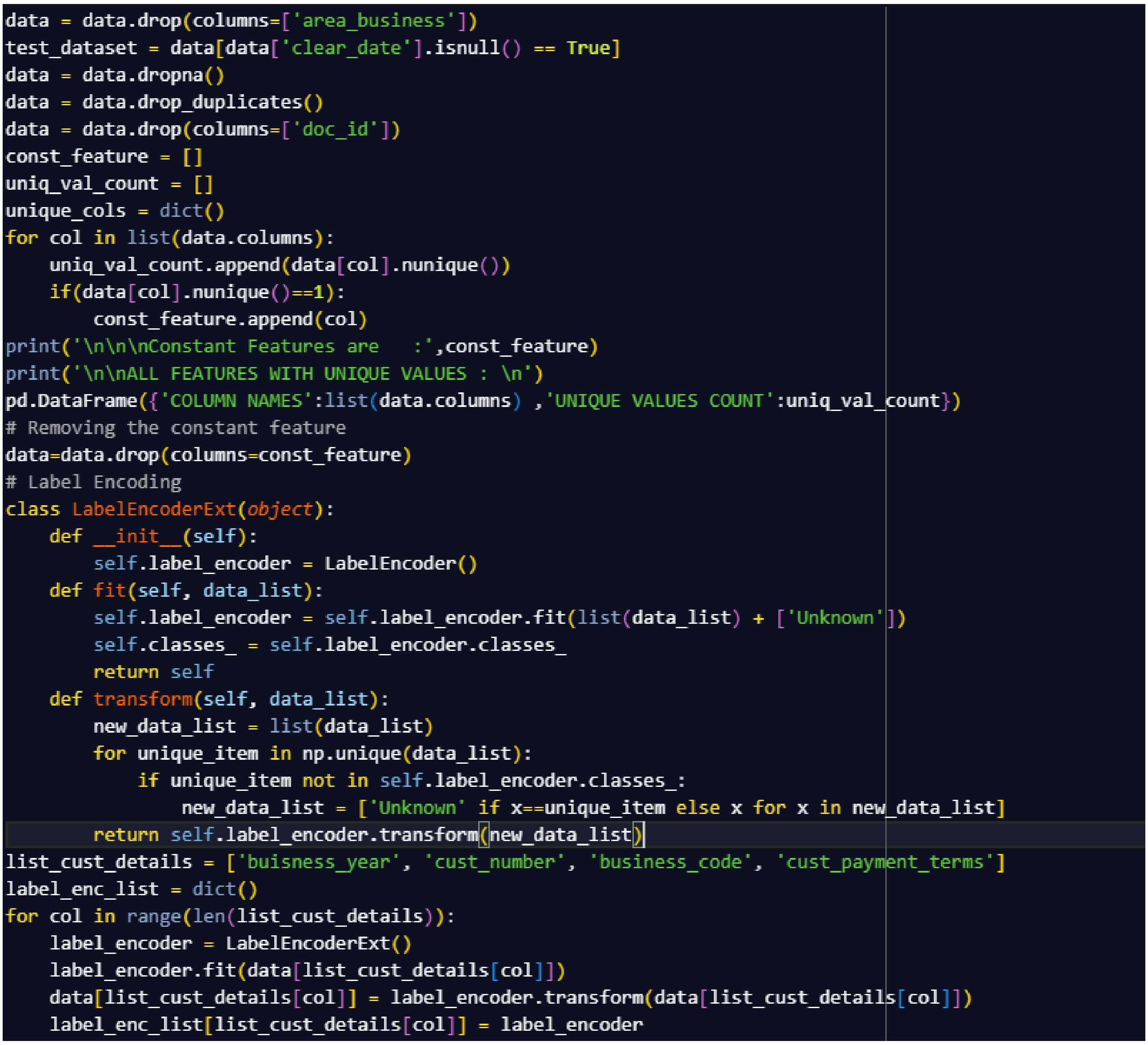


```
# Custom color palette
colors = ['#000000', '#AD1B02', '#D85604', '#E88D14', '#F3BE26', '#E669A2']
sns.set(style="whitegrid")
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

# Plot 1: Invoice Amount Distribution
sns.histplot(data['total_open_amount'], kde=True, color=colors[0], ax=axes[0, 0])
axes[0, 0].set_title('Distribution of Total Open Amounts')
axes[0, 0].set_xlabel('Total Open Amount')
axes[0, 0].set_ylabel('Frequency')

# Plot 2: Invoice Status
sns.countplot(x='isOpen', data=data, palette=[colors[1], colors[2]], ax=axes[0, 1])
axes[0, 1].set_title('Invoice Status (Open vs Closed)')
axes[0, 1].set_xlabel('Invoice Status (0: Closed, 1: Open)')
axes[0, 1].set_ylabel('Count')

# Plot 3: Customer Engagement
customer_invoices = data.groupby('name_customer').size().nlargest(10)
sns.barplot(x=customer_invoices.index, y=customer_invoices.values, palette=colors[3:6], ax=axes[1, 0])
axes[1, 0].set_title('Top 10 Customers by Invoice Count')
axes[1, 0].set_xlabel('Customer Name')
axes[1, 0].set_ylabel('Number of Invoices')
axes[1, 0].tick_params(axis='x', rotation=45)

# Plot 4: Average Total Open Amount per Business Year
avg_amount_per_year = data.groupby('buisness_year')['total_open_amount'].mean()
sns.barplot(x=avg_amount_per_year.index, y=avg_amount_per_year.values, palette=[colors[4], colors[5]], ax=axes[1, 1])
axes[1, 1].set_title('Average Total Open Amount per Business Year')
axes[1, 1].set_xlabel('Business Year')
axes[1, 1].set_ylabel('Average Open Amount')

plt.tight_layout()
plt.show()
```



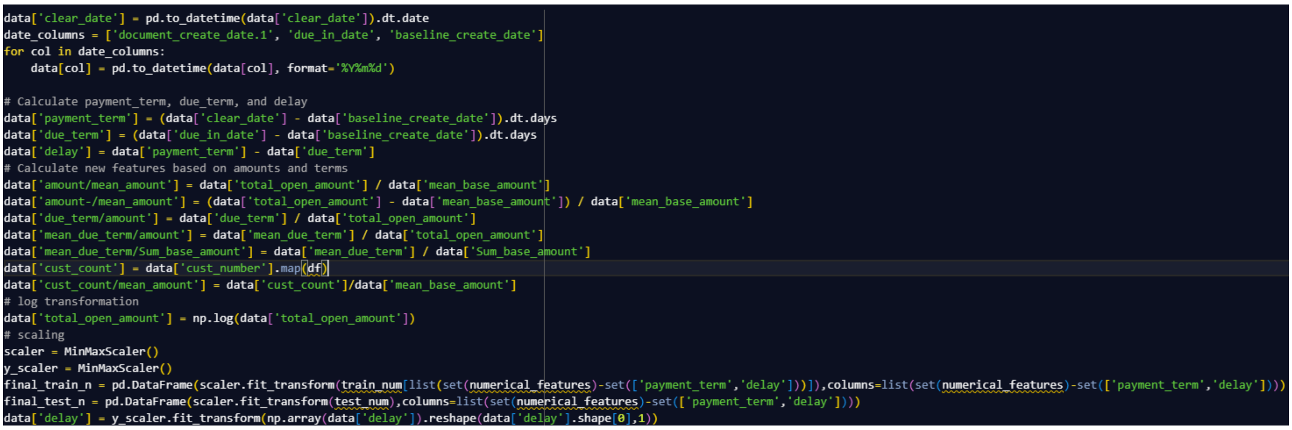
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.dates as mdates
from matplotlib.ticker import FuncFormatter

# Ensure clear_date, document_create_date, and due_in_date are datetime objects
data['clear_date'] = pd.to_datetime(data['clear_date'])
data['document_create_date'] = pd.to_datetime(data['document_create_date'])
data['document_create_date.1'] = pd.to_datetime(data['document_create_date.1'])
data['due_in_date'] = pd.to_datetime(data['due_in_date'])
# Calculate days difference
data['days_difference'] = (data['clear_date'] - data['due_in_date']).dt.days
# Custom color palette
color = '#D85604'  # Use the fourth color from your palette
# Set the aesthetic style of the plot
sns.set(style="whitegrid")
# Create the figure for plotting
fig, ax = plt.subplots(figsize=(12, 6))  # Create a single plot
# Generate the scatter plot
sns.scatterplot(x='due_in_date', y='days_difference', data=data, color=color, ax=ax)
ax.set_title('Due Date vs. Clear Date')
ax.set_xlabel('Due Date')
ax.set_ylabel('Days Difference (Clear - Due)')
# Set x-axis date formatting and ticks
ax.xaxis.set_major_locator(mdates.MonthLocator())
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m'))
# Adjust layout
plt.tight_layout()
plt.show()
```

Data Preprocessing



```python
data = data.drop(columns=['area_business'])
test_dataset = data[data['clear_date'].isnull() == True]
data = data.dropna()
data = data.drop_duplicates()
data = data.drop(columns=['doc_id'])
const_feature = []
uniq_val_count = []
unique_cols = dict()
for col in list(data.columns):
    uniq_val_count.append(data[col].nunique())
    if(data[col].nunique()==1):
        const_feature.append(col)
print('\n\n\nConstant Features are    :',const_feature)
print('\n\nALL FEATURES WITH UNIQUE VALUES : \n')
pd.DataFrame({'COLUMN NAMES':list(data.columns) ,'UNIQUE VALUES COUNT':uniq_val_count})
# Removing the constant feature
data=data.drop(columns=const_feature)
# Label Encoding
class LabelEncoderExt(object):
    def __init__(self):
        self.label_encoder = LabelEncoder()
    def fit(self, data_list):
        self.label_encoder = self.label_encoder.fit(list(data_list) + ['Unknown'])
        self.classes_ = self.label_encoder.classes_
        return self
    def transform(self, data_list):
        new_data_list = list(data_list)
        for unique_item in np.unique(data_list):
            if unique_item not in self.label_encoder.classes_:
                new_data_list = ['Unknown' if x==unique_item else x for x in new_data_list]
        return self.label_encoder.transform(new_data_list)
list_cust_details = ['buisness_year', 'cust_number', 'business_code', 'cust_payment_terms']
label_enc_list = dict()
for col in range(len(list_cust_details)):
    label_encoder = LabelEncoderExt()
    label_encoder.fit(data[list_cust_details[col]])
    data[list_cust_details[col]] = label_encoder.transform(data[list_cust_details[col]])
    label_enc_list[list_cust_details[col]] = label_encoder
```

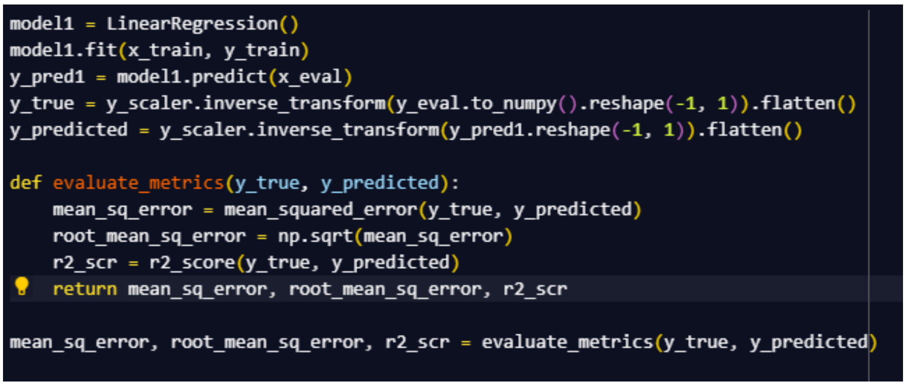
Feature Engineering



```python
data['clear_date'] = pd.to_datetime(data['clear_date']).dt.date
date_columns = ['document_create_date.1', 'due_in_date', 'baseline_create_date']
for col in date_columns:
    data[col] = pd.to_datetime(data[col], format='%Y%m%d')

# Calculate payment_term, due_term, and delay
data['payment_term'] = (data['clear_date'] - data['baseline_create_date']).dt.days
data['due_term'] = (data['due_in_date'] - data['baseline_create_date']).dt.days
data['delay'] = data['payment_term'] - data['due_term']
# Calculate new features based on amounts and terms
data['amount/mean_amount'] = data['total_open_amount'] / data['mean_base_amount']
data['amount-/mean_amount'] = (data['total_open_amount'] - data['mean_base_amount']) / data['mean_base_amount']
data['due_term/amount'] = data['due_term'] / data['total_open_amount']
data['mean_due_term/amount'] = data['mean_due_term'] / data['total_open_amount']
data['mean_due_term/Sum_base_amount'] = data['mean_due_term'] / data['Sum_base_amount']
data['cust_count'] = data['cust_number'].map(df)
data['cust_count/mean_amount'] = data['cust_count']/data['mean_base_amount']
# log transformation
data['total_open_amount'] = np.log(data['total_open_amount'])
# scaling
scaler = MinMaxScaler()
y_scaler = MinMaxScaler()
final_train_n = pd.DataFrame(scaler.fit_transform(train_num[list(set(numerical_features)-set(['payment_term','delay']))]),columns=list(set(numerical_features)-set(['payment_term','delay'])))
final_test_n = pd.DataFrame(scaler.fit_transform(test_num),columns=list(set(numerical_features)-set(['payment_term','delay'])))
data['delay'] = y_scaler.fit_transform(np.array(data['delay']).reshape(data['delay'].shape[0],1))
```

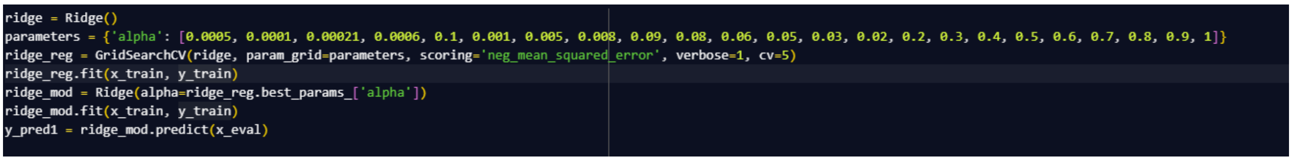
Models Training and Evaluation

Linear Regression

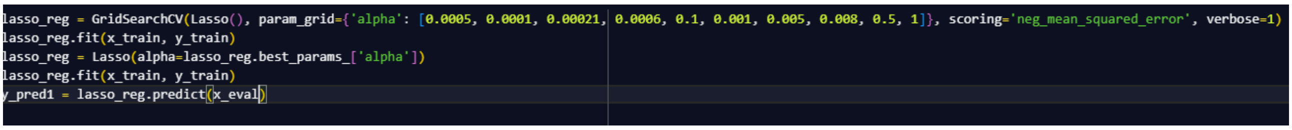


```python
model1 = LinearRegression()
model1.fit(x_train, y_train)
y_pred1 = model1.predict(x_eval)
y_true = y_scaler.inverse_transform(y_eval.to_numpy().reshape(-1, 1)).flatten()
y_predicted = y_scaler.inverse_transform(y_pred1.reshape(-1, 1)).flatten()

def evaluate_metrics(y_true, y_predicted):
    mean_sq_error = mean_squared_error(y_true, y_predicted)
    root_mean_sq_error = np.sqrt(mean_sq_error)
    r2_scr = r2_score(y_true, y_predicted)
    return mean_sq_error, root_mean_sq_error, r2_scr

mean_sq_error, root_mean_sq_error, r2_scr = evaluate_metrics(y_true, y_predicted)
```

Ridge and Lasso Regression

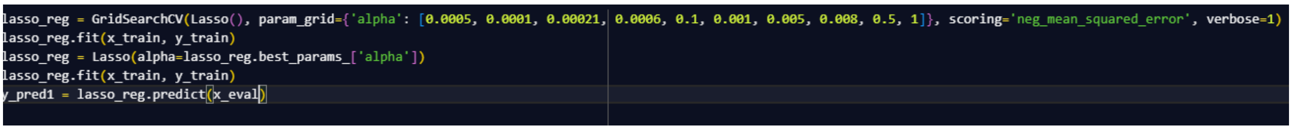


```python
ridge = Ridge()
parameters = {'alpha': [0.0005, 0.0001, 0.00021, 0.0006, 0.1, 0.001, 0.005, 0.008, 0.09, 0.08, 0.06, 0.05, 0.03, 0.02, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]}
ridge_reg = GridSearchCV(ridge, param_grid=parameters, scoring='neg_mean_squared_error', verbose=1, cv=5)
ridge_reg.fit(x_train, y_train)
ridge_mod = Ridge(alpha=ridge_reg.best_params_['alpha'])
ridge_mod.fit(x_train, y_train)
y_pred1 = ridge_mod.predict(x_eval)
```



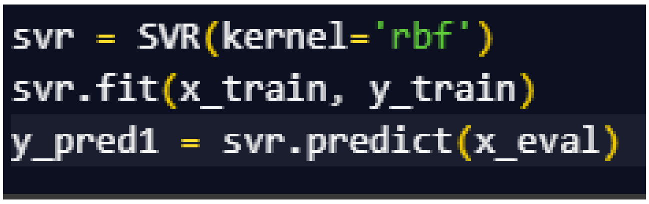
```python
lasso_reg = GridSearchCV(Lasso(), param_grid={'alpha': [0.0005, 0.0001, 0.00021, 0.0006, 0.1, 0.001, 0.005, 0.008, 0.5, 1]}, scoring='neg_mean_squared_error', verbose=1)
lasso_reg.fit(x_train, y_train)
lasso_reg = Lasso(alpha=lasso_reg.best_params_['alpha'])
lasso_reg.fit(x_train, y_train)
y_pred1 = lasso_reg.predict(x_eval)
```

Random Forest Regressor



```python
lasso_reg = GridSearchCV(Lasso(), param_grid={'alpha': [0.0005, 0.0001, 0.00021, 0.0006, 0.1, 0.001, 0.005, 0.008, 0.5, 1]}, scoring='neg_mean_squared_error', verbose=1)
lasso_reg.fit(x_train, y_train)
lasso_reg = Lasso(alpha=lasso_reg.best_params_['alpha'])
lasso_reg.fit(x_train, y_train)
y_pred1 = lasso_reg.predict(x_eval)
```

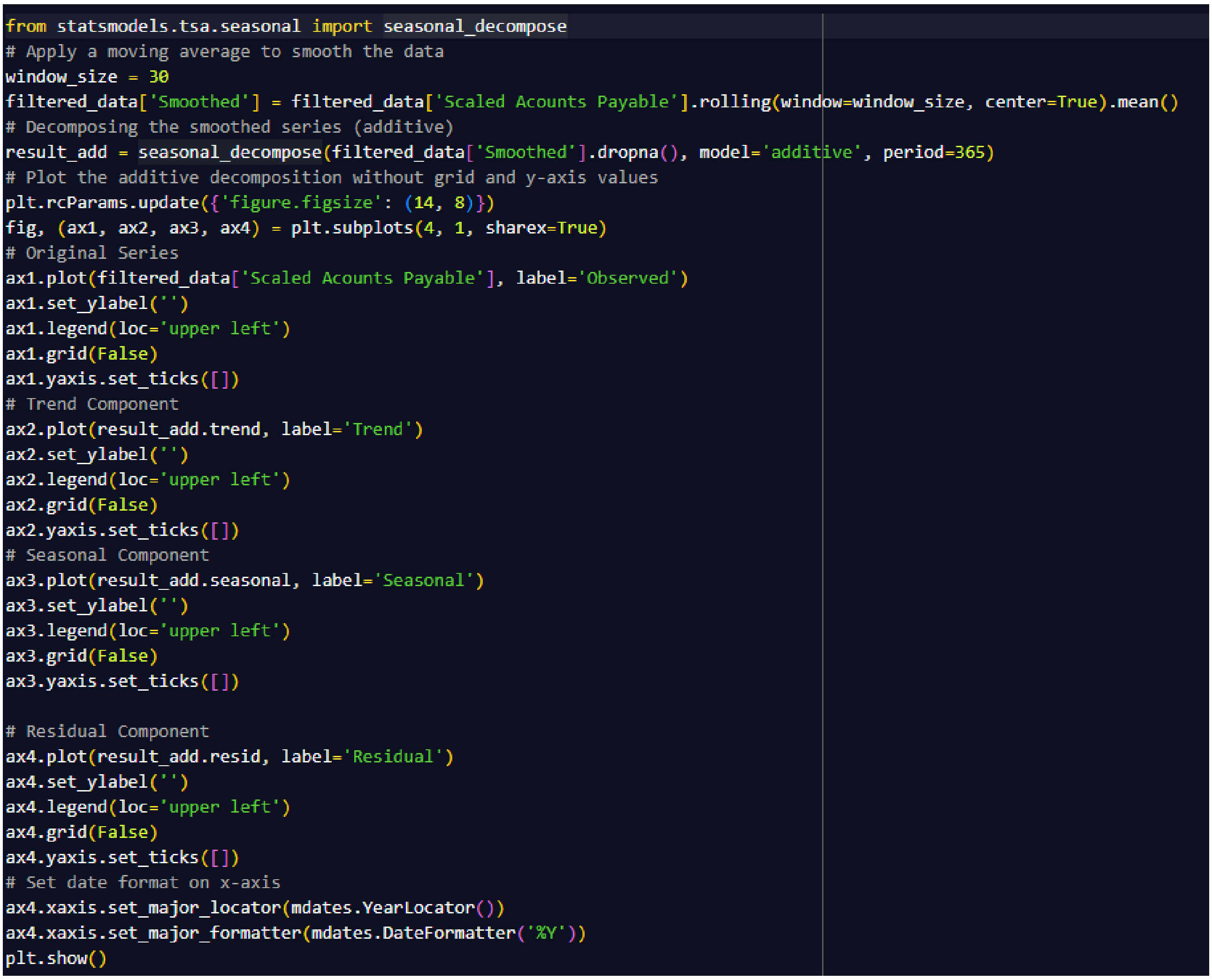
SVR



```python
svr = SVR(kernel='rbf')
svr.fit(x_train, y_train)
y_pred1 = svr.predict(x_eval)
```

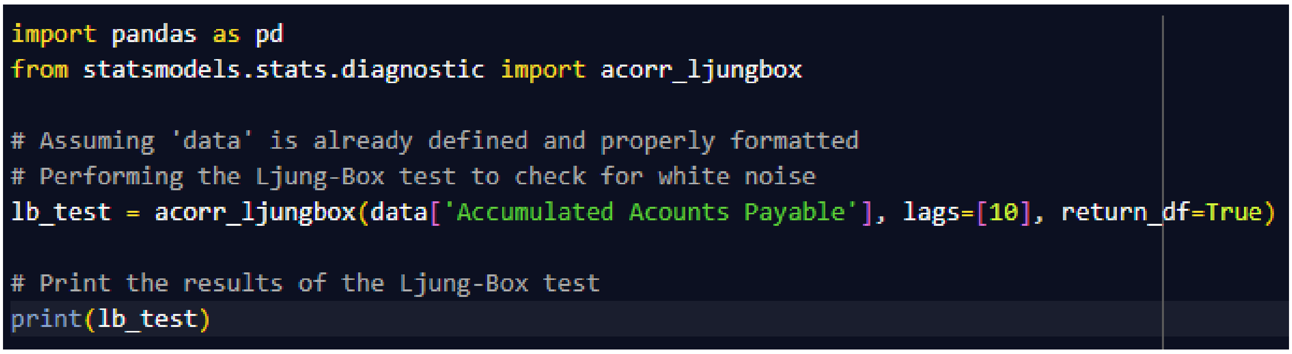
AP Forecasting

Time Series Decomposition



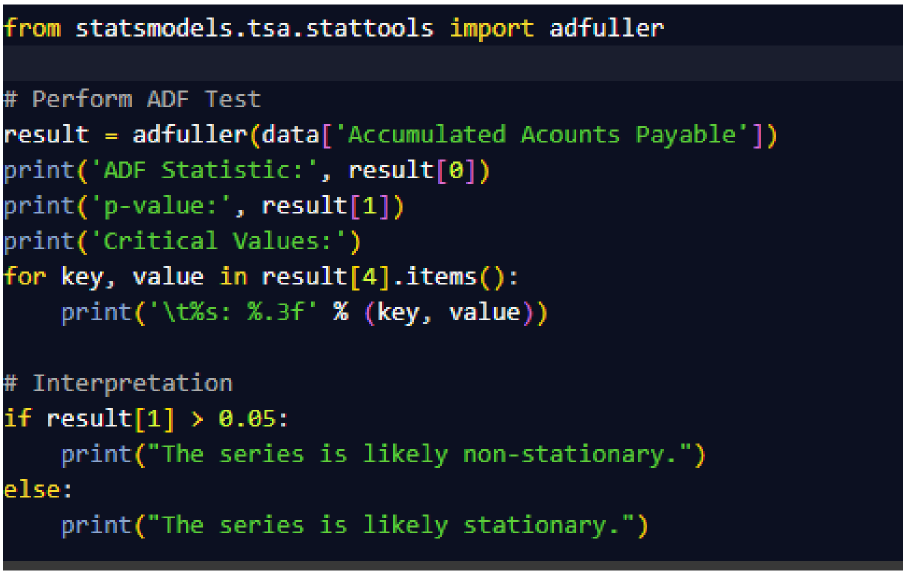
```
from statsmodels.tsa.seasonal import seasonal_decompose
# Apply a moving average to smooth the data
window_size = 30
filtered_data['Smoothed'] = filtered_data['Scaled Acounts Payable'].rolling(window=window_size, center=True).mean()
# Decomposing the smoothed series (additive)
result_add = seasonal_decompose(filtered_data['Smoothed'].dropna(), model='additive', period=365)
# Plot the additive decomposition without grid and y-axis values
plt.rcParams.update({'figure.figsize': (14, 8)})
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, sharex=True)
# Original Series
ax1.plot(filtered_data['Scaled Acounts Payable'], label='Observed')
ax1.set_ylabel('')
ax1.legend(loc='upper left')
ax1.grid(False)
ax1.yaxis.set_ticks([])
# Trend Component
ax2.plot(result_add.trend, label='Trend')
ax2.set_ylabel('')
ax2.legend(loc='upper left')
ax2.grid(False)
ax2.yaxis.set_ticks([])
# Seasonal Component
ax3.plot(result_add.seasonal, label='Seasonal')
ax3.set_ylabel('')
ax3.legend(loc='upper left')
ax3.grid(False)
ax3.yaxis.set_ticks([])

# Residual Component
ax4.plot(result_add.resid, label='Residual')
ax4.set_ylabel('')
ax4.legend(loc='upper left')
ax4.grid(False)
ax4.yaxis.set_ticks([])
# Set date format on x-axis
ax4.xaxis.set_major_locator(mdates.YearLocator())
ax4.xaxis.set_major_formatter(mdates.DateFormatter('%Y'))
plt.show()
```

White Noise Test



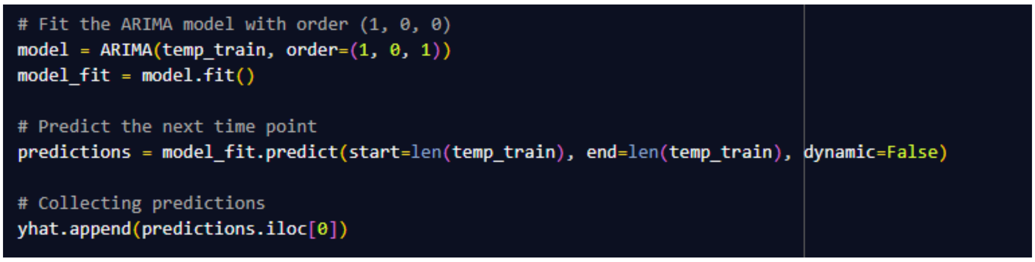
```
import pandas as pd
from statsmodels.stats.diagnostic import acorr_ljungbox

# Assuming 'data' is already defined and properly formatted
# Performing the Ljung-Box test to check for white noise
lb_test = acorr_ljungbox(data['Accumulated Acounts Payable'], lags=[10], return_df=True)

# Print the results of the Ljung-Box test
print(lb_test)
```

Stationarity Check



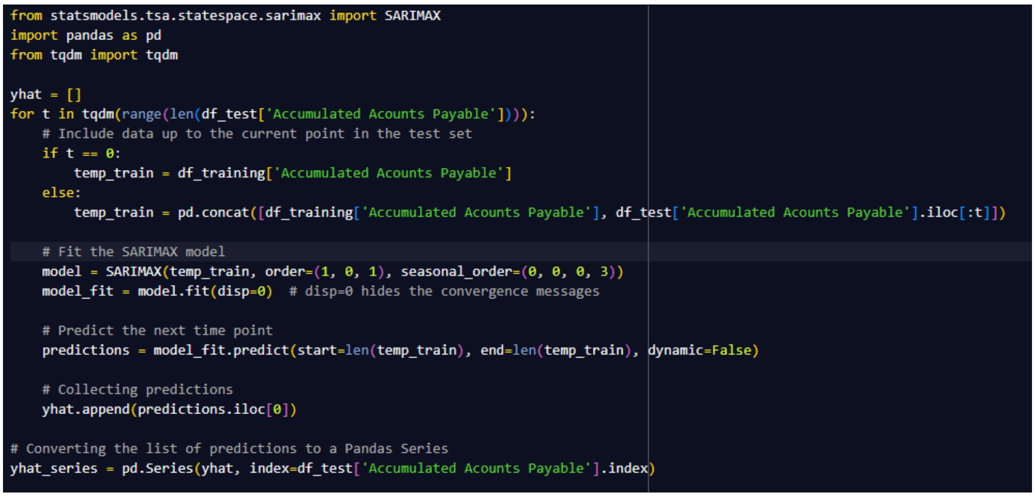
```python
from statsmodels.tsa.stattools import adfuller

# Perform ADF Test
result = adfuller(data['Accumulated Acounts Payable'])
print('ADF Statistic:', result[0])
print('p-value:', result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

# Interpretation
if result[1] > 0.05:
    print("The series is likely non-stationary.")
else:
    print("The series is likely stationary.")
```

ARIMA



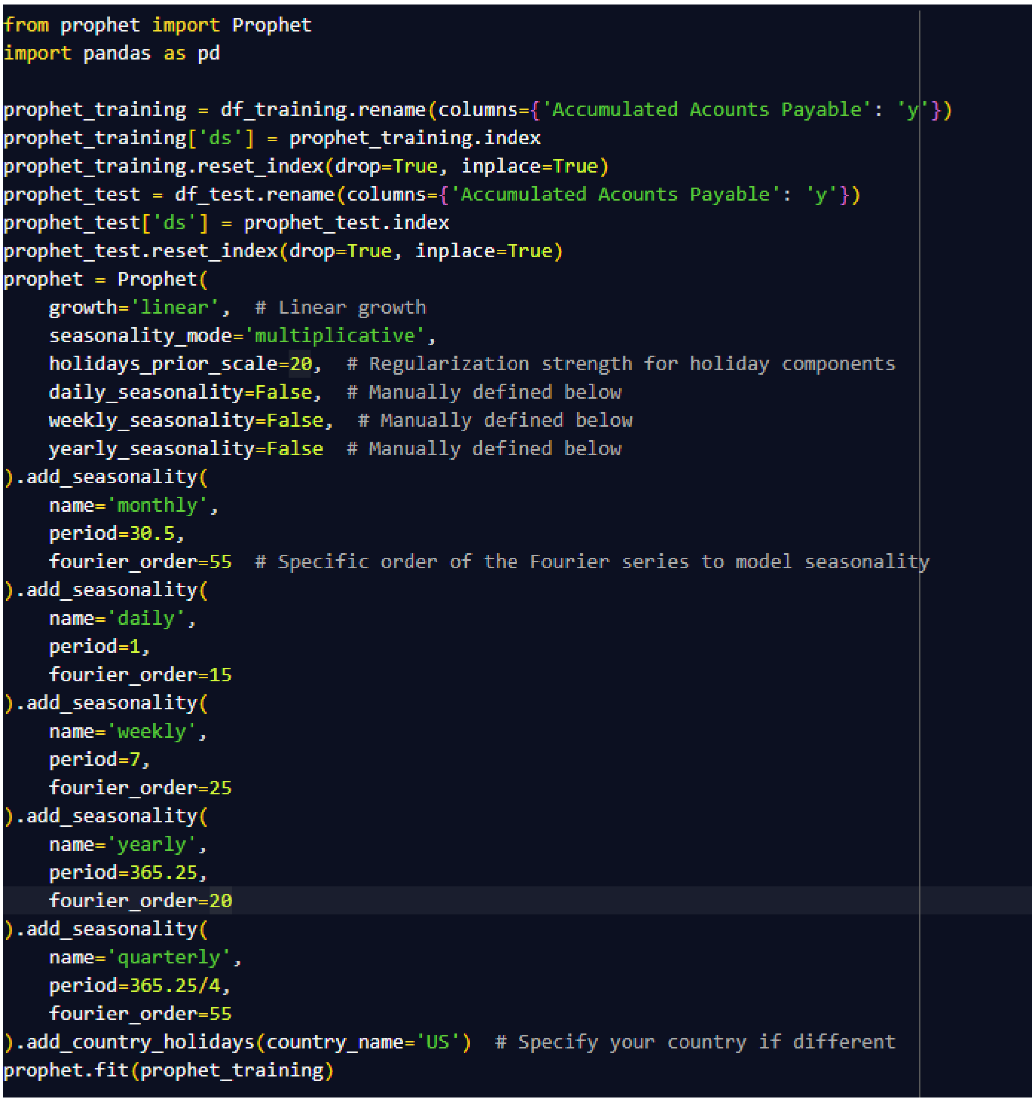
```python
# Fit the ARIMA model with order (1, 0, 0)
model = ARIMA(temp_train, order=(1, 0, 1))
model_fit = model.fit()

# Predict the next time point
predictions = model_fit.predict(start=len(temp_train), end=len(temp_train), dynamic=False)

# Collecting predictions
yhat.append(predictions.iloc[0])
```

SARIMA



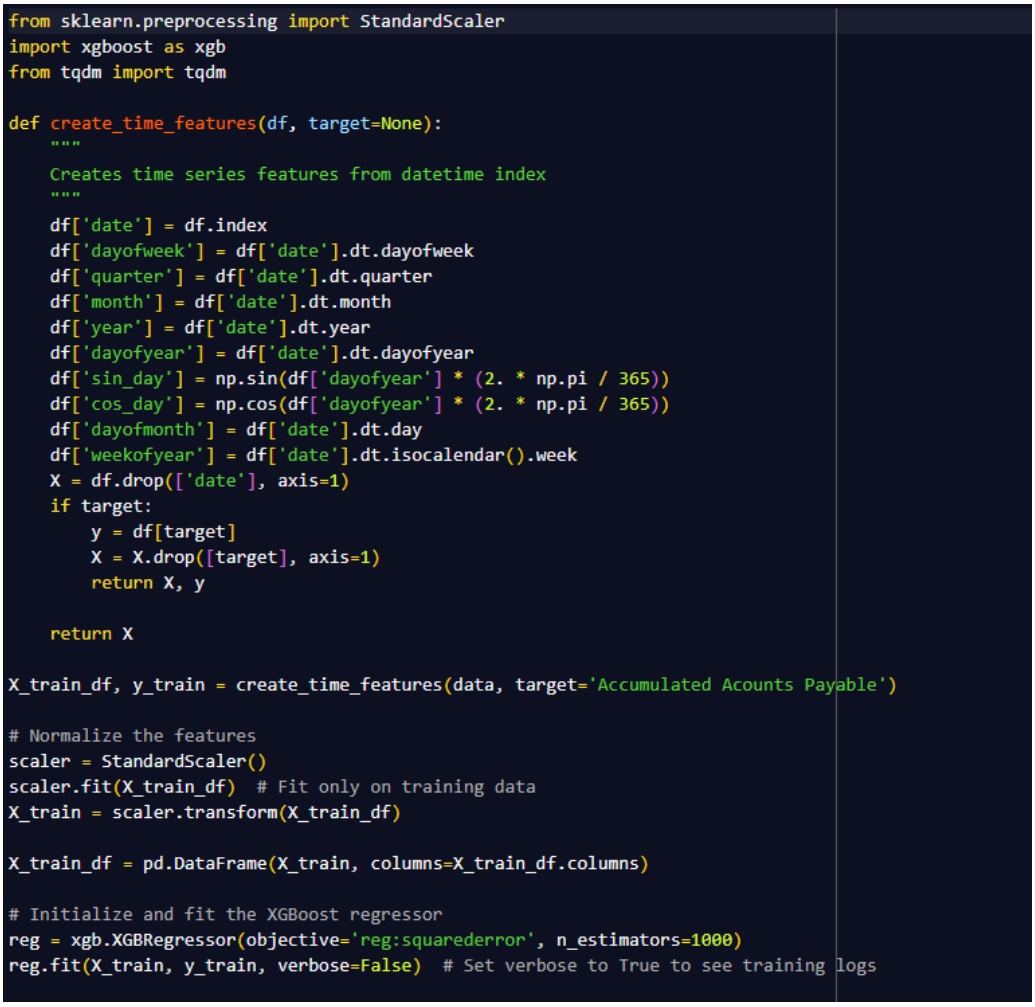
```python
from statsmodels.tsa.statespace.sarimax import SARIMAX
import pandas as pd
from tqdm import tqdm

yhat = []
for t in tqdm(range(len(df_test['Accumulated Acounts Payable']))):
    # Include data up to the current point in the test set
    if t == 0:
        temp_train = df_training['Accumulated Acounts Payable']
    else:
        temp_train = pd.concat([df_training['Accumulated Acounts Payable'], df_test['Accumulated Acounts Payable'].iloc[:t]])

    # Fit the SARIMAX model
    model = SARIMAX(temp_train, order=(1, 0, 1), seasonal_order=(0, 0, 0, 3))
    model_fit = model.fit(disp=0)  # disp=0 hides the convergence messages

    # Predict the next time point
    predictions = model_fit.predict(start=len(temp_train), end=len(temp_train), dynamic=False)

    # Collecting predictions
    yhat.append(predictions.iloc[0])

# Converting the list of predictions to a Pandas Series
yhat_series = pd.Series(yhat, index=df_test['Accumulated Acounts Payable'].index)
```

Prophet



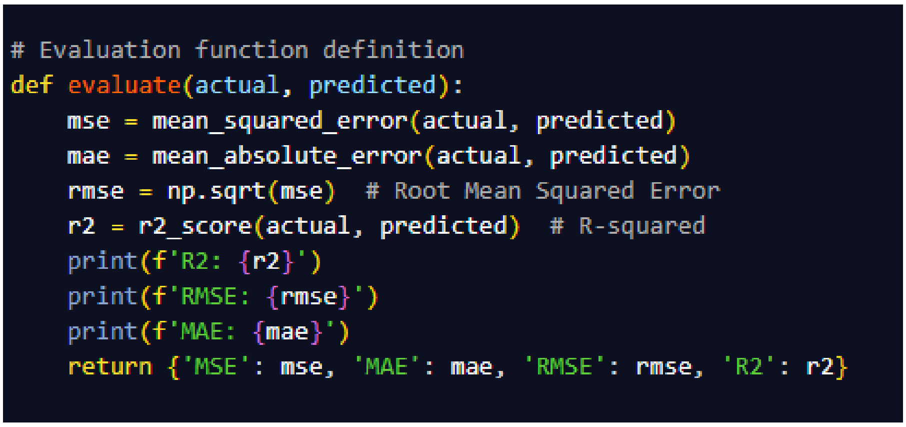
```python
from prophet import Prophet
import pandas as pd

prophet_training = df_training.rename(columns={'Accumulated Acounts Payable': 'y'})
prophet_training['ds'] = prophet_training.index
prophet_training.reset_index(drop=True, inplace=True)
prophet_test = df_test.rename(columns={'Accumulated Acounts Payable': 'y'})
prophet_test['ds'] = prophet_test.index
prophet_test.reset_index(drop=True, inplace=True)
prophet = Prophet(
    growth='linear',  # Linear growth
    seasonality_mode='multiplicative',
    holidays_prior_scale=20,  # Regularization strength for holiday components
    daily_seasonality=False,  # Manually defined below
    weekly_seasonality=False,  # Manually defined below
    yearly_seasonality=False  # Manually defined below
).add_seasonality(
    name='monthly',
    period=30.5,
    fourier_order=55  # Specific order of the Fourier series to model seasonality
).add_seasonality(
    name='daily',
    period=1,
    fourier_order=15
).add_seasonality(
    name='weekly',
    period=7,
    fourier_order=25
).add_seasonality(
    name='yearly',
    period=365.25,
    fourier_order=20
).add_seasonality(
    name='quarterly',
    period=365.25/4,
    fourier_order=55
).add_country_holidays(country_name='US')  # Specify your country if different
prophet.fit(prophet_training)
```

XGBoost



```python
from sklearn.preprocessing import StandardScaler
import xgboost as xgb
from tqdm import tqdm

def create_time_features(df, target=None):
    """
    Creates time series features from datetime index
    """
    df['date'] = df.index
    df['dayofweek'] = df['date'].dt.dayofweek
    df['quarter'] = df['date'].dt.quarter
    df['month'] = df['date'].dt.month
    df['year'] = df['date'].dt.year
    df['dayofyear'] = df['date'].dt.dayofyear
    df['sin_day'] = np.sin(df['dayofyear'] * (2. * np.pi / 365))
    df['cos_day'] = np.cos(df['dayofyear'] * (2. * np.pi / 365))
    df['dayofmonth'] = df['date'].dt.day
    df['weekofyear'] = df['date'].dt.isocalendar().week
    X = df.drop(['date'], axis=1)
    if target:
        y = df[target]
        X = X.drop([target], axis=1)
        return X, y

    return X

X_train_df, y_train = create_time_features(data, target='Accumulated Acounts Payable')

# Normalize the features
scaler = StandardScaler()
scaler.fit(X_train_df)  # Fit only on training data
X_train = scaler.transform(X_train_df)

X_train_df = pd.DataFrame(X_train, columns=X_train_df.columns)

# Initialize and fit the XGBoost regressor
reg = xgb.XGBRegressor(objective='reg:squarederror', n_estimators=1000)
reg.fit(X_train, y_train, verbose=False)  # Set verbose to True to see training logs
```

Evaluation



```python
# Evaluation function definition
def evaluate(actual, predicted):
    mse = mean_squared_error(actual, predicted)
    mae = mean_absolute_error(actual, predicted)
    rmse = np.sqrt(mse)  # Root Mean Squared Error
    r2 = r2_score(actual, predicted)  # R-squared
    print(f'R2: {r2}')
    print(f'RMSE: {rmse}')
    print(f'MAE: {mae}')
    return {'MSE': mse, 'MAE': mae, 'RMSE': rmse, 'R2': r2}
```