PROJECT FLASH:

INTRODUCING SAC AS AN 1 FOR ALL, ZERO CODE SOLUTION

DEDICATION

Student’s Declaration

1. We hereby declare that this assignment is based on our own work except where acknowledgement of sources is made.
2. We also declare that this work has not been previously submitted or concurrently submitted for any other courses in Sunway University of College or other institutions.

DEDICATION

This research paper is dedicated to those who have journeyed alongside us and encouraged us throughout the entire research. We appreciate your patience, guidance, and support to see this through to its completion.

ACKNOWLEDGEMENTS

Firstly, we would like to thank God for the opportunity to embark on this study. He is the source of our ability to engage in research and complete this endeavor. It has been an incredible journey, equipping ourselves in the field of business analytics and, finally, having the opportunity to apply our newfound skillsets and experiences to this research.

We would also like to extend our sincere thanks to our families for being unwavering pillars of support throughout this research period. Their patience during our long hours of dedicated work has been invaluable.

Lastly, we wish to convey our heartfelt gratitude to Dr. Narishah Mohamed Salleh, Mdm. Azura Zakaria and Dr. Mikkay Wong Ei Leen for their invaluable guidance, unwavering support, and insightful mentorship throughout our Capstone journey. Their dedication and expertise have greatly enriched our academic experience and played an instrumental role in shaping the depth and quality of our research. We truly appreciate their efforts in fostering our growth as scholars and their steadfast commitment to our success.

ABSTRACT

This capstone project evaluates SAP Analytics Cloud (SAC) as a comprehensive no-code analytics platform, focusing on its potential to simplify business analytics processes for students and professionals. Through two use-case scenarios—time-series forecasting and interactive dashboard development—SAC is benchmarked against Python-based traditional solutions. The study employs the Business Analytics Lifecycle (BALC) framework, addressing fragmented workflows and technical barriers in analytics education.

Results demonstrate SAC's ability to deliver competitive predictive accuracy and enhanced usability while reducing technical complexity. Key findings include SAC's intuitive interface, streamlined data preparation, robust model evaluation, and seamless visualization integration. These features establish SAC as a user-friendly, efficient alternative to coding-intensive platforms. The study concludes that SAC not only democratizes analytics by eliminating coding prerequisites but also provides practical tools for real-world applications, fostering a data-driven culture in both academic and professional environments.

Keywords: SAP Analytics Cloud (SAC), Business Analytics Lifecycle (BALC), No-code Solutions, Time-series Forecasting, Data Visualization, Analytics Education

Table of Contents

List of tables

List if Figures

Chapter 1

1.1 Introduction

In today's complex and rapidly evolving landscape of business intelligence and analytics, the challenge lies not only in processing vast amounts of data but in enabling actionable insights through accessible tools. For students, educators, and aspiring business analysts, this complexity often manifests as fragmented systems, steep learning curves, and limited resources. To address these challenges, SAP Analytics Cloud (SAC) offers an integrated, all-encompassing platform that combines data modeling, visualization, and machine learning capabilities into a singular, comprehensive solution.

1.2 Significance of research

This research highlights SAC as an ideal analytics solution for academic and professional environments. The platform's intuitive interface and no-cost model align perfectly with the needs of BA students, enabling them to gain practical, hands-on experience without budgetary constraints. The findings will offer insights into how SAC can streamline the adoption of analytics practices, ensuring equitable access to high-quality tools and fostering a data-literate workforce.

1.3 Problem Statement

The growing complexity of business analytics demands solutions that can seamlessly integrate data modeling, visualization, and predictive modeling into a unified workflow. Traditional tools often compel users to navigate between disparate platforms, each with its own learning curve, technical prerequisites, and financial costs. This fragmented approach creates significant barriers for students, educators, and small-to-medium enterprises (SMEs), who may lack the technical expertise or resources to leverage advanced analytics effectively. The result is inefficiency, missed opportunities for actionable insights, and limited access to powerful data-driven decision-making.

The focus of this study is to explore SAC as a zero-cost, no-code solution designed to bridge the gap between technical proficiency and practical application. By benchmarking SAC against traditional machine learning approaches and platforms through use-cases, this research underscores its potential as a unifying tool that democratizes analytics. SAC's ability to offer robust data modeling, intuitive visualizations, and efficient machine learning within a unified platform positions it as an indispensable resource for upcoming business analysts.

This project addresses the technical and strategic depth required to fully evaluate such a tool. By leveraging SAC's capabilities, we demonstrate how it stands not merely as an alternative but as a comprehensive solution for modern analytics challenges—equipping aspiring analysts and organizations with the tools to thrive in a data-centric world.

Research Questions:

1. How does SAP Analytics Cloud enable cost-effective and scalable business analytics for students and non-technical users?

2. What are the usability advantages of SAC's zero-code design compared to traditional analytics tools?

3. How effectively can SAC integrate predictive modeling and visualization to meet the demands of educational and professional use cases?

1.4 Product Summary

SAP Analytics Cloud (SAC) is a state-of-the-art business intelligence and analytics platform designed to integrate data visualization, predictive analytics, and planning into a unified environment. As a key component of the SAP ecosystem, SAC offers numerous technical benefits, making it an essential tool for businesses aiming to optimize decision-making and enhance operational efficiency.

One of SAC's standout features is its ability to seamlessly integrate with a diverse array of data sources, including both SAP and non-SAP systems such as SAP HANA, SAP BW, Oracle, SQL databases, and cloud-based solutions like Google BigQuery and Amazon Redshift. This capability enables businesses to consolidate data from multiple platforms, ensuring streamlined data management. Additionally, SAC incorporates powerful in-built machine learning capabilities that automate data analysis tasks, such as predictive forecasting and anomaly detection, without requiring advanced coding expertise.

SAC leverages the SAP HANA infrastructure to deliver real-time data processing and visualization, empowering organizations to respond promptly to emerging trends. Its user-friendly interface allows for the creation of interactive dashboards and narrative storytelling through drag-and-drop visualizations, making it easier for users to interpret and present data. Furthermore, as a native SAP application, SAC integrates seamlessly with other SAP solutions like S/4HANA, SAP SuccessFactors, and SAP Ariba, ensuring consistent data governance and cohesive workflows.

Security and compliance are integral to SAC, benefiting from SAP's robust global standards, including data encryption, masking, and adherence to regulations such as GDPR and ISO certifications. This guarantees the safeguarding of enterprise data. Moreover, SAC's cloud-based architecture eliminates the need for on-premises infrastructure, offering accessibility from any device with an internet connection, ensuring flexibility and scalability.

SAC is highly customizable, with pre-built templates and adaptable analytics models that cater to businesses of varying sizes and industries. Its technical strengths enable organizations to centralize their analytics, enhance collaboration, and drive actionable insights, establishing it as a cornerstone for enterprises pursuing digital transformation.

Chapter 2: Literature Review

2.0 Overview

This literature review examines the challenges inherent in traditional analytics workflows and evaluates the capabilities of SAP Analytics Cloud (SAC) as a contemporary solution. It synthesizes existing knowledge, methodologies, and trends in the domains of data preprocessing, machine learning, and visualization. The goal is to identify the most effective tools, techniques, and frameworks that align with the project's objectives and provide a robust theoretical foundation for implementation.

This chapter emphasizes how SAC overcomes technical and accessibility barriers, streamlining data processing, automation, and actionable insights. It establishes the relevance of SAC in bridging the gap between traditional tools and modern analytics requirements, especially for non-technical users such as current students and fresh business analysts.

2.1 Domain

2.1.1 Theoretical Foundations of No-Code Analytics Platforms

Table 2. 1 Summary of Theoretical Foundations of No-Code Analytics Platforms

Literature	**Literature Summary**	**Critical Thoughts**
Davis (1989)	TAM explains adoption based on perceived ease of use and usefulness.	SAC aligns with TAM, promoting adoption by non-technical users.
Norman (2013)	HCI emphasizes user-centered design principles.	SAC's design principles reflect HCI, enhancing user experience.

Diakopoulos (2020)	Data democratization allows broader access to analytics.	SAC embodies data democratization by providing no-code solutions.

The theoretical foundations of no-code platforms emphasize three primary principles—Technology Acceptance Model (TAM), Human-Computer Interaction (HCI), and Data Democratization—as summarized in **Table 2.1.**

The Technology Acceptance Model (TAM) proposed by Davis (1989) underscores the importance of ease of use and perceived usefulness in technology adoption. This framework provides the rationale for designing tools like SAP Analytics Cloud (SAC) that focus on usability, especially for non-technical users. By eliminating the need for coding and incorporating drag-and-drop functionalities, SAC embodies the principles of TAM, thereby lowering barriers to adoption in education and business environments. The alignment of SAC's features with TAM ensures broader accessibility and acceptance among users with diverse backgrounds.

Human-Computer Interaction (HCI) principles, highlighted by Norman (2013), focus on creating intuitive interfaces that cater to users' natural interactions with technology. SAC applies these principles through its interactive dashboards and automated workflows, simplifying complex analytics tasks. This user-centric approach enhances productivity and reduces the cognitive load on users, making SAC a highly effective tool for non-experts who need to navigate advanced analytics.

Lastly, Data Democratization, as discussed by Diakopoulos (2020), aims to provide equitable access to analytics tools, empowering users at all technical levels. Traditionally, analytics workflows required extensive coding expertise, but SAC eliminates these requirements by integrating automation and pre-built models. This democratization aligns with the increasing demand for inclusivity in analytics, ensuring that both students and professionals can make data-driven decisions efficiently.

These theoretical underpinnings validate SAC's design as an accessible and intuitive platform, equipping organizations to transition towards data-centric decision-making with minimal training requirements. By incorporating TAM, HCI, and Data Democratization principles, SAC provides a robust foundation for enabling widespread adoption in diverse sectors.

2.1.2 Key Pillars of a Business Analyst

Table 2. 2 Literature Review on the Key Pillars of a Business Analyst

Name of Literature	**Short Summary**	**Critical Thoughts**
Ying Li (2021)	Discusses the essential skills for business analysts, including visualization, data modeling, and ML modeling.	Raises the question of whether current pedagogical and professional practices adequately prepare analysts to integrate these skills effectively.
Orgut et al. (2020)	Explores the growing demand for advanced modeling capabilities in analytics.	Critically examines the scalability of modeling tools to meet complex and evolving demands while remaining interpretable and accessible.
Chaudhary & Mehta (2021)	Identifies barriers faced by non-technical users in adopting advanced analytics tools.	Highlights the tension between accessibility and analytical rigor, underscoring the necessity of systems that empower diverse

		user bases without oversimplifying analytical depth.

The evolving landscape of analytics has redefined the role of business analysts, positioning them at the nexus of technical and strategic domains. Their ability to synthesize data into actionable insights demands mastery in three core areas: data visualization, data modeling, and machine learning modeling. These competencies form the foundation upon which modern decision-making processes are built.

Ying Li (2021) underscores the multidimensional nature of business analysis, noting that success requires analysts to seamlessly blend technical proficiency with business acumen. However, a persistent challenge lies in ensuring that these skills translate into practical, high-impact applications, particularly within the constraints of organizational contexts.

Orgut et al. (2020) delve into the increasing complexity of analytical workflows, driven by the proliferation of advanced modeling techniques. While these tools expand the scope of actionable insights, they also introduce a degree of complexity that must be managed to ensure usability and scalability without compromising analytical integrity.

Chaudhary and Mehta (2021) bring attention to the barriers faced by non-technical users, emphasizing the need for frameworks and methodologies that democratize access to analytics. Their critique highlights an inherent tension: simplifying tools for broader accessibility must not dilute the analytical rigor necessary for robust decision-making.

Together, these works converge on a critical insight: the role of the business analyst is not merely to perform technical tasks but to strategically interpret and communicate insights that drive meaningful outcomes. The integration of visualization, data modeling, and machine learning modeling represents the triad of capabilities that distinguish business analysts as enablers of data-driven transformation. These three pillars collectively underpin the analytical rigor and contextual relevance essential for addressing contemporary organizational challenges.

2.2 Problem formulation

2.2.1 The Cost Barrier in Learning Analytics Skills

Table 2. 3 Summary of Cost Barriers in Learning Analytics Skills and SAC's Free Solution

Literature	Literature Summary	Critical Thoughts
Verma et al. (2022)	Highlights that premium analytics platforms like Tableau and Power BI have licensing fees, restricting accessibility for students and small businesses.	SAC's free tier enables learning and practical application without financial strain, making it a preferred choice for beginners.
Smith et al. (2021)	Explores how the high cost of traditional tools creates barriers for learning advanced analytics skills.	SAC provides cost-effective access to features like ML modeling and visualization, leveling the playing field for underfunded institutions.
SAP (2023)	Discusses SAC's free access tier, which includes advanced functionalities such as ML capabilities, dashboards, and data integration tools.	SAC's approach democratizes access to enterprise-grade analytics tools, enabling learning at no cost.

Cost is one of the most significant barriers to learning advanced analytics skills, particularly for students, small businesses, and underfunded institutions. As summarized in **Table 2.3**, premium platforms such as Tableau and Power BI require substantial licensing fees for full access to their features, limiting their accessibility to those with financial resources.

Verma et al. (2022) highlights that this high cost creates disparities in skill acquisition, particularly for students and small organizations with tight budgets. Similarly, Smith et al. (2021) emphasize that the financial burden of using traditional tools can delay or even prevent individuals from learning advanced analytics techniques. In contrast, SAP Analytics Cloud (SAC) addresses these issues by providing a free-tier solution that includes essential functionalities such as data visualization, machine learning modeling, and dashboard creation.

The free access offered by SAC ensures that users from all backgrounds can learn and apply analytics concepts without financial constraints. For example, students can develop practical, hands-on skills in machine learning and real-time data visualization, while educators can integrate SAC into curricula without worrying about affordability. This levels the playing field and aligns with the broader goal of democratizing analytics.

Additionally, SAP (2023) demonstrates that SAC's free-tier offering includes enterprise-grade features often found only in costly premium tools. This provides small businesses and learners with the tools needed to compete in data-driven industries. By addressing cost barriers, SAC not only enhances accessibility but also encourages innovation and skill-building across a diverse user base.

This makes SAC particularly relevant for educational institutions, where budget constraints often prevent access to expensive analytics platforms. Its free-tier solution promotes inclusivity and accessibility, aligning with the goals of data democratization and equitable learning opportunities.

2.2.2 The Disparity of Systems for Students

Table 2. 4 Summary of Workflow Challenges and Accessibility in Advanced Analytics Tools

Name of Literature	**Short Summary**	**Critical Thoughts**
Gupta & Joshi (2022)	Examines the fragmentation in Python-based workflows and its impact on real-time data analysis.	Highlights SAC's unified platform as a solution to fragmented tools.
Ahmad & Abu Bakar (2023)	Discusses barriers faced by students in accessing advanced analytics tools due to technical complexity.	Validates SAC’s accessibility for non-technical users.
Rahman et al. (2020)	Explores integration challenges in traditional workflows, especially for real-time analytics.	Demonstrates how SAC’s seamless integration addresses these challenges.

Students face significant challenges when using fragmented analytics systems, such as Python and Excel, which require multiple tools to perform tasks. Gupta and Joshi (2022) highlight how these fragmented workflows increase complexity and reduce efficiency, often making real-time data analysis impractical. This fragmentation not only slows down the analytics process but also creates a significant barrier for students who lack technical expertise in coding or tool integration.

Ahmad and Abu Bakar (2023) emphasize that the technical barriers posed by traditional analytics tools are particularly daunting for students who do not have a background in computer science. SAC addresses these challenges by consolidating functionalities such as data preprocessing, modeling, and visualization into a single, intuitive platform. Its no-code interface reduces the cognitive and technical burden on students, allowing them to focus on interpreting data rather than struggling with tool integration.

Rahman et al. (2020) further validate SAC's ability to seamlessly integrate real-time data, positioning it as an essential tool for modern education. Unlike traditional tools, SAC provides students with a hands-on experience that mirrors real-world applications, enhancing their learning outcomes. This integrated approach enables students to develop comprehensive analytical skills while overcoming the disparities that hinder effective analytics education.

2.2.3 Technical Difficulty of Python

Table 2. 5 Summary of the Technical Difficulty of Python

Name of Literature	Short Summary	Critical Thoughts
Gupta & Joshi (2022)	Examines the fragmentation in Python-based workflows and its impact on real-time data analysis.	Critiques the inefficiency and learning challenges introduced by fragmented workflows, advocating for more cohesive analytical ecosystems.
Ahmad & Abu Bakar (2023)	Discusses barriers faced by students in accessing advanced analytics tools due to technical complexity.	Highlights the necessity for user-centric design in analytics tools to bridge the gap between technical complexity and accessibility.
Rahman et al. (2020)	Explores integration challenges in traditional workflows, especially for real-time analytics.	Emphasizes the value of seamless integration in analytics systems, which can facilitate more effective educational and analytical outcomes.

Python has long been a favored tool in the analytics community for its flexibility and extensive library support. However, as highlighted in **Table 2.5**, its high technical demands present significant challenges for students and non-technical users.

Gupta and Joshi (2022) emphasize that Python's steep learning curve acts as a major obstacle, particularly for individuals without a programming background. Learning Python requires not only an understanding of programming fundamentals but also proficiency in syntax, debugging, and library management. This can be intimidating for students who are new to analytics, leading to slower adoption rates or outright abandonment of analytics training programs. In contrast, SAP Analytics Cloud (SAC) addresses this issue by providing a **no-code platform** where users can perform analytics tasks without writing a single line of code. By eliminating the need to learn programming, SAC enables students and non-technical users to focus on interpreting data and deriving insights.

Ahmad and Abu Bakar (2023) further highlight that the complexity of coding-based platforms like Python discourages many users from pursuing analytics education or projects. The challenge lies not just in mastering the language but also in understanding advanced workflows such as machine learning modeling and data integration. SAC resolves this issue through **pre-built machine learning models** and **automated workflows** that guide users step-by-step. For instance, SAC's **Smart Predict** tool allows users to build predictive models without needing any prior knowledge of machine learning algorithms or coding.

Additionally, Rahman et al. (2020) identify inefficiencies and errors as common issues in Python-based workflows. Manual scripting is inherently prone to human error, particularly for non-technical users who may not fully understand debugging processes. SAC mitigates these risks through automation and integrated tools for data preparation, visualization, and modeling. By automating repetitive tasks and minimizing user intervention, SAC reduces the likelihood of errors and enhances workflow efficiency.

2.3 Methodology framework

2.3.1 How does BALC fare against other framework for our project?

Table 2. 6 Methodology framework Literature Discussion

Framework	**Literature Title & Summary**	**How It Relates to Us**
MLOps	*"Machine Learning Operations (MLOps): Principles and Practices"* highlights how MLOps streamlines the deployment and monitoring of ML models. It emphasizes CI/CD pipelines, model retraining, and scalability to ensure iterative improvement through continuous delivery.	While MLOps is essential for model deployment, its technical complexity and dependence on ML expertise make it less applicable for our analytics-focused BALC approach, which prioritizes simplicity and broader accessibility.
DevOps	*"The Role of DevOps in Modern Software Development"* (SelectHub) describes how DevOps focuses on collaboration between development and operations teams, streamlining delivery pipelines, automation, and testing for scalability and iterative application deployment.	DevOps' emphasis on collaboration and automation resonates with our project. However, BALC's specific focus on analytics lifecycle, including data modeling and visualization, makes it more relevant to our context.
	BALC	
	"The Analytics Lifecycle Management Framework" (Towards Data Science)	BALC aligns perfectly with our project's objectives,

	discusses how BALC structures analytics workflows into clear phases—data preparation, modeling, deployment, and monitoring. It integrates data and analytics needs, making it ideal for business intelligence and analytics.	offering a structured methodology to manage the entire lifecycle of analytics projects while ensuring accessibility, iterative development, and seamless integration with visualization tools.

The project’s decision to adopt BALC is based on its suitability for managing the entire lifecycle of analytics workflows (Ying Li, 2021). BALC's distinct phases provide a structured roadmap that aligns with the project’s objectives, particularly in addressing the inefficiencies of fragmented analytics tools. In contrast, MLOps focuses primarily on technical model deployment and monitoring, which requires advanced ML expertise and is less applicable here (Towards Data Science, 2021). Similarly, while DevOps fosters efficiency in software development, it lacks the specific integration of analytics components like visualization and data modeling, which are essential for this project (SelectHub, 2020).

Chapter 3 will operationalize BALC by focusing on the four most relevant phases—**Problem Formulation**, **Solution Design**, **Solution Development**, and **Success Measurement**. These phases will guide the methodology to ensure clarity in problem identification, solution implementation, and evaluation. Phases such as Solution Deployment, Maintenance, and Sunset are excluded because they extend beyond the academic scope and pertain to post-project lifecycle management.

2.3.2 BALC FRAMEWORK and its Application

Table 2. 7 Summary of BALC Framework and Its Application

Literature	Literature Summary	Critical Thoughts
Li (2021)	Describes BALC as a structured methodology for managing analytics workflows in iterative phases.	SAC aligns with BALC by integrating analytics processes into a single platform, supporting iterative development.
Towards Data Science (2021)	Highlights the importance of clarity and structured transitions in analytics projects as emphasized by BALC.	SAC's organized workflow aligns with BALC, promoting efficiency and collaboration.
SelectHub (2020)	Compares BALC to CRISP-DM and Agile frameworks, noting BALC's flexibility for both educational and business use.	SAC's compatibility with BALC makes it ideal for a wide range of projects, including educational use cases.

The **Business Analytics Lifecycle (BALC)** framework provides a structured methodology for managing analytics projects. As outlined in **Table 2.7**, BALC emphasizes clarity, iterative development, and end-to-end project management, making it particularly relevant for integrating tools like SAP Analytics Cloud (SAC).

Li (2021) identifies BALC as a systematic approach designed to manage analytics workflows in iterative phases: **Problem Formulation**, **Solution Design**, **Solution Development**, and **Success Measurement**. This structure ensures that each phase of an analytics project is well-defined and contributes to achieving overall objectives. SAC seamlessly supports this lifecycle by offering tools for data preparation, machine learning modeling, and visualization within a unified platform. For instance, during the **Problem Formulation** phase, SAC's **collaborative features** enable teams to define project goals and share initial findings, fostering alignment among people of interest in our project.

The importance of structured transitions between project phases, as emphasized by Towards Data Science (2021), is another critical aspect of BALC. In traditional workflows, transitioning between phases often involves switching tools or manually exporting and re-importing datasets, leading to inefficiencies and potential errors. SAC eliminates these pain points by integrating all analytics functions into a single platform. This not only reduces delays but also ensures that insights and models developed in earlier phases can be directly applied to subsequent ones without additional manual intervention.

SelectHub (2020) compares BALC with other popular analytics frameworks such as CRISP-DM and Agile Analytics. While CRISP-DM provides a structured process for data mining projects, it lacks the flexibility required for dynamic business environments. Agile Analytics, on the other hand, focuses on iterative development but often requires substantial coordination among stakeholders, which can be challenging in resource-constrained settings. BALC strikes a balance between structure and flexibility, making it particularly suited for educational and SME use cases. SAC's features, such as **real-time collaboration** and **automated data pipelines**, align closely with BALC's requirements, enabling users to move through the lifecycle phases efficiently.

2.4 Data Understanding: Visualization and Preliminary Analysis

Table 2. 8 Summary of Visualization Techniques for Data Understanding

Name of Literature	**Short Summary**	**Critical Thoughts**
Zhang et al. (2020)	Discusses the use of histograms to analyze data distributions effectively.	SAC's histogram feature simplifies exploratory data analysis for non-technical users.
Cheng et al. (2023)	Highlights box-plots as critical for identifying outliers and summarizing data variability.	SAC integrates boxplots, enabling quick identification of outliers without complex coding.
Gupta & Joshi (2022)	Describes line charts as essential tools for visualizing trends and patterns in time-series data.	SAC's line chart capabilities enhance time-series analysis, providing real-time trend insights.

Understanding data is a crucial first step in any analytics workflow, as it sets the foundation for effective modeling and decision-making. As summarized in **Table 2.8**, visual techniques such as **histograms**, **box-plots**, and **line charts** play a vital role in this process by enabling analysts to explore and interpret data efficiently.

Histograms, as highlighted by Cheng et al. (2023), are particularly effective for analyzing data distributions. They allow users to visualize how data points are spread across different ranges, making it easier to identify skewness, central tendencies, and potential anomalies. SAP Analytics Cloud (SAC) provides an intuitive interface for generating histograms, enabling users to perform this analysis without requiring advanced technical skills. For example, a retail analyst could use histograms in SAC to understand sales distribution across different product categories, uncovering patterns that may warrant further investigation.

Box-plots, discussed by Gupta and Joshi (2022), are essential for identifying outliers and summarizing variability in datasets. Outliers often indicate data errors or unique trends that can significantly impact analytics outcomes if left unaddressed. SAC's built-in box-plot functionality simplifies this process by automating outlier detection and visualization. This capability is particularly valuable for non-technical users who may not have the expertise to manually identify anomalies in raw data.

Line charts, as described by Zhang et al. (2020), are indispensable tools for analyzing time-series data. They enable users to observe trends, patterns, and fluctuations over time, providing insights that are critical for forecasting and decision-making. SAC enhances time-series analysis with real-time line chart visualization, allowing users to identify trends dynamically as new data is added. For instance, a business analyst could use SAC to monitor sales performance over multiple quarters, identifying seasonal trends and informing future marketing strategies.

By integrating these visualization techniques into a single platform, SAC ensures that users can explore their data comprehensively without switching between tools or requiring additional training. The automation of tasks such as outlier detection and trend identification not only saves time but also reduces the likelihood of errors, enabling users to focus on deriving actionable insights.

In summary, SAC's visualization capabilities address the challenges traditionally associated with data understanding, particularly for non-technical users. By making advanced visual analytics accessible and intuitive, SAC empowers organizations and individuals to extract meaningful insights from their data, paving the way for effective modeling and decision-making.

2.5 Time-Series Forecasting Algorithms

Table 2. 9 Summary of Time-Series Forecasting Algorithms

Literature	**Literature Summary**	**Critical Thoughts**
Zhang et al. (2020)	ARIMA works well for stationary data but struggles with non-linear patterns and seasonality.	SAC’s integration of LSTM addresses these limitations, making it suitable for complex datasets.
Ahmad & Abu Bakar (2023)	LSTM excels in capturing sequential dependencies and non-linear patterns in time-series data.	SAC leverages LSTM for accurate time-series forecasting, providing non-technical users with advanced analytics capabilities.
SAP Community (2023)	Explains how LSTM is integrated into SAC for time-series forecasting, enabling accurate predictions with minimal configuration.	SAC’s use of LSTM enhances forecasting accuracy, even in complex, dynamic environments.

Time-series forecasting is a critical component of analytics, particularly for industries that rely on predicting future trends and patterns. As outlined in **Table 2.9**, algorithms such as **ARIMA** and **Long Short-Term Memory (LSTM)** networks play pivotal roles in this domain, with SAC adopting LSTM to overcome the limitations of traditional methods.

ARIMA (Autoregressive Integrated Moving Average) is widely used for time-series forecasting due to its ability to model stationary data effectively. Zhang et al. (2020) highlight that ARIMA works well for datasets with consistent patterns over time but struggles with non-linearity and seasonality, which are common in real-world scenarios. For instance, retail sales data often exhibit seasonal trends that ARIMA cannot fully capture without extensive preprocessing. In contrast, SAC's integration of LSTM eliminates these limitations by automatically handling non-linear relationships and capturing long-term dependencies within data.

LSTM, as described by Ahmad and Abu Bakar (2023), is a specialized type of recurrent neural network (RNN) designed to model sequential data effectively. Unlike traditional algorithms, LSTM can learn and remember patterns over long time horizons, making it ideal for forecasting dynamic and complex datasets. SAC leverages LSTM in its **Smart Predict** feature, allowing users to create accurate forecasts without requiring technical expertise in machine learning. For example, a supply chain manager could use SAC to predict future inventory needs based on historical demand, reducing costs associated with overstocking or understocking.

The integration of LSTM into SAC is further elaborated by SAP Community (2023), which emphasizes its ability to simplify forecasting workflows. By automating key steps such as feature engineering, hyperparameter tuning, and model evaluation, SAC ensures that even users with minimal technical backgrounds can generate reliable predictions. This accessibility aligns with SAC's broader goal of democratizing advanced analytics and making it available to a diverse audience.

Moreover, SAC's visualization tools complement its forecasting capabilities by allowing users to analyze prediction outputs in real time. For instance, forecast results can be displayed alongside historical data using **interactive dashboards**, enabling stakeholders to make informed decisions quickly.

In summary, SAC's adoption of LSTM for time-series forecasting represents a significant advancement in making complex analytics techniques accessible to non-technical users. By addressing the limitations of traditional methods like ARIMA and providing an intuitive, automated workflow, SAC ensures that users can derive accurate forecasts and actionable insights, regardless of their technical expertise.

2.6 Model Evaluation – RMSE

Table 2. 10 Summary of RMSE as a Model Evaluation Metric

Literature	**Literature Summary**	**Critical Thoughts**
Chai & Draxler (2014)	RMSE is one of the most commonly used metrics to measure the accuracy of forecasting models. It represents the square root of the average squared differences between predicted and observed values.	RMSE is ideal for assessing SAC's time-series forecasting models, as it quantifies prediction accuracy in an easily interpretable format.
Zhang et al. (2020)	RMSE effectively captures the magnitude of errors, making it a standard metric for comparing different forecasting models.	SAC's automated RMSE calculations allow non-technical users to evaluate model performance effortlessly.

2.6 Model Evaluation – RMSE

SAP Community (2023)	RMSE is integrated into SAC's Smart Predict tool, providing users with instant feedback on the performance of predictive models.	SAC's inclusion of RMSE enhances user confidence in the accuracy of forecasts, supporting better decision-making.

Model evaluation is a crucial aspect of any analytics workflow, as it determines the reliability and accuracy of predictive models. Among the various metrics used for this purpose, **Root Mean Square Error (RMSE)** is widely recognized for its ability to quantify prediction accuracy, as summarized in **Table 2.10**.

Chai and Draxler (2014) describe RMSE as a measure of the difference between predicted and observed values. By taking the square root of the average squared differences, RMSE provides an easily interpretable metric that reflects the magnitude of prediction errors. This makes it particularly useful for time-series forecasting, where small deviations can significantly impact decision-making. For instance, in inventory management, an RMSE of 5 units may indicate acceptable variance, while larger deviations may signal the need to refine forecasting models. SAC integrates RMSE as a standard metric in its **Smart Predict** tool, enabling users to evaluate model performance quickly and accurately.

Zhang et al. (2020) highlight that RMSE is not only effective for assessing individual models but also for comparing the performance of different algorithms. This is especially valuable in SAC, where users can test various forecasting approaches, such as ARIMA and LSTM, and rely on RMSE to identify the most accurate model. By automating RMSE calculations, SAC eliminates the need for manual computation, making it accessible to users who may lack technical expertise.

SAP Community (2023) further elaborates on how RMSE is seamlessly integrated into SAC's predictive analytics workflow. Users receive real-time feedback on model performance, allowing them to refine parameters or select alternative models as needed. This iterative approach enhances the accuracy of forecasts and ensures that users can trust the outputs of their analytics processes.

Moreover, RMSE's intuitive nature makes it a preferred metric for stakeholders who may not have a deep understanding of analytics. Unlike more complex metrics, such as R^2 or Mean Absolute Error (MAE), RMSE directly reflects the scale of errors in the same units as the predicted values, making it easier for decision-makers to interpret. For example, a sales manager can immediately understand that an RMSE of \$1,000 indicates an average error of \$1,000 in revenue forecasts.

2.7 Comparative Analysis of No-Code and Traditional Platforms

Table 2. 11 Summary of Comparative Analysis of No-Code and Traditional Platforms

Literature	**Literature Summary**	**Critical Thoughts**
SAP (2023)	SAC integrates data visualization, machine learning, and reporting into a unified platform, simplifying workflows.	SAC's all-in-one platform reduces dependency on multiple tools, making it cost-effective and user-friendly.
Tableau Community (2022)	Tableau offers advanced visualizations but requires external tools for machine learning and data integration.	SAC's integration of ML tools provides a competitive advantage over visualization-centric platforms like Tableau.

Power BI (2023)	Power BI excels in reporting but lacks advanced predictive analytics capabilities, requiring external integrations for ML.	SAC’s predictive analytics capabilities surpass Power BI by offering no-code machine learning models.

The analytics landscape includes a diverse range of platforms, each with unique strengths and limitations. As summarized in **Table 2.11**, SAP Analytics Cloud (SAC) distinguishes itself through its integrated approach, addressing key gaps in traditional platforms like Tableau and Power BI.

SAP (2023) highlights SAC’s ability to integrate data visualization, machine learning, and reporting into a single unified platform. Traditional tools often require users to switch between multiple applications for different tasks, such as using Tableau for visualization, Python for machine learning, and Excel for reporting. This fragmented approach can lead to inefficiencies and increased complexity. SAC resolves these issues by providing an all-in-one solution, enabling users to perform end-to-end analytics workflows without leaving the platform. For example, a marketing analyst can import campaign data, apply predictive models, and generate actionable reports—all within SAC.

Tableau, as described by Tableau Community (2022), is renowned for its advanced visualization capabilities. However, it lacks built-in tools for machine learning and data integration, requiring users to rely on external platforms to perform predictive analytics. This limitation increases the learning curve and costs associated with implementing Tableau in organizations that require advanced analytics. In contrast, SAC offers **no-code machine learning tools** such as **Smart Predict**, which simplify predictive analytics for users with minimal technical expertise. This integration allows SAC to provide not only visual insights but also prescriptive recommendations, giving it a significant advantage over Tableau.

Power BI, another popular analytics tool, excels in data connectivity and reporting, as noted by Power BI (2023). Its ability to integrate with Microsoft’s ecosystem makes it a strong choice for reporting-focused organizations. However, Power BI lacks robust machine learning capabilities, requiring users to integrate it with external tools such as Azure Machine Learning or Python for advanced analytics. SAC surpasses Power BI in this regard by offering **built-in ML models** that are accessible directly within the platform. This feature allows users to develop predictive insights without the need for additional software or technical skills.

Additionally, SAC's accessibility and affordability further strengthen its position in the market. While Tableau and Power BI often require expensive licensing fees, SAC provides a **free-tier option**, as discussed in Section 2.2, making it an attractive choice for students, small businesses, and educational institutions. This democratization of analytics ensures that users from diverse backgrounds can leverage SAC's capabilities without financial constraints.

2.7 Summary

This chapter demonstrates how SAP Analytics Cloud addresses the challenges of traditional analytics workflows through automation, integration, and accessibility. By aligning with frameworks like BALC and incorporating advanced features such as LSTM and machine learning automation, SAC transforms analytics into an efficient and user-friendly process. Through critical analysis of literature and detailed evaluation of SAC's capabilities, this chapter highlights its potential to revolutionize modern analytics practices.

Chapter 3: Methodology

3.1 Overview

In this chapter, the methodology utilized to achieve the objectives outlined in Chapter 1 and deliver the outputs detailed in Chapter 4 is discussed. The methodology is broken down into several stages: project formulation, solution design, solution development, and success measurement. Each stage is described in detail to provide a clear understanding of the approach and rationale behind the methods employed.

3.2 Business Analytics Lifecycle (BALC)

Based on the research completed in the Literature Review section, this project adopts the Business Analytics Life Cycle (BALC) framework. According to Ying Li (2021), BALC traditionally comprises seven stages: Problem Formulation, Solution Design, Solution Development, Solution Deployment, Success Measurement, Solution Maintenance and Enhancement, and Solution Sunset.

For this project, we focus on four key phase shown in **Fig 3.1**: Problem Formulation, Solution Design, Solution Development, and Success Measurement. The final three stages before Success Measurement are omitted as they pertain to long-term maintenance and eventual retirement of solutions, which extend beyond the scope of this capstone project's immediate objectives of evaluating SAP Analytics Cloud (SAC) as a comprehensive analytics solution.

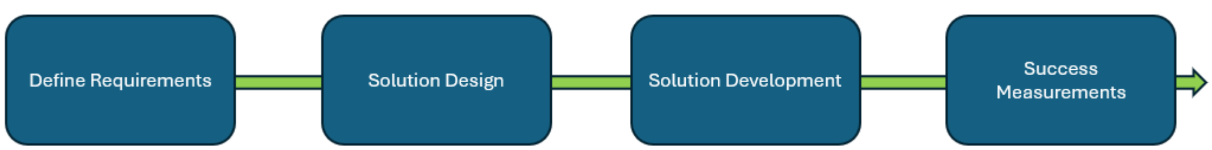


Fig 3. 1 Business Analytics Lifecycle

Table 3. 1 BALC Phases Details

Phase	Explanation
Problem Formulation	This phase involves identifying the core challenges and defining the objectives of the project. In this case, the problem was the inefficiency caused by fragmented tools for analytics and the need for an integrated approach to simplify analytics workflows for business analysts and non-technical users at **ZERO COST**.
Solution Design	During this phase, the framework for addressing the identified problem was developed. For this project, this included designing two distinct use cases—time-series forecasting and dashboard visualization—to evaluate the strengths of the chosen tools against traditional methods like Python or Power BI.
Solution Development	The implementation of the proposed solution took place here. This included building time-series forecasting model and creating comprehensive dashboards for business analytics. BALC guided this by ensuring the solution was iteratively refined to align with the objectives.
Success Measurement	This phase focused on evaluating the effectiveness of the developed solutions. Key performance indicators such as accuracy, usability, and integration were assessed. Comparisons were made between traditional analytics tools and the proposed unified approach, ensuring that the solution met the project's goals.

The three omitted phases—**Solution Deployment**, **Solution Maintenance and Enhancement**, and **Solution Sunset**—were excluded as they pertain to long-term implementation and lifecycle management, which extend beyond the academic scope of this capstone project. These phases are typically relevant for projects aimed at operationalizing solutions in real-world environments over extended periods. Since this project is focused on evaluating the feasibility and effectiveness of a proposed solution within an academic context, these later lifecycle stages are not applicable. By focusing on the four selected phases, the project ensures that its scope remains manageable and directly aligned with its research objectives.

3.3 Solution Design

The solution design stage outlines the conceptual framework for addressing the problem identified during the project formulation stage. As illustrated in Figure 3.2, the design focuses on two use case scenarios conducted using SAP Analytics Cloud (SAC) and each use-case scenario's respective traditional solution. Each step in the process is implemented using both tools to facilitate a detailed comparative analysis.

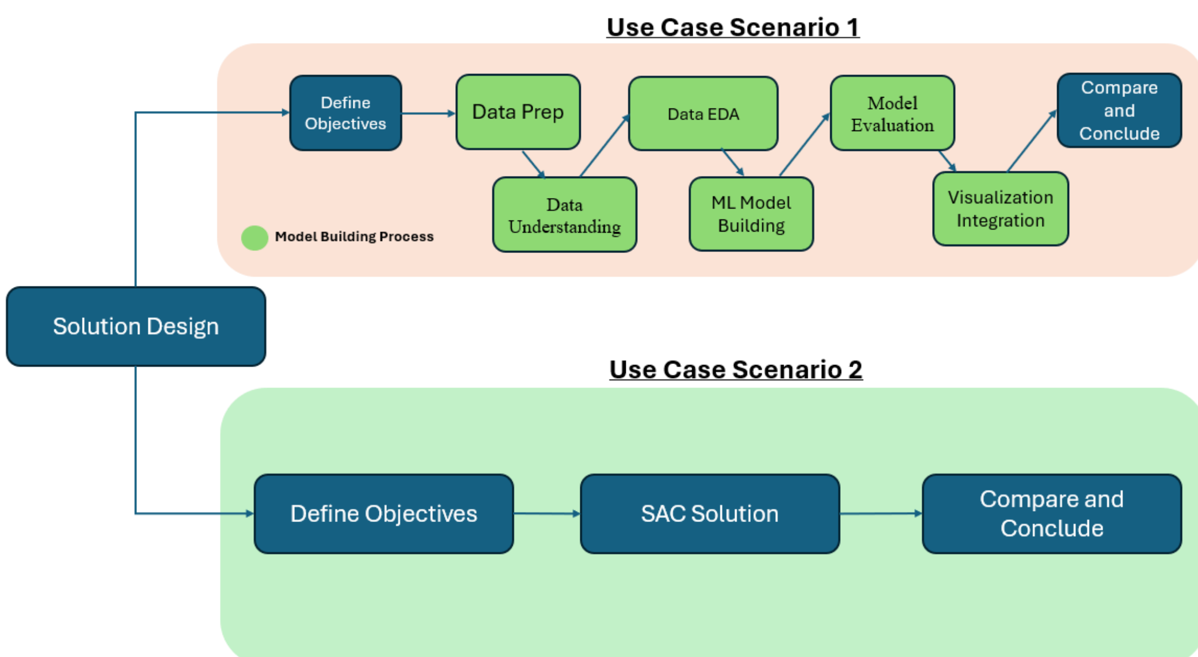


Fig 3. 2 Solution Design Overview

As illustrated in the flowchart in Fig 3.2, the process begins with defining specific objectives for each use case. In Use Case Scenario 1, the goal was to compare the traditional solution (LSTM) with SAP Analytics Cloud (SAC) for stock price prediction. Similarly, Use Case Scenario 2 focused on evaluating SAC's capability to generate dashboards for company performance visualization.

Table 3. 2 Solution Design Breakdown UCS-1

Step No.	Description
1	**Defining Objectives:** Establishing the goal of creating an accurate predictive model for stock price forecasting.
2	**Data Preparation:** Cleaning and transforming the dataset to ensure compatibility with Python and SAC.
3	**Data Understanding:** Plotting the data spread through SAC and Python code.
4	**Exploratory Data Analysis (EDA):** Using Python and SAC to plot line chart
5	**Model Development:** Building and training a time-series model using Python's LSTM algorithm and SAC's built-in machine learning framework.
6	**Model Evaluation:** Comparing the performance metrics (e.g., accuracy, RMSE) of the models generated in Python and SAC.

Table 3. 3 Solution Design Breakdown UCS-2

Step No.	Description
1	**Defining Objectives:** Identifying key performance indicators (KPIs) and visualization goals.
2	**Data Preparation:** Preparing the dataset for dashboard creation.
3	**Visualization Development:** Creating dashboards in SAC using its drag-and-drop interface.
4	**Visualization Evaluation:** Evaluating the user experience, customization capabilities, and interactivity of dashboards created in the platform.

Through this methodology, the objectives outlined in Chapter 1 are addressed systematically. By employing both Python and SAC for each use case, the study evaluates the capabilities of SAC as a unified solution based on the core pillars of this assignment: usability, integration, efficiency, and accuracy.

By addressing these key pillars, the solution design ensures a comprehensive evaluation of SAC as a viable alternative to traditional Python-based approaches. This structured approach lays the foundation for the subsequent stages of solution development and success measurement.

3.4 Solution Development

The solution development part will contain the breakdown of the structure of our solution and it’s detailed steps. This section outlines the step-by-step implementation of the solution using Python and SAP Analytics Cloud (SAC) for both use-case scenarios.

3.4.1 Use-Case scenario 1

Define Objectives

Table 3. 4 Defining Objectives

Objective 1	Does SAC provide competitive accuracy and reliability in time-series forecasting compared to Python’s advanced models like LSTM, and are the results sufficient for making business decisions?
Objective 2	How well does SAP Analytics Cloud (SAC) simplify the process of creating a time-series forecasting model compared to Python, especially for a business analyst with limited technical knowledge?
Objective 3	To what extent does SAC facilitate the integration of time-series forecasts into business workflows, including visualization and stakeholder communication, compared to Python?

Objective 1: Evaluate Predictive Accuracy

The first objective is to determine **how accurately SAP Analytics Cloud (SAC) can predict time-series data compared to Python's Long Short-Term Memory (LSTM) models**. Predictive accuracy is assessed using metrics such as RMSE, MAPE, and R^2. This objective focuses on ensuring that SAC's predictions are robust and reliable enough for real-world decision-making. By benchmarking SAC against Python, this evaluation directly addresses SAC's capability as a competitive analytics platform.

Objective 2: Assess Usability for Non-Technical Users

The second objective evaluates **how SAC simplifies the process of creating a time-series forecasting model compared to Python**, particularly for business analysts with limited technical expertise. SAC's no-code interface and pre-built tools will be analyzed for their ability to streamline complex processes. This objective ensures that the usability of SAC aligns with its positioning as a user-friendly tool, reducing the learning curve for business analysts.

Objective 3: Examine Workflow Integration

The third objective assesses **how effectively SAC integrates time-series forecasts into business workflows**, including visualization and stakeholder communication. This involves evaluating SAC's ability to seamlessly transition from model building to visualization within the same platform. This objective ties the use case to broader business goals, emphasizing SAC's role in supporting decision-making processes and stakeholder engagement.

3.4.2 Structure of approach for Use-case scenario 1

After defining the objectives for this project, it is essential to outline the structure and approach taken for the implementation of the **Time-Series Stock Price Prediction** model. To ensure clarity and consistency for readers, the process will be presented in two distinct sections:

1. **Python-based Workflow**:
   The first section will provide a comprehensive walkthrough of the model development process in Python. This includes steps starting from data preparation, exploratory data analysis (EDA), and model building to model evaluation. The Python implementation will showcase the flexibility, control, and technical depth required to achieve accurate time-series forecasting using a coding-based approach.

2. **SAP Analytics Cloud (SAC) ML building workflow**:
   The second section will mirror the same steps—data preparation, EDA, model building, and evaluation—within SAP Analytics Cloud. This section will highlight SAC's no-code approach and user-friendly interface, demonstrating its ability to streamline the same tasks while reducing the technical expertise required.

3.4.3 Model Building Process

3.4.5 Python

To ensure clarity, the flowchart below, Fig 3.3, provides a summarized overview of the Python model-building process, which comprises six key steps: **Data Preparation**, **Preprocessing**, **Scaling**, **Model Construction**, **Model Training**, and **Model Evaluation**. These steps outline the sequential tasks necessary to build and evaluate a Long Short-Term Memory (LSTM) model for stock price prediction. The entire process, from data preparation to evaluation, took approximately **3 minutes and 07 seconds**, showcasing the efficiency of Python's implementation for this task. Each step will be explored in detail in the following sections.

3.4.3 Model Building Process

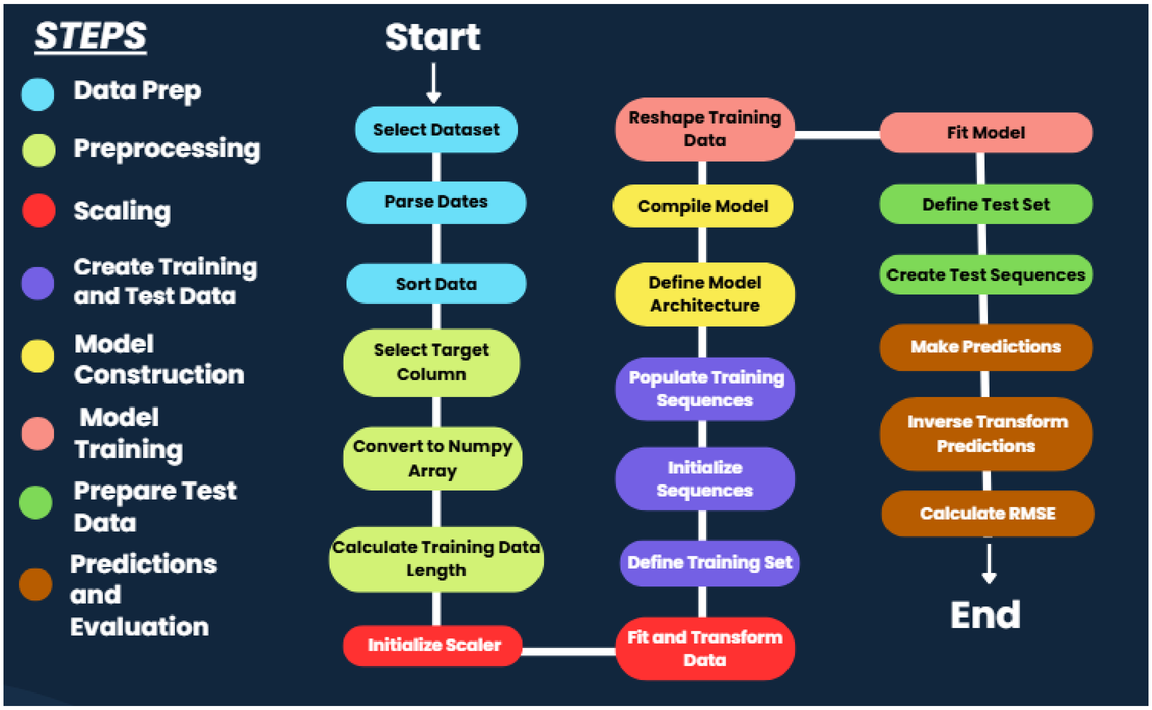


Fig 3. 3 Python Complete Process Overview

3.4.6 Data Preparation

To prepare the data for time-series analysis in Python, a series of data cleaning and preprocessing steps were implemented.

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
nyse_data = pd.read_csv('NYSE.csv', parse_dates=['Date'], index_col='Date')
```

```
[ ] nyse_data = pd.read_csv('NYSE.csv')
    print(nyse_data.columns)  # This will show you what columns are available

Index(['Date', 'Price'], dtype='object')
```

Fig 3. 4 Loading the Dataset into Python

The dataset was loaded into Python using the pandas library, which provides a robust framework for data manipulation. **Fig 3.4** illustrates the Python code for loading the dataset. Loading the dataset into a structured data frame allows efficient manipulation and analysis. Parsing dates while loading ensures that the 'Date' column is treated as a time index, which is a fundamental requirement for time-series forecasting. Setting the 'Date' column as the index ensures that every record corresponds to a unique time point, facilitating time-dependent computations.

```
[ ]  nyse_data['Date'] = pd.to_datetime(nyse_data['Date'])
     nyse_data.sort_values('Date', inplace=True)
```

Fig 3. 5 Formatting and Sorting the Date Column

For time-series models to function correctly, data must follow a sequential order. Any inconsistencies in the date format or an unsorted timeline could result in inaccurate trend detection or erroneous model outputs. The dataset's 'Date' column was formatted into a standard datetime object, and the data was sorted chronologically, as shown in **Fig 3.5**.

```
[ ]  print(nyse_data.duplicated(subset=['Date']).sum())
     print(nyse_data.isna().sum())
```

Fig 3. 6 Identifying Duplicates and Missing Values

Duplicates can distort statistical summaries and inflate the weight of certain time points, leading to biased model predictions. Missing values disrupt mathematical computations, as many machine learning models cannot handle null entries directly. Addressing missing values ensures that all data points contribute effectively to the model's training, reducing the likelihood of errors or inaccuracies. Duplicate and missing entries in the dataset were identified and addressed using duplicated() and isna() methods, as shown in **Fig 3.6**.

3.4.7 Data Understanding

Histogram

```
plt.figure(figsize=(10, 6))
plt.hist(nyse_data['Price'], bins=30, color='skyblue', edgecolor='black')
plt.title('Histogram of Price', fontsize=16)
plt.xlabel('Price', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Fig 3. 7 python code for Histogram of Stock Prices

A histogram was generated to examine the distribution of stock prices, as shown in **Fig 3.7**. The plot divides the range of stock prices into 30 bins, representing the frequency of prices within each range. The histogram provides a visual representation of the dataset's distribution. It helps identify whether the data is normally distributed, skewed, or contains irregularities, such as missing ranges or clusters. By analyzing the frequency distribution, we can determine if the data is evenly spread or concentrated in specific ranges.

Box-Plot

```
plt.figure(figsize=(8, 6))
plt.boxplot(nyse_data['Price'], vert=False, patch_artist=True, boxprops=dict(facecolor='lightgreen'))
plt.title('Boxplot of Price', fontsize=16)
plt.xlabel('Price', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.show()
```

Fig 3. 8 python code for Box Plot of Stock Prices

A box plot was created to detect outliers and summarize the dataset's variability, as shown in **Fig 3.8**. The plot displays the median, interquartile range (IQR), and potential outliers.

Box plots are effective for identifying outliers and understanding the spread of data. Outliers can distort predictions and must be accounted for during model training. Additionally, the IQR highlights the range of typical stock price fluctuations.

This step provides insights into data variability and outlier behavior, ensuring the dataset is consistent and aligned with the assumptions of time-series modeling.

3.4.8 Exploratory Data Analysis

```
import matplotlib.pyplot as plt

plt.figure(figsize=(15, 5))  # Set the figure size as per your preference
plt.plot(nyse_data['Date'], nyse_data['Price'], color='blue')  # Plotting the line chart
plt.title('Price per Date')  # Adding a title
plt.xlabel('Date')  # X-axis label
plt.ylabel('Price')  # Y-axis label
plt.grid(True)  # Adding grid lines for better readability
plt.xticks(rotation=45)  # Rotate date labels for better visibility
plt.tight_layout()  # Adjust layout to make room for label rotation
plt.show()  # Display the plot
```

Fig 3. 9 Python Code for Line Chart of Stock Prices Over Time

The line chart, created using the matplotlib library, plots stock prices against their corresponding dates, as shown in **Fig 3.9**. The x-axis represents the dates, while the y-axis displays the stock prices. The line chart provides a clear temporal view of stock price movements, helping to identify long-term trends, seasonality, and any abrupt changes in the dataset. This visualization is critical for time-series analysis as it highlights patterns that may influence the predictive model.

The line chart is particularly helpful during the **Exploratory Data Analysis (EDA)** stage because it offers a temporal view of the data, which is critical for understanding the behavior of time-series datasets like stock prices. Unlike other types of charts, the line chart excels at visualizing trends and changes over time, making it indispensable for identifying key patterns and irregularities in sequential data.

3.4.9 ML model Building

The model-building process involves constructing a Long Short-Term Memory (LSTM) model for stock price prediction. This process consists of two key stages: **Model Construction** and **Model Training**

```python
nyse_data['Date'] = pd.to_datetime(nyse_data['Date'])
nyse_data.sort_values('Date', inplace=True)
data = nyse_data.filter(['Price'])
dataset = data.values
training_data_len = int(np.ceil( len(dataset) * .95 ))
print("Training data length:", training_data_len)
print("Dataset shape:", dataset.shape)
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)
train_data = scaled_data[0:training_data_len, :]
x_train = []
y_train = []

for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_data[i, 0])
    if i <= 61:
        print(x_train)
        print(y_train)
        print()
```

```python
x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape= (x_train.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(x_train, y_train, batch_size=1, epochs=1)
test_data = scaled_data[training_data_len - 60: , :]
x_test = []
y_test = dataset[training_data_len:, :]
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i, 0])
x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
print("RMSE:", rmse)
```

Fig 3. 10 LSTM Model Construction Training Process Python Code

The construction and training of the LSTM model form the backbone of the predictive modeling process, enabling it to learn sequential patterns and forecast stock prices with precision. The architecture is designed with two LSTM layers—one with 128 units configured to return sequences and another with 64 units—to capture both short-term and long-term dependencies in the data. The sequential structure ensures the model retains temporal relationships while refining its understanding of trends. These layers are followed by two Dense layers, where the final layer outputs a single predicted stock price. While this architecture provides significant flexibility and control, it requires a deep understanding of machine learning and neural networks to design and optimize effectively.

The training process, powered by the Adam optimizer and the Mean Squared Error (MSE) loss function, fine-tunes the model's weights to minimize prediction errors. Adam's adaptive learning rate ensures efficient convergence, while MSE penalizes larger errors more heavily, improving the model's accuracy in predicting stock prices. The dataset is reshaped into a 3D format required by LSTM models, and training is conducted using a batch size of 1 for computational precision.

3.4.10 Model Evaluation

```
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
print("RMSE:", rmse)
```

Fig 3. 11 Python Code for LSTM Model Evaluation Metrics

The final step in evaluating the LSTM model involves calculating the **Root Mean Squared Error (RMSE)**, a widely recognized metric for assessing the accuracy of regression-based models. The RMSE is computed by taking the square root of the mean squared differences between the predicted stock prices and the actual test values. This metric provides a clear and interpretable measure of the model's prediction error, expressed in the same unit as the target variable (e.g., stock price). One of the key advantages of RMSE is its sensitivity to larger errors, as it penalizes them more heavily due to the squaring of differences. This makes RMSE particularly useful for identifying significant discrepancies in predictions, which are often critical in time-series forecasting tasks.

The use of RMSE is also supported by literature reviewed in Chapter 2, where it was highlighted as a standard evaluation metric for time-series models. Its widespread adoption in academic and industry applications underscores its reliability and relevance. A lower RMSE score indicates that the model has effectively captured the underlying patterns in the data and is capable of making accurate predictions. In this study, the RMSE calculation reflects the model's ability to generalize to unseen data, providing a quantitative measure of its success in forecasting stock prices. This step is crucial for validating the overall effectiveness of the LSTM model and aligning the results with the objectives of this project.

Researcher Observations:

High Degree of Flexibility and Customization:

Python offers unparalleled flexibility in constructing machine learning models, as demonstrated by the ability to fine-tune the LSTM architecture, optimize hyperparameters, and preprocess data according to specific requirements. This flexibility allows for precise control over every aspect of the modeling process, enabling the creation of highly tailored solutions. However, this customization comes with a steep learning curve and demands a deep understanding of machine learning concepts and coding proficiency, making it less accessible for non-technical users.

Time-Intensive Workflow:

While Python provides powerful tools for building predictive models, the process is often time-consuming. For example, the model-building steps—including data preparation, preprocessing, training, and evaluation—took approximately 3 minutes and 7 seconds for a small dataset. Scaling this workflow for larger datasets or more complex architectures could significantly increase time requirements. Additionally, the need for iterative experimentation, such as hyperparameter tuning and handling of potential errors, further extended development time.

Limited Integration and Visualization Capabilities at No Cost:

The Python-based approach lacks built-in integration with visualization and dashboard tools, requiring external libraries (e.g., Matplotlib, Seaborn) or separate platforms to present results effectively. This disjointed workflow may hinder business analysts or decision-makers who need end-to-end solutions.

3.5 SAC Solution

3.5.1 Data Preparation and Cleaning

Data preparation in SAP Analytics Cloud (SAC) is straightforward and user-friendly, leveraging its graphical interface for data import, validation, and preparation. Below are the steps undertaken to prepare the stock price dataset for time-series analysis, ensuring the data is clean, complete, and ready for further processing.

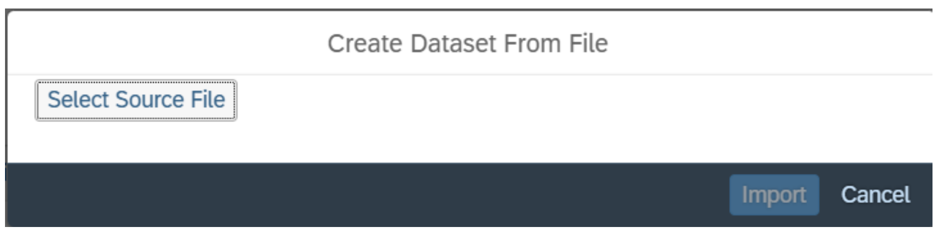


Fig 3. 12 Importing Dataset into SAC

The first step involves importing the dataset into SAC, as shown in **Fig 3.12**. The "Select Source File" option is used to choose the HistoricalQuotes.csv file.

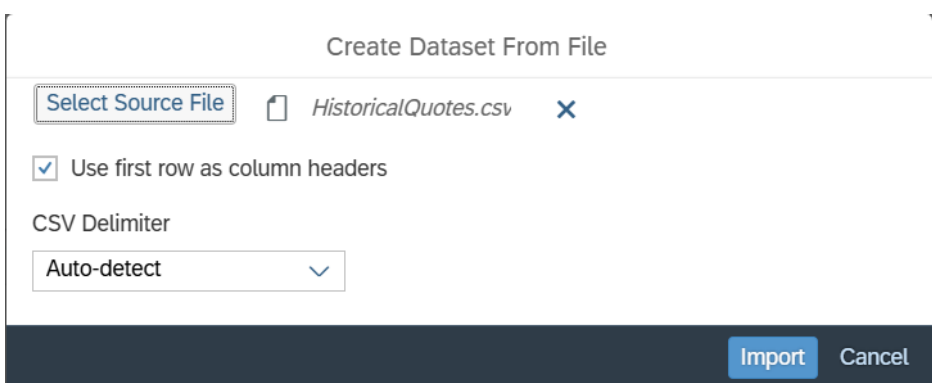


Fig 3. 13 Configuring Dataset Import Settings

Once the file is selected, SAC provides options to configure the file settings, as depicted in **Fig 3.13**. Here, the first row is specified as the column header, and the delimiter is auto-detected.

Specifying column headers and ensuring correct delimiter detection ensures that the dataset structure is accurately interpreted by SAC. This prevents misalignment or mislabeling of data columns during import.

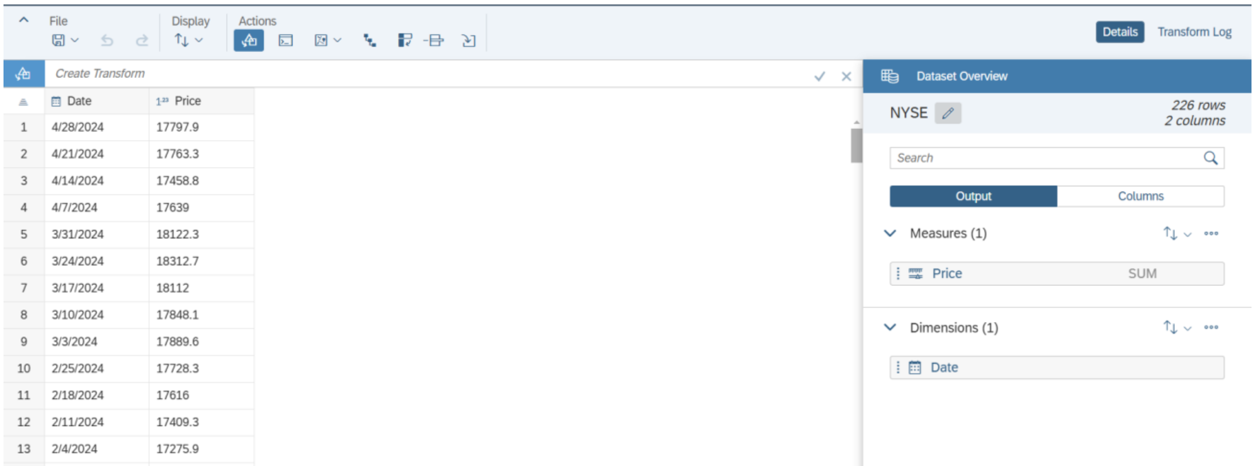


Fig 3. 14 Successfully Imported and Validated Dataset in SAC

After importing, the dataset is displayed in SAC's interface, as shown in **Fig 3.14**. The data overview indicates that the dataset contains 226 rows and 2 columns (Date and Price), with no missing values or errors detected. Null values, or missing data, are then methodically eliminated to ensure data consistency and integrity. The fact that many machine learning algorithms are not designed to handle missing data is what drives the removal of null values. Since algorithms frequently rely on mathematical operations, the existence of null values may cause those operations to go wrong.

3.5.2 Data Understanding

In SAP Analytics Cloud (SAC), data understanding is an intuitive and user-friendly process. SAC automatically provides an overview of the dataset, including key metrics and error checks, directly within its interface. When you scroll down on the right-hand side of **Fig 3.8**, SAC displays a summary of the dataset, such as the total number of rows, columns, and an automatic identification of any anomalies, such as missing values or inconsistencies. Both the histogram and the Box-Plot will automatically appear without any hint to system logic. Their results have been discussed in chapter – 4.

3.5.3 Exploratory Data Analysis

In SAP Analytics Cloud (SAC), creating a line chart for EDA involves an intuitive, step-by-step process using its drag-and-drop interface. This approach is especially user-friendly, requiring no coding knowledge. Below is the process for creating a line chart to analyze stock price trends over time through the UI in **Fig 3.15**.

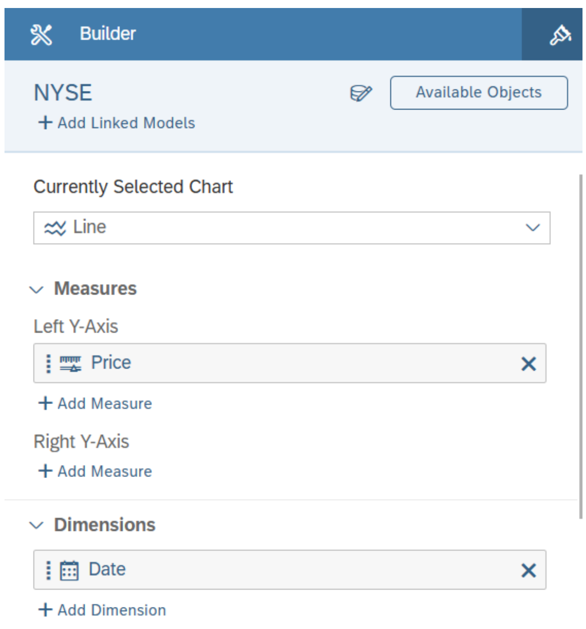


Fig 3. 15 Configuring a Line Chart in SAC

Here are the steps in building a line chart in SAP Analytics Cloud:

Step 1: Selecting the Dataset	At the top of the interface, the dataset NYSE is selected as the source for the analysis, as shown in **Fig 3.12**.
Step 2: Choosing the Chart Type	The "Currently Selected Chart" dropdown is used to select the **Line** chart, which is ideal for time-series analysis, as shown in **Fig 3.12**.
Step 3: Adding the Measure	Under the "Measures" section, the **Price** column is added to the left Y-axis.
Step 4: Adding the Dimension	Under the "Dimensions" section, the **Date** column is added to the X-axis.

3.5.4 ML model Building

Building a time-series forecasting model in SAP Analytics Cloud (SAC) involves selecting parameters in an intuitive interface, significantly simplifying the process for users. The configuration options available, as shown in **Fig 3.16**, allow users to define key aspects of the model and customize it for their specific forecasting needs.

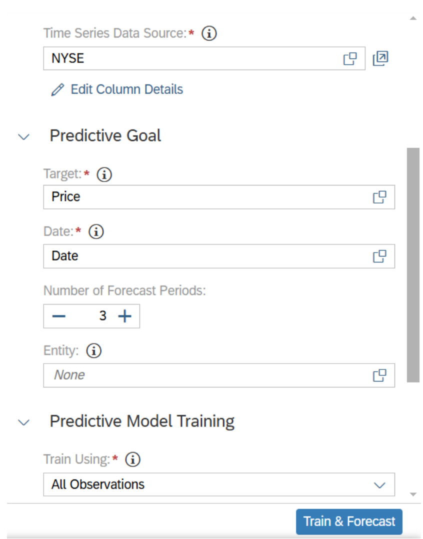


Fig 3. 16 Building ML model in SAC

Step 1: Selecting the Dataset	At the top of the interface, the dataset NYSE is selected as the time-series data source, ensuring that the correct data is used for predictive modeling, as shown in Figure 3.13.
Step 2: Defining the Predictive Goal	The target variable (Price) is selected for forecasting, and the date column (Date) is specified as the time dimension. The number of forecast periods (e.g., 3) is also set to define the forecasting horizon.
Step 3: Configuring Model Training	The training data scope is configured under "Train Using," with options like All Observations or custom-defined subsets of the dataset, allowing flexibility to focus on relevant time periods for training.
Step 4: Training and Forecasting	The model is trained by clicking the Train & Forecast button. SAC automatically builds and evaluates the predictive model using its in-built machine learning capabilities, providing results with minimal technical input.

Table 3. 5 SAC Model Building Steps

Back-end Algorithm process:

When building a time-series forecasting model in SAP Analytics Cloud (SAC), the system employs a sophisticated behind-the-scenes workflow. The process begins with decomposing the time series into its fundamental components—trend, cycles, fluctuations, and residuals. SAC primarily uses three core algorithms: **Additive Models**, **Piece-Wise Trend Models**, and **Exponential Smoothing Models**. Each algorithm is designed to address different characteristics of the dataset, and the system automatically selects the best-performing model based on predictive accuracy.

The **Additive Model** decomposes the time series into distinct components: trend, cycles (seasonal or periodic patterns), optional influencers (external factors), fluctuations, and residuals. This model assumes that the time series is a linear combination of these components, making it effective for datasets with clear seasonal trends and consistent variations. It ensures that unexplained variances (residuals) are minimized, although residuals are considered unnecessary for decision-making.

The **Piece-Wise Trend Model** focuses on identifying and optimizing change points in trends and cycles. By constructing a piece-wise additive framework, this model accounts for shifts in linear trends, periodic patterns, and fluctuations. It is particularly effective for datasets that exhibit non-linear or abrupt changes in trend direction, ensuring the model remains adaptable to dynamic changes over time.

The **Exponential Smoothing Model** uses weighted averages of past values to predict future outcomes, with three variations—Simple, Double, and Triple. The **Simple model** predicts based on a single level of past values, the **Double model** incorporates trends (level and slope), and the **Triple model** adds seasonal cycles into its calculation. These models are ideal for smoothing out noise in time-series data and capturing long-term trends and seasonality, ensuring robustness in predictions.

Once the algorithms have processed the data, SAC evaluates their performance by calculating accuracy metrics. It then selects the model that delivers the best balance of complexity and predictive performance. For instance, the system prefers simpler models if they provide comparable accuracy to more complex ones, ensuring interpretability and efficiency. This automated selection process eliminates the need for manual intervention, making SAC a highly user-friendly platform for time-series forecasting.

3.5.5 Visualization Integration

Fig 3. 17 Visualization Integration in SAC

The final step in the use-case scenario involves generating a forecast directly from a visualization in SAP Analytics Cloud (SAC), as shown in **Fig 3.17**. This feature allows users to derive predictive insights seamlessly from existing charts without requiring any coding knowledge. By interacting with the chart's context menu, accessible via the three-dot icon in the top-right corner, users can select **More Options** and then choose **Forecast** from the dropdown menu. SAC's built-in algorithms automatically process the underlying data, extrapolating future values based on historical trends, seasonality, and fluctuations. The forecast is then visually overlaid onto the chart, enabling an immediate comparison between actual and predicted values.

This no-code functionality is highly user-friendly and accessible to business users, making it a powerful tool for real-time decision-making. Additionally, the forecasting feature is available at no additional cost, unlike many traditional predictive tools that often require specialized licenses or expertise. This step highlights SAC's ability to integrate predictive analytics directly into the visualization process, streamlining workflows and empowering users to leverage actionable insights efficiently and effectively.

3.5.6 Comparison and conclusion

The methodologies and frameworks outlined for this use-case sets the foundation for a detailed comparative results analysis in Chapter 4. The results obtained from both traditional methods (e.g., Python and LSTM) and SAP Analytics Cloud (SAC) across the two use-case scenarios will be thoroughly examined to identify their relative strengths and weaknesses.

This comparative analysis will focus on key metrics such as predictive accuracy, usability, integration capabilities, and resource efficiency. The insights gained from this analysis will inform the conclusions presented in Chapter 4, where the findings will be synthesized to evaluate the overall effectiveness of SAC as a unified analytics solution.

3.6 Use-Case scenario 2

3.6.1 Define Objectives

This use-case scenario aims to assess the capabilities of **SAP Analytics Cloud (SAC)** as a visualization tool for creating comprehensive, interactive dashboards. The focus is on determining whether SAC can meet the requirements of modern business intelligence platforms while providing a no-code, user-friendly solution for business users. By analyzing SAC's performance in building dashboards that are dynamic, customizable, and visually impactful, this use case will establish its viability as a competitor to industry-leading tools like Power BI and Tableau.

The dashboard created for this scenario visualizes key financial and operational metrics such as revenue, gross margins, and product performance across multiple dimensions like time, location, and product categories. This setup allows stakeholders to analyze data interactively, extract actionable insights, and make informed decisions. The results from this scenario will help determine if SAC is a robust alternative for organizations looking to adopt advanced analytics without incurring high technical or financial costs.

For the visualization use case, the key metrics to evaluate SAC's effectiveness include:

1. To what extent can SAP Analytics Cloud compare with current solutions when it comes to visualization options?
2. To what extent does SAC's visualization interface enable users to customize and adapt dashboards to specific business needs?
3. How does SAC compare to traditional visualization tools in terms of ease of use?

3.6.2 Data Prep

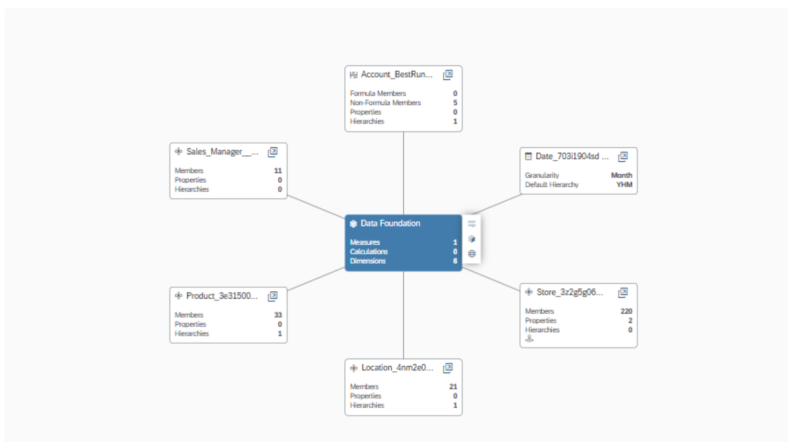


Fig 3. 18 SAC Star Schema Model

A **star schema** is a database structure used to organize data for analytics and reporting. It consists of a **central fact table** that contains the measurable data (e.g., sales, revenue, gross margin) and multiple **dimension tables** that provide descriptive attributes related to the facts (e.g., product, location, date). This structure is visually represented like a star, with the fact table at the center and dimension tables branching out around it.

In the context of SAP Analytics Cloud (SAC), the star schema is instrumental in organizing data efficiently for visualization and analysis. SAC's use of star schemas allows for **optimized querying** and **faster performance** when generating insights, as the schema simplifies complex relationships between data elements. Each dimension table connects to the fact table via unique identifiers, enabling users to drill down into granular details or aggregate data for higher-level insights.

For example:

- The **Fact Table** might contain total sales or revenue.
- **Dimension Tables** like "Product" can allow users to analyze revenue by individual product categories (e.g., Alcohol, Carbonated Drinks). Similarly, the "Location" dimension table can help drill down by geography (e.g., state or region).

This structured approach ensures that SAC is capable of handling large datasets efficiently, similar to how platforms like Power BI or Tableau operate. However, SAC's **no-code approach** to setting up star schemas gives it a competitive advantage for users with limited technical expertise. The intuitive drag-and-drop interface makes it easier to create connections between tables, define hierarchies, and configure relationships without requiring complex scripting or advanced database knowledge.

3.6.3 Visualization Development

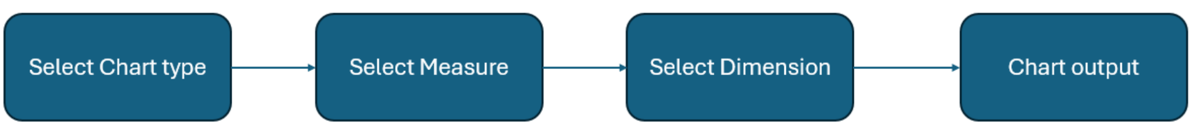


Fig 3. 19 SAC standardized Visualization Building Process

All the charts presented in use-case scenario 2 were constructed following a standardized process, shown in **Fig 3.19**. This includes four sequential steps: selecting the chart type, determining the measure, specifying the dimension, and finally generating the chart output. This systematic approach ensures consistency and accuracy in data visualization.

1. Gross Margin Percentage and Net Revenue Cards

The KPI cards are used to display critical metrics like Gross Margin Percentage and Net Revenue because they provide a high-level snapshot of performance that is immediately interpretable. KPI cards are ideal for summarizing key data points in a visually appealing format, allowing stakeholders to quickly assess financial health without diving into granular details. Their simplicity ensures that essential information is not overshadowed by complex visualizations.

2. Net Revenue by Product Pie Chart

A pie chart is used to represent the proportion of revenue contributed by different product categories. This visualization is particularly effective for showing percentage breakdowns, as it allows users to immediately grasp how each product contributes to overall revenue. Pie charts work best when comparing parts to a whole, making them the optimal choice for highlighting the revenue distribution across product lines.

3. Time-Series Chart: Net Revenue per Product Line

The time-series chart is employed to track net revenue for each product line over time, making it an ideal tool for identifying trends, patterns, and seasonality. Time-series charts are best suited for data that changes over time, as they allow stakeholders to visualize historical performance and forecast future trends. This makes it easier to detect growth opportunities or periods of decline that may require strategic intervention.

4. Gross Margin Percentage by Product (Location-Wise Bar Chart)

The bar chart is used to compare gross margins for different product lines across various locations. Bar charts are effective for categorical comparisons, as they clearly highlight differences in values across regions and products. This type of visualization is ideal for analyzing regional performance, enabling stakeholders to identify high-margin areas or underperforming regions that may need targeted strategies.

5. Year-Over-Year (YOY) Gross Margin Percentage by Product Line

This bar chart is designed to showcase changes in gross margins over multiple years for each product line. Bar charts are well-suited for year-over-year comparisons because they emphasize differences across time periods, making it easy to spot trends or anomalies. This visualization is particularly useful for assessing long-term performance and measuring the impact of strategic decisions on profitability.

6. Filters: Date, Product, and Location

Filters enhance the interactivity of the dashboard by allowing users to customize the data they view. The Date filter enables a focus on specific time periods, while the Product and Location filters allow users to isolate performance metrics for particular categories or regions. These dynamic controls are essential for dashboards that need to cater to diverse stakeholder requirements, as they provide the flexibility to drill down into the data most relevant to the user's objectives.

3.6.4 Conclusion

To evaluate whether SAP Analytics Cloud (SAC) meets the objectives for Use-Case Scenario 2, the focus will be on determining its capability as a visualization and modeling solution. The evaluation will be centered on four key aspects that reflect the objectives and requirements of this use case:

1. **Knowledge Required to Build the Dashboard**:
   The complexity of knowledge and technical expertise required to construct the dashboard will be assessed. This involves analyzing whether SAC's no-code, drag-and-drop interface simplifies the dashboard-building process, allowing users with minimal technical experience to create comprehensive and interactive visualizations.

2. **User-Friendliness**:
   The usability of SAC will be evaluated by examining its interface design, ease of navigation, and intuitiveness. This includes assessing whether the platform supports a seamless workflow for building dashboards and provides guidance or automation features that enhance the user experience.

3. **Visualization Options**:
   SAC's range of visualization types will be reviewed to determine if it offers sufficient flexibility to represent complex datasets effectively. This includes evaluating whether SAC supports advanced visualizations like time-series charts, KPI cards, and even unique options like boxplots, which can provide deeper insights and are not readily available in some other platforms without coding.

The success of this use case will be determined by analyzing SAC's performance across these four aspects. If SAC proves to be user-friendly, requires minimal technical knowledge, supports robust drill-down capabilities, and offers diverse visualization options, it can be concluded that SAC is capable of serving as a comprehensive visualization solution. This focused evaluation ensures that the assessment remains practical and directly tied to the goals of the use case. The evaluation results will be reflected in the findings in chapter-4.

Ch-4 Results

4.1 Overview

This chapter presents the results of implementing SAP Analytics Cloud (SAC) as a no-code business analytics solution, explored through two use-case scenarios: time-series forecasting (stock price prediction) and dashboard-based performance analysis. The outcomes highlight SAC's efficiency, ease of use, and seamless integration compared to traditional coding-intensive approaches.

The analysis evaluates SAC's predictive accuracy and visualization capabilities, emphasizing its role as an accessible, all-in-one platform for business analysts. Key results include SAC's ability to deliver comparable performance metrics to traditional solutions while significantly reducing technical complexity and development time. Additionally, the chapter discusses the interactive web-based application showcasing SAC's functionality, reinforcing its suitability as a modern analytics tool.

The results from both the traditional solution and the SAP Analytics Cloud (SAC) solution will be compared step by step throughout Chapter 4. This detailed comparison will focus on SAC's accuracy and performance relative to established current solutions, highlighting its ability to deliver comparable or superior results. Each stage of the model-building and visualization development process—such as data preparation, model training, dashboard customization, and interactivity—will be analyzed to demonstrate how SAC balances predictive accuracy with ease of use, development time, and flexibility. This approach ensures a comprehensive evaluation of SAC's effectiveness as a modern analytics platform.

4.2 Use-Case Scenario 1 – TIME SERIES

This section evaluates the application of SAP Analytics Cloud (SAC) for time-series forecasting, benchmarking its performance against Python's Long Short-Term Memory (LSTM) models. The objective is to determine SAC's capability to deliver competitive predictive accuracy while offering a user-friendly, no-code interface for non-technical users. By analyzing data preparation, exploratory data analysis (EDA), model development, and evaluation metrics, this use case provides insights into SAC's potential to simplify predictive modeling workflows. The results will demonstrate how SAC democratizes access to advanced analytics by minimizing technical prerequisites and enabling seamless integration of predictive insights into decision-making processes.

4.2.1 Data Preparation and cleaning

a. SAC

Delete Duplicate rows

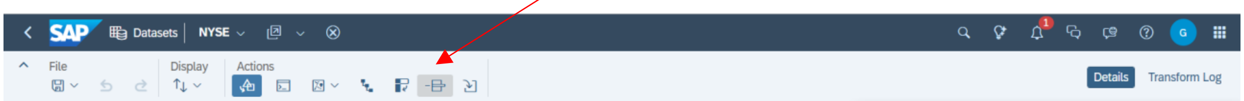


Fig 4. 1 SAC User-friendly UI "Clear Duplicates" button

Data Preparation is the primary step to verify and clean the dataset, ensuring it is ready for analysis. As shown in **Figure 4.1**, SAC simplifies this process with a single-click functionality that removes duplicate rows. Removing duplicates is critical because they can lead to biased results and inaccuracies in predictions. After using the "Delete Duplicate Rows" feature (refer to **Figure 4.1**), it was confirmed that there were no duplicates in the dataset, ensuring the data was already clean and prepared for the next stages of analysis.

b. Python

```
0
Date     0
Price    0
dtype: int64
```

Fig 4. 2 Python Data cleaning output

The output shown in **Fig 4.2** confirms that there are no missing values in the dataset, as shown by the count of zero null entries for both the **Date** and **Price** columns. Similarly, in SAC, this result is achieved effortlessly with zero coding required. SAC automatically identifies and flags missing values during the data import process, providing a streamlined and user-friendly data validation workflow.

Comparison

Both Python and SAP Analytics Cloud (SAC) successfully identified that there were no missing values in the dataset. However, while Python required explicit coding to check for null values, SAC handled this automatically during the data import process. This highlights SAC's efficiency in automating data validation tasks, making it more user-friendly and accessible for non-technical users.

4.2.2 Data Understanding

Data understanding is the foundational step in building a robust and accurate time-series model. It involves exploring and analyzing the dataset to identify patterns, trends, and anomalies that inform subsequent data preprocessing and modeling. We will be visualizing the data through histogram and a boxplot. Here below are the output and their relevant explanations.

a. SAC

1. Histogram Analysis

Histograms are a powerful tool for visualizing the frequency distribution of stock prices over the defined timeline. SAC offers an interactive histogram feature, allowing dynamic bin adjustments from 5 to 40 bins via a simple drag-and-drop mechanism.

Figures **4.3** to **4.6** represent histograms with different bin counts (5, 10, 20, and 40 bins), helping refine our understanding of the dataset's distribution:

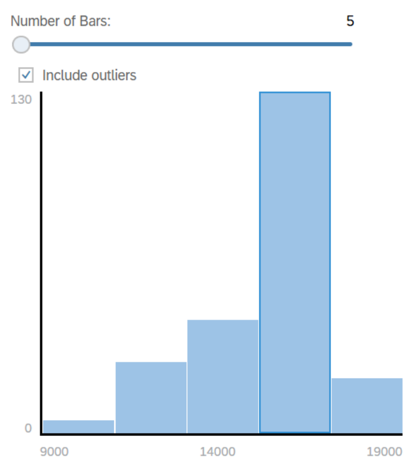


Fig 4. 3 Number of Bars: 5

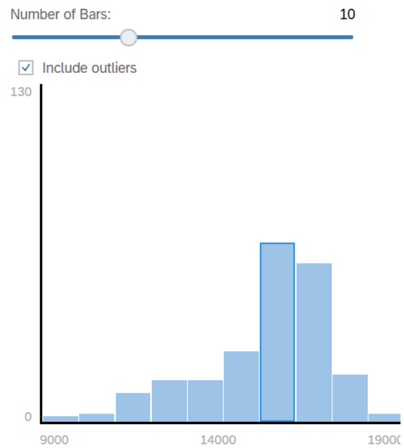


Fig 4. 4 Number of Bars: 10

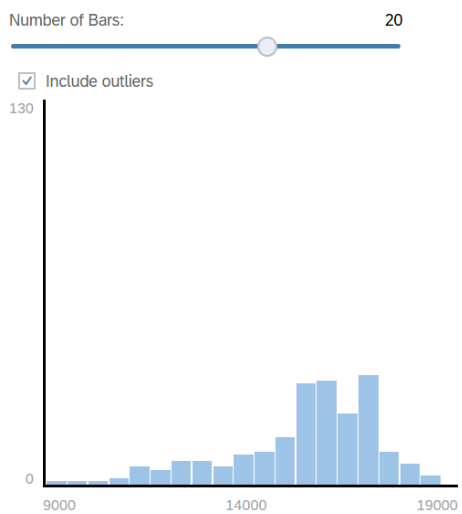


Fig 4. 5 Number of Bars: 20

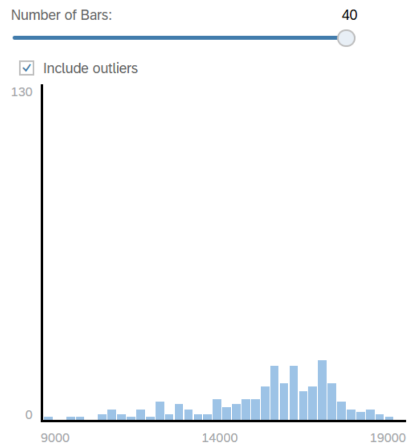


Fig 4. 6 Number of Bars: 40

Table 4. 1 Number of Bars

Figure No.	Number of Bins	Analysis
Figure 4.3	5	The broader binning provides an overview of the data distribution, showing general concentration near the midrange.
Figure 4.4	10	Increased granularity reveals distinct clusters in the data, indicating skewness and potential outliers.
Figure 4.5	20	Further granularity highlights minor peaks and dips, aiding in identifying subtle variations in the dataset.
Figure 4.6	40	The highest granularity exposes finer patterns and confirms the absence of significant anomalies in the dataset.

From these histograms, we learned that the dataset exhibits a steady distribution without significant irregularities or gaps, ensuring suitability for time-series modeling. The data's central concentration aligns well with predictive forecasting requirements (L. 2020).

Hovering over a specific bin in SAC reveals **summary statistics, including count, range, minimum, and maximum values, as shown in Fig 4.7 below:**

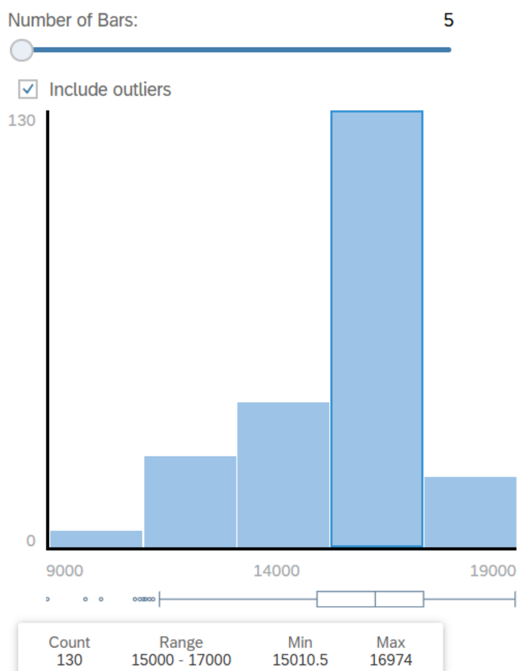


Fig 4. 7 SAC Description Statistics

The descriptive statistics shown in **Fig 4.7** provide a numerical summary of the dataset's distribution:

Table 4. 2 Description Statistics overview

Count:	130	Represents the total number of data points included in this analysis.

Range:	15,000 - 17,000	Indicates the spread of the data, showing the interval of values observed.
Minimum:	15,010.5	The smallest value in the dataset, serving as the lower boundary of data.
Maximum:	16,974	The largest value in the dataset, representing the upper boundary of data.

2. Box Plot

Fig 4. 8 SAC Box Plot

The key elements of the box plot shown in **Fig 4.8** are as follows: **Outlier Detection and Data Variability Using Box Plots**

The box plot in SAC complements the histogram by providing a statistical summary of the dataset. Key elements include:

Table 4. 3 SAC Statistical Summary

Statistic	Explanation	Values
Median	Represents the central tendency, indicating the typical value around which prices fluctuate.	Approximately 14,000
Interquartile Range (IQR)	Encompasses the middle 50% of values, defining the range of standard price fluctuations.	From 13,500 to 15,500
Whiskers	Extend to capture most of the dataset values while excluding extreme outliers.	Approximately 9,000 to 19,000

Outliers	Identified as extreme values below the lower whisker, often linked to market anomalies or sharp dips.	Below 9,000

b. Python

1. Histogram Analysis

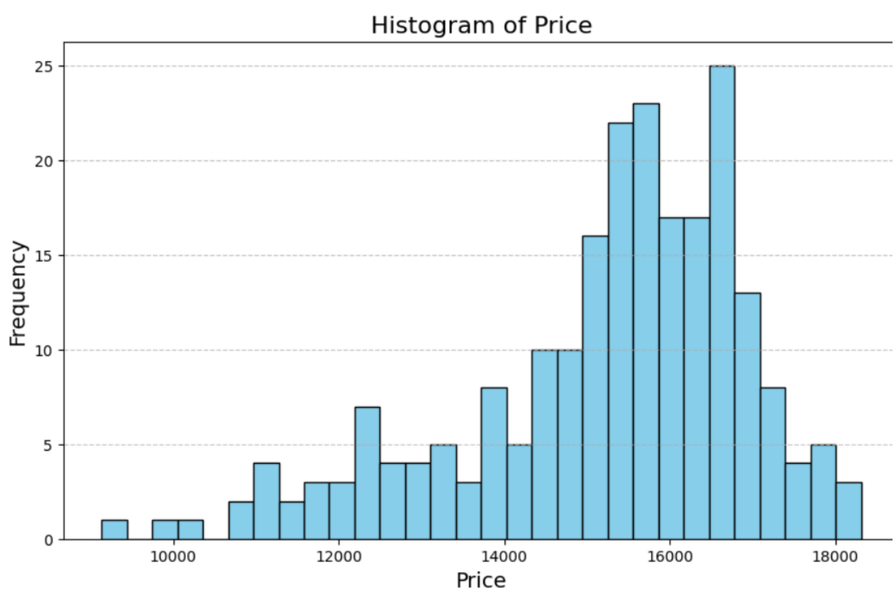


Fig 4. 9 Python Histogram output

Fig 4.9 depicts the histogram of the **Price** variable, visualizing the distribution of stock prices within the dataset. The x-axis represents the price range, while the y-axis indicates the frequency of occurrences within each bin. This visualization reveals that the majority of stock prices are concentrated between **13,000 and 16,500**, with a noticeable peak around **16,000**, suggesting that prices are not uniformly distributed and tend to cluster around certain values. The histogram highlights the skewness of the data, as well as any anomalies or outliers outside the main range.

2. Boxplot Analysis

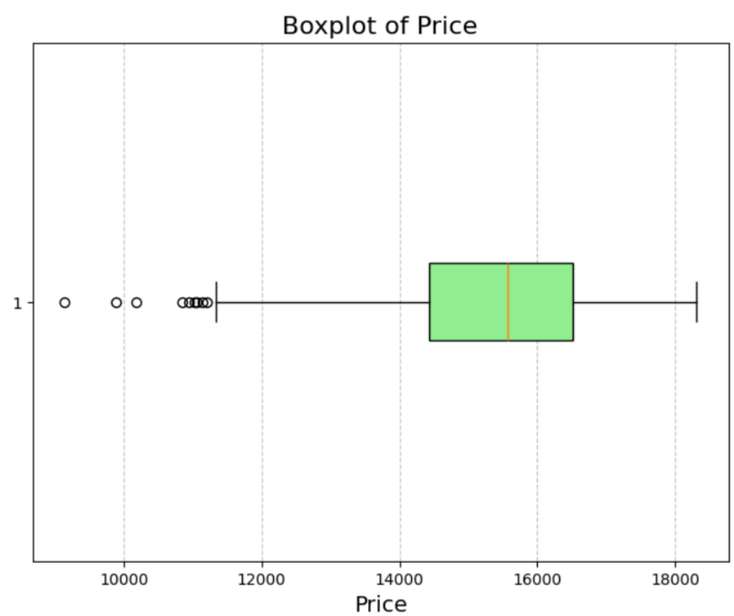


Fig 4. 10 Python Boxplot output

The boxplot in **Fig 4.10** provides a concise summary of the **Price** variable, illustrating the central tendency, variability, and distribution of the data. The **box** represents the interquartile range (IQR), capturing the middle 50% of values, while the line inside the box indicates the **median**.

The **whiskers** extend to capture most of the data points within 1.5 times the IQR, while **outliers** are represented as individual points outside the whisker range. This visualization is particularly useful for identifying anomalies, such as prices below **10,000**, and understanding the spread of stock prices. It highlights the skewness and variability of the dataset effectively.

Comparison

The data understanding phase revealed that both Python and SAP Analytics Cloud (SAC) effectively captured essential insights, such as identifying outliers, understanding data distributions, and summarizing central tendencies. However, SAC demonstrated significant advantages in terms of usability and accessibility. While Python required explicit coding to generate visualizations like histograms and boxplots, SAC achieved comparable outputs with a no-code, drag-and-drop interface, allowing users to explore data more intuitively. SAC's native support for advanced visualizations like boxplots, which typically require additional scripts in other platforms like Power BI, further highlights its flexibility. Moreover, SAC's ability to seamlessly integrate these visualizations into interactive dashboards enhances real-time exploration and decision-making. Overall, SAC streamlined the data understanding process, providing a user-friendly and efficient alternative without compromising analytical depth.

4.2.3 Exploratory Data Analysis

a. SAC

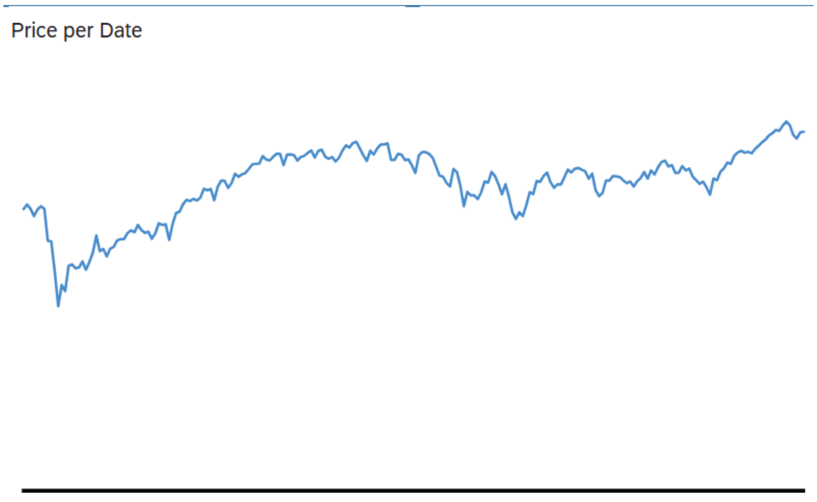


Fig 4. 11 SAC Line Chart

Figure 4.11 illustrates the movement of stock prices over time, with the x-axis representing dates and the y-axis showing the corresponding stock prices. This chart provides a clear visualization of how prices have evolved throughout the observed period through a **Line Chart**.

The graph shows an initial sharp decline in stock prices, followed by a recovery and an overall upward trend. While there are fluctuations along the way, the general direction suggests growth over time. These periodic rises and falls indicate short-term market volatility, likely influenced by various external factors such as market demand or economic changes.

The initial drop and subsequent recovery highlight significant points in the dataset that could correlate with major market events. These insights make the dataset well-suited for time-series forecasting, as the chart displays meaningful patterns that can help predict future trends.

In summary, this line chart provides valuable information about the dataset's historical behavior, confirming its suitability for building predictive models. The overall upward movement and distinct fluctuations offer a strong foundation for understanding past trends and projecting future stock prices.

b. Python

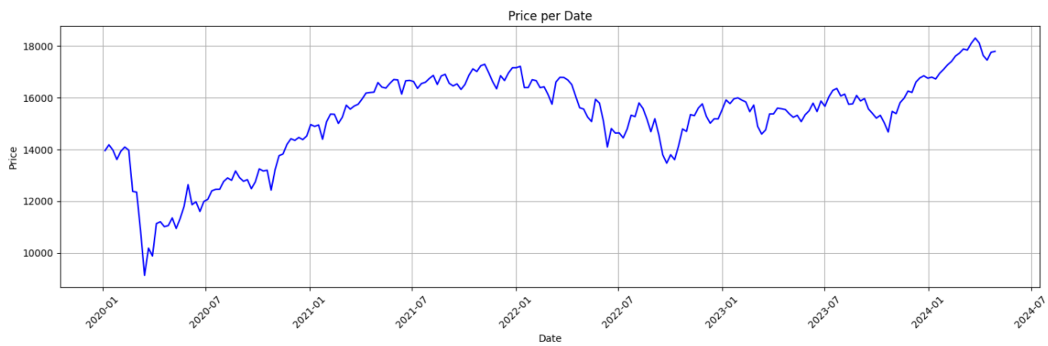


Fig 4. 12 Python Line Chart

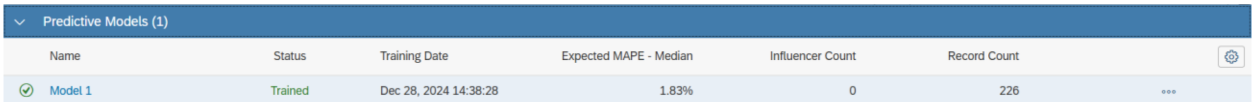
Fig 4.12 represents a line chart plotting the stock price over time, capturing the overall trend, fluctuations, and growth in prices across the dataset. The chart clearly shows patterns of increases and decreases, highlighting periods of stability and volatility. This visualization is particularly effective in understanding time-series trends, making it a crucial step in data understanding for forecasting purposes.

Comparison

Both Python and SAP Analytics Cloud (SAC) effectively visualized the time-series data, showcasing the price trends over time. However, while Python required specific coding to configure the chart's layout, axes, and labels, SAC achieved the same output through its drag-and-drop functionality and built-in visualization tools. SAC's no-code approach makes it significantly more accessible, allowing users to interactively build such visualizations without technical expertise. Additionally, SAC offers seamless interactivity and integration with dashboards, which further enhances its usability compared to Python's standalone output.

4.2.6 Model Development

a. SAC



Predictive Models (1)

Name	Status	Training Date	Expected MAPE - Median	Influencer Count	Record Count
Model 1	Trained	Dec 28, 2024 14:38:28	1.83%	0	226

Fig 4. 13 SAC Model Training Result

When the model is trained in SAP Analytics Cloud (SAC), the system provides key outputs that summarize the model's training and performance, shown in **Fig 4.13**. These outputs include the **Model Name**, which uniquely identifies the trained model; the **Model Training Status**, confirming whether the training process was successfully completed; the **Training Date**, which records the exact time the training was finalized; the **Expected MAPE (Median)**, a critical metric that measures the model's accuracy as the Mean Absolute Percentage Error; the **Influencer Count**, showing the number of additional variables influencing the target variable; and the **Record Count**, indicating the total number of data points used in training. The time took for the model to trained has been timed to **11 seconds**.

Table 4. 4 SAC Model Training Outputs

Output	**Value**
Model Name	Model 1
Model Training Status	Trained
Training Date	December 28, 2024, 14:38:28
Expected MAPE (Median)	1.83%

Influencer Count	0
Record Count	226

As shown in **Figure 4.13**, the model named "Model 1" has been successfully trained, with an **Expected MAPE of 1.83%**, which signifies high predictive accuracy. The **Influencer Count of 0** suggests that no external variables were identified as significant influencers, implying that the dataset itself is sufficient for accurate time-series forecasting. The **Record Count of 226** confirms the dataset's adequacy for training a robust and reliable model. These outputs highlight the model's efficiency and suitability for deployment in forecasting tasks.

b. Python

```
[array([0.52560597, 0.5501389 , 0.52783921, 0.48814206, 0.52276268,
       0.54078109, 0.52754507, 0.35381012, 0.35065091, 0.18724331,
       0.        , 0.11482107, 0.08142056, 0.21824718, 0.22605806,
       0.20531619, 0.20974999, 0.24196307, 0.19762514, 0.23953374,
       0.29084373, 0.38217768, 0.29783757, 0.31013672, 0.26920856,
       0.31137862, 0.32054033, 0.35617408, 0.36261234, 0.36296095,
       0.39572961, 0.41062149, 0.40044665, 0.43987145, 0.41221199,
       0.39651397, 0.40311564, 0.36518329, 0.3939866 , 0.44876083,
       0.43968626, 0.44301977, 0.35907184, 0.44506781, 0.50417779,
       0.51133504, 0.55180565, 0.57564137, 0.56888719, 0.58114276,
       0.57185032, 0.58735225, 0.63550302, 0.6275941 , 0.63386895,
       0.57345171, 0.64670189, 0.67938341, 0.67863173, 0.64026363])]
[0.6665504657116399]
```

Fig 4. 14 Python Model building output

The Python output shown in **Fig 4.14** provides a detailed statistical breakdown of the normalized values and predicted outputs during the model-building process. These values represent scaled features and the results from the model's training and testing stages, showcasing the intermediate steps in preparing the data for accurate predictions. While this output is valuable for debugging and verification, it is presented in a raw, unformatted format that may not be user-friendly for non-technical stakeholders.

Comparison

Both Python and SAP Analytics Cloud (SAC) effectively handle model-building tasks by processing and generating statistical outputs. However, SAC presents this statistical information in an interactive and visually appealing interface. With **zero coding**, SAC allows users to access key metrics, charts, and summaries that provide the same insights but in a much more digestible and user-friendly manner. SAC's well-designed UI not only simplifies the interpretation of statistical outputs but also integrates these insights seamlessly into dashboards, making it an ideal solution for users without technical expertise. This feature highlights SAC's advantage in transforming raw data into actionable visual insights without the need for manual scripting.

4.2.7 Model Evaluation

Fig 4. 15 Model Performance Comparison (Python Left side and SAC right side)

Fig 4.15 compares the performance of the Python-based model and the SAP Analytics Cloud (SAC) model in terms of accuracy and Root Mean Squared Error (RMSE). The Python model achieved an **accuracy of 94.63%** with an RMSE of **816.29**, while SAC slightly outperformed it with an **accuracy of 95.13%** and a lower RMSE of **740.03**. This improvement demonstrates SAC's ability to optimize predictive performance using its built-in algorithms.

Comparison

While both platforms delivered high levels of accuracy, SAC's performance stands out due to its slightly higher accuracy and lower error rate. Additionally, SAC achieved these results with **zero coding**, providing an automated and user-friendly workflow that simplifies the model-building and evaluation process. In contrast, Python required manual configuration, coding, and debugging to produce comparable results. SAC's ability to deliver increased accuracy with minimal effort further highlights its value as a powerful and accessible analytics tool for both technical and non-technical users.

4.2.8 Additional Supporting Metrics (SAC ONLY)

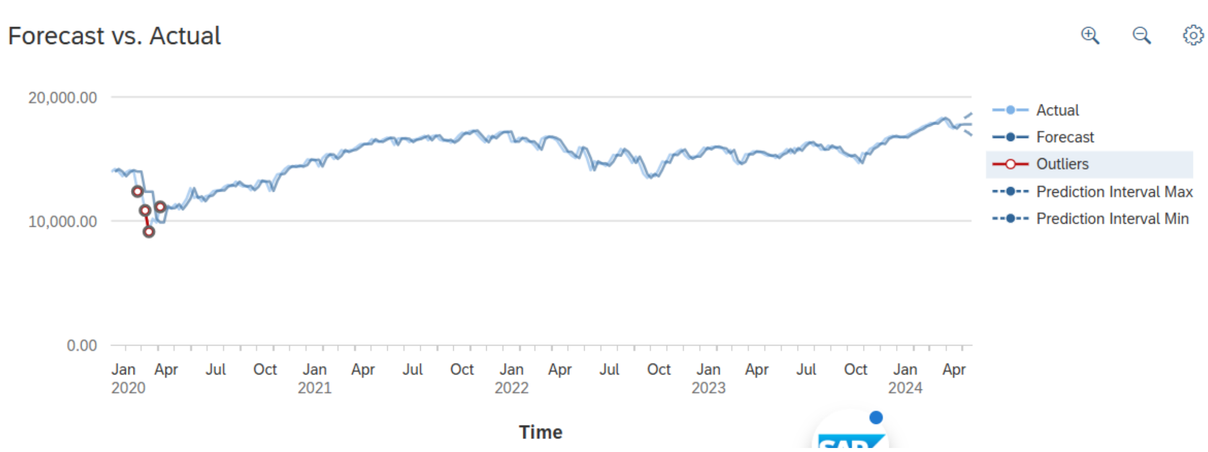


Fig 4. 16 SAC Forecast vs. Actual

Fig 4.16 presents the comparison between forecasted and actual stock prices over a timeline from January 2020 to April 2024. The blue line represents the actual observed values, while the dotted line shows the forecasted values. Outliers are marked with red circles, indicating significant deviations from the trend. The dashed lines outline the prediction interval bounds, providing a range within which the forecasted values are expected to fall.

The alignment between forecasted and actual values reflects the model's ability to capture key trends with accuracy. The identification of outliers ensures that anomalies are accounted for, which enhances the reliability of the analysis. Additionally, the forecasted values staying within the prediction interval bounds for most of the period indicates robust predictions. This visualization offers a clear understanding of the forecasting accuracy and helps assess the variability, ensuring informed decision-making based on predictable patterns.

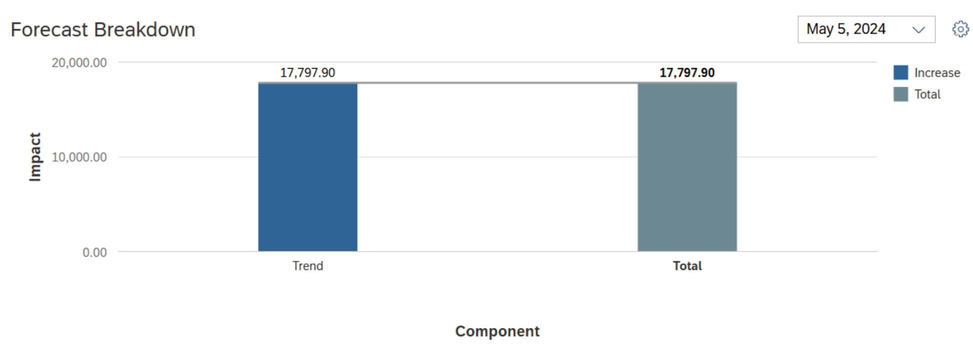


Fig 4. 17 SAC Forecast Breakdown

Fig 4.17 is a bar chart that breaks down the forecasted value for May 5, 2024, into its contributing components. The y-axis represents the numerical impact, while the x-axis categorizes the components, which include **Trend** and **Total**. Both bars show a value of **17,797.90**, indicating that the forecast is solely driven by the trend component, with no additional contributions from other factors such as seasonality or external variables.

This diagram simplifies the analysis by directly visualizing the contribution of individual components to the forecast. Unlike traditional tools that may require additional coding or manual integration of visuals, this approach provides a seamless and intuitive understanding of the forecast structure. The clarity and automation in breaking down the forecast make it easier to interpret and act upon, reducing the complexity and time typically associated with creating such analyses in coding-heavy tools or combining data across platforms.

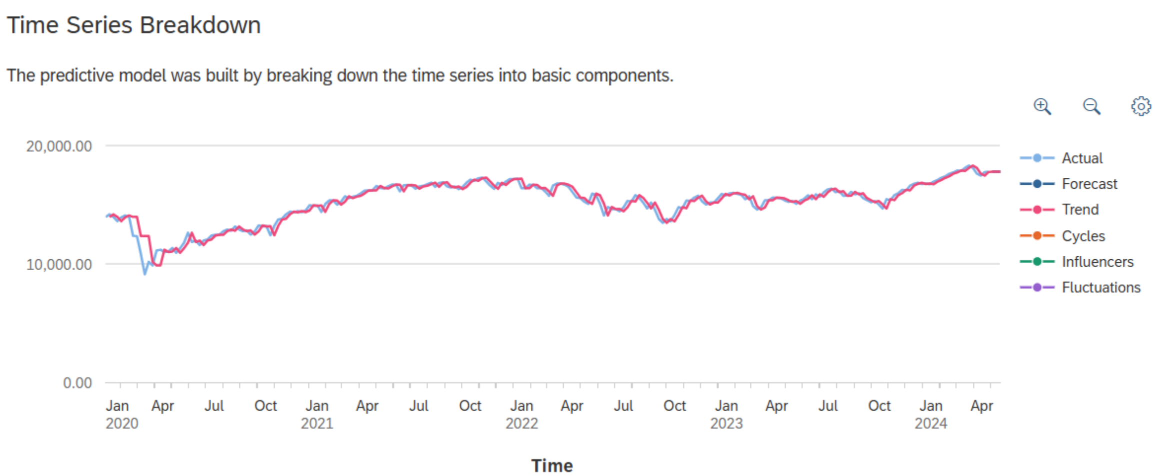


Fig 4. 18 SAC Time Series Breakdown

Fig 4.18 is a line chart that breaks the time series data into its core components: actual values, forecasted values, trend, cycles, influencers, and fluctuations. The x-axis represents the timeline from January 2020 to April 2024, while the y-axis displays the stock price values. The actual values are represented by the blue line, and the forecasted values are shown in red, closely aligning throughout the period, which indicates accurate prediction performance. The trend component, depicted in orange, captures the overall growth pattern of the data over time. Cycles, represented in green, reflect periodic patterns such as seasonality, while the influencers, shown in purple, highlight external factors affecting the dataset. Finally, the fluctuations, displayed in pink, represent irregular variations or residual errors after accounting for all other components.

This breakdown provides a clear understanding of how the forecast is constructed by isolating each contributing factor. The close alignment of the actual and forecasted values demonstrates the model's ability to predict trends effectively, while the identification of cycles and fluctuations adds depth to the analysis by distinguishing between predictable patterns and irregular changes. The minimal impact from influencers indicates that external factors have a limited role in shaping the dataset. This visualization simplifies the complex task of deconstructing time-series data into interpretable components, offering a transparent and actionable foundation for understanding and decision-making.

Time Series Component Impact

Component	Details	Impact
Trend		98.01%
Final Residuals		1.99%

Fig 4. 19 SAC Time Series Component Impact

Fig 4.19 presents a breakdown of the impact of different components on the time-series forecast. The table lists the two primary components: **Trend** and **Final Residuals**, along with their corresponding percentage impact on the forecast. The **Trend** component dominates with an impact of **98.01%**, while the **Final Residuals**, representing irregular or unexplained variations, contribute just **1.99%**.

This breakdown highlights the overwhelming influence of the trend component in shaping the forecast, indicating that the dataset exhibits a clear and consistent directional movement over time. The minimal impact of final residuals suggests that the model has effectively captured the significant patterns and accounted for most of the variability in the data. This level of transparency in the contribution of components ensures confidence in the forecast and offers actionable insights for understanding the driving factors behind the predictions.

Fig 4. 20 Target Statistics

Fig 4.20 displays the target statistics for the validation dataset, providing a summary of key numerical measures that characterize the dataset. The metrics include:

Table 4. 5 SAC Target Statistics

Minimum:	14,675.8
Maximum	18,312.7
Mean	16,299.19
Standard Deviation	954.41

These statistics offer insights into the distribution of the target variable. The minimum and maximum values indicate the range of the dataset, while the mean provides a measure of central tendency, highlighting that the average value of the target variable is around 16,299.19. The standard deviation of 954.41 reflects the spread of the data around the mean, indicating moderate variability in the target values.

This summary aids in understanding the characteristics of the validation dataset, ensuring that the model’s predictions are evaluated against a dataset with consistent and well-distributed properties. The metrics confirm the dataset's suitability for validation, supporting reliable performance assessment of the forecasting model.

Expected MAPE - Average	Expected MAE - Average	Expected MASE - Average	Expected RMSE - Average	Expected R^2 - Average
1.83%	296.86	0.95	353.73	0.85

Fig 4. 21 SAC Model Performance Metrics

Fig 4.21 provides a summary of the model's additional performance metrics, which are essential for evaluating the accuracy and reliability of the time-series forecasting model. The metrics in the **Fig 4.21** are summarized in the table including:

Table 4. 6 SAC Model Evaluation Metrics

Model Evaluation Metrics		
Metric	**Meaning**	**Value**
Expected MAPE (Average)	Mean Absolute Percentage Error, a measure of prediction accuracy in percentage terms	1.83%
Expected MAE (Average)	Mean Absolute Error, representing the average magnitude of the prediction errors.	296.86
Expected MASE (Average)	Mean Absolute Scaled Error, comparing the model's error to a naive benchmark model.	0.95
Expected RMSE (Average):	Root Mean Squared Error, indicating the standard deviation of prediction errors.	353.73

Expected R^2 (Average)	Coefficient of Determination, measuring the proportion of variance in the target variable explained by the model.	0.85

The **Expected MAPE** of 1.83% indicates a very low percentage error, reflecting high predictive accuracy. The **Expected MAE** of 296.86 measures the average absolute difference between the predicted and actual values, signifying minimal deviation. The **Expected MASE** of 0.95 sshows that the model performs slightly better than a naive baseline prediction. The **Expected RMSE** of 353.73 highlights the standard deviation of the prediction errors, showing low variability in errors. Finally, the **Expected R^2** of 0.85 confirms that 85% of the variance in the target variable is explained by the model.

These metrics collectively indicate that the forecasting model is highly accurate, with minimal errors and strong explanatory power, making it a reliable tool for time-series prediction.

When evaluating the model in SAP Analytics Cloud (SAC), various metrics are provided to assess its performance. These metrics offer insights into the model's accuracy, error magnitude and deep statistical information.

4.2.8 Additional Supporting Metrics (SAC ONLY)

4.2.8 Visualization Integration (SAC ONLY)

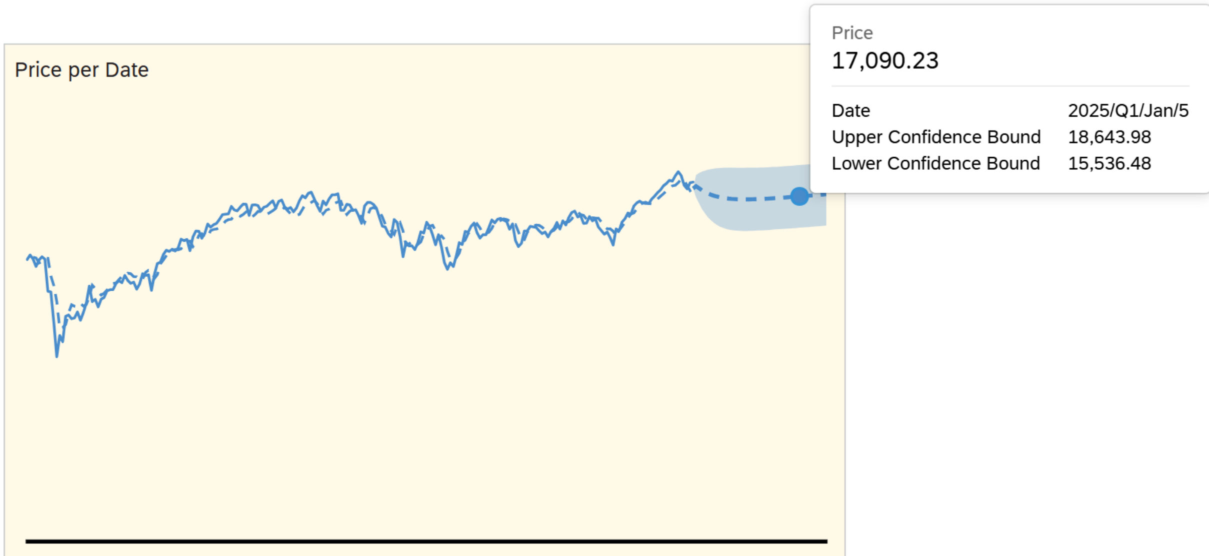


Fig 4. 22 SAC ML model and Visualization Integration

Fig 4.22 showcases SAP Analytics Cloud's (SAC) ability to integrate predictive analytics with interactive visualizations, specifically in the context of stock price prediction. The chart provides a forecast of future stock prices, along with confidence bounds that define the range within which the stock price is likely to fall. This integration is particularly beneficial for stock price prediction as it not only presents the predicted values but also conveys the uncertainty and reliability of those predictions.

The interactive tooltip further enhances this visualization by allowing users to hover over specific dates to see the predicted price, the associated date, and the upper and lower confidence bounds. For example, users can assess that on 2025/Q1/Jan/5, the predicted stock price is 17,090.23, with an upper bound of 18,643.98 and a lower bound of 15,536.48. This level of detail enables stakeholders, such as investors or analysts, to evaluate potential risks and opportunities based on forecasted price ranges rather than relying solely on a single predicted value.

This integrated visualization is particularly beneficial for decision-making in stock markets, where understanding variability and uncertainty is critical for risk assessment. By providing a clear visual representation of both predictions and confidence intervals, SAC allows users to make more informed decisions, such as determining entry and exit points for trades or understanding price volatility.

Moreover, this capability is offered by SAC at zero cost, unlike other platforms that may require additional software or coding expertise to produce similar outputs. This makes SAC an accessible and cost-effective tool for businesses and individuals seeking to leverage advanced predictive insights without incurring additional expenses.

4.2.9 Use-Case Scenario 1 Comparison and Conclusion:

The focus of the comparison will be on the following metrics to determine if SAP Analytics Cloud (SAC) is a viable solution for time-series forecasting compared to Python's Long Short-Term Memory (LSTM) model:

Table 4. 7 SAC vs Python-based model comparison

Metric	**Python (LSTM)**	**SAP Analytics Cloud (SAC)**	**Comparison**
Accuracy	94.63% accuracy, RMSE: 816.29	95.13% accuracy, RMSE: 740.03	SAC achieved slightly higher accuracy and lower RMSE, demonstrating better optimization.
Ease of Use	Requires complex coding and debugging	No-code, drag-and-drop interface	SAC is more accessible and user-friendly, especially for non-technical users.
Visualization Integration	Separate coding required for data visualization	Built-in visualization with confidence intervals	SAC integrates predictive outputs directly into visualizations, enhancing interactivity.

Time and Resource Efficiency	Model building took 3 minutes and 7 seconds	Model building completed in 11 seconds	SAC is faster and requires less effort to set up and execute.
Statistical Insights	Requires manual coding for decomposition	Automatic decomposition into trends, cycles, and residuals	SAC simplifies exploratory analysis with pre-built visual tools.
Customization	Highly flexible; supports advanced configurations	Limited to predefined workflows	Python offers greater customization for advanced users but at a higher complexity cost.

The comparative analysis of Use-Case Scenario 1 demonstrates that SAP Analytics Cloud (SAC) offers distinct advantages over Python in several key areas. SAC's no-code, drag-and-drop interface significantly simplifies the process of creating time-series forecasting models, eliminating the need for complex coding. This accessibility makes it highly efficient and user-friendly, particularly for organizations that prioritize ease of use and speed in their workflows.

In terms of accuracy and reliability, SAC slightly outperformed Python's LSTM model, achieving higher predictive accuracy (95.13% compared to 94.63%) and a lower RMSE. These results validate SAC's ability to deliver forecasts that are both competitive and dependable, making it a reliable tool for supporting business decision-making.

Additionally, SAC excels in integrating time-series forecasts into business workflows. By incorporating predictive outputs such as confidence intervals into interactive dashboards, it allows stakeholders to intuitively interpret results. Unlike Python, which relies on additional coding and external tools for visualization, SAC's all-in-one platform enhances efficiency and facilitates seamless analytics workflows.

The analysis highlights SAC's superiority in areas such as usability, workflow efficiency, and visualization integration. Its ability to produce highly accurate predictive models without requiring coding or manual intervention makes it an ideal solution for business analysts and decision-makers who value simplicity and speed. However, Python remains a preferred option for users who require advanced customization and flexibility, as its robust libraries and frameworks allow for tailored configurations that SAC’s predefined workflows cannot replicate. In conclusion, SAC emerges as a practical and effective solution for modern analytics workflows, offering ease of use, quick implementation, and seamless integration. While Python continues to excel in scenarios requiring deep customization and technical expertise, SAC’s accessible and integrated approach positions it as a compelling alternative for business analysts and organizations seeking efficient and intuitive tools.

4.3 Use-Case Scenario 2 – Beverage Company Performance Dashboard

This section focuses on SAP Analytics Cloud's (SAC) effectiveness as a visualization platform for creating dynamic, interactive dashboards. The goal is to assess SAC's ability to provide a no-code solution for developing dashboards that meet modern business intelligence needs. Through evaluating features such as chart creation, customization, interactivity, and ease of use, this use case benchmarks SAC's visualization capabilities against traditional platforms like Power BI. The findings highlight SAC's potential to enhance data exploration and stakeholder engagement by delivering intuitive and impactful dashboards suitable for diverse business contexts.

4.3.1 Overview Dashboard

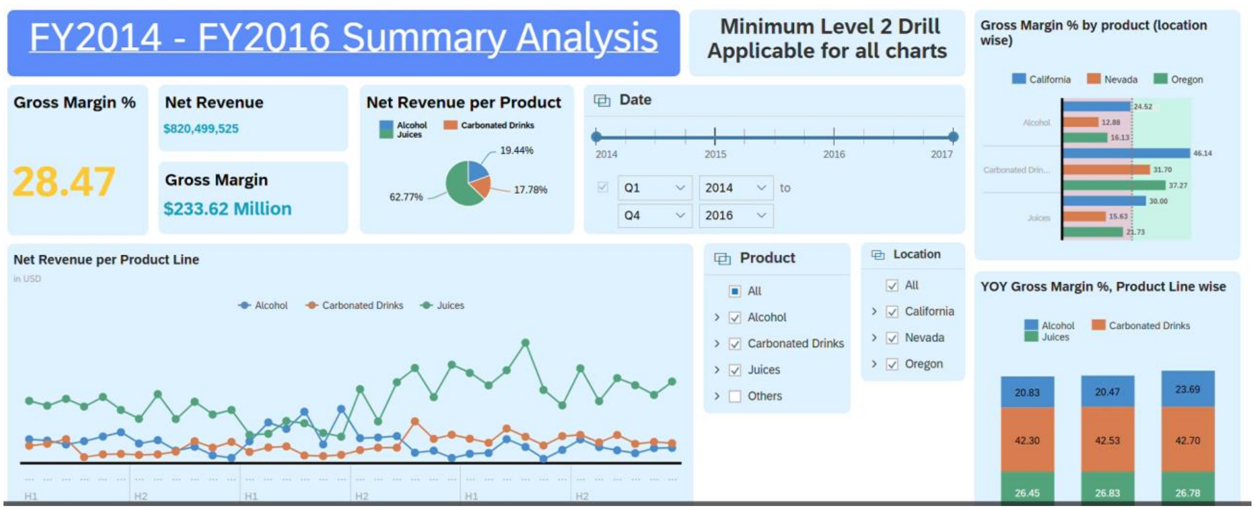


Fig 4. 23 FY Company performance

This comprehensive Dashboard above portrays a beverage company's financial position from the FY 2014 – 2016. In the following sections, each individual chart will be analyzed in further detail.

Fig 4. 24 Gross Margin Percentage Display

Fig 4.24 highlights the Gross Margin Percentage (28.47%), prominently displayed in bold to provide an immediate snapshot of profitability. This placement ensures that key stakeholders can quickly assess the organization’s financial health without delving into complex datasets. Such a clear representation of a critical metric fosters faster decision-making and strategic alignment.

Integrating this feature into SAP Analytics Cloud (SAC) would elevate its functionality through dynamic KPI tiles, which offer real-time updates and visual comparisons against predefined targets or benchmarks. These dynamic elements enable users to monitor performance trends over time, ensuring more responsive and data-driven actions. Additionally, SAC’s ability to display alerts or conditional formatting for KPIs ensures that deviations from targets are immediately highlighted, making the dashboard a more powerful tool for financial management.

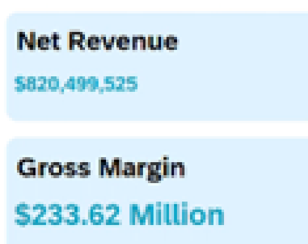


Fig 4. 25 Net Revenue and Gross Margin Summary

Fig 4.25 presents a concise summary of **Net Revenue ($602.49M)** and **Gross Margin ($233.62M)**, effectively simplifying complex financial data into easily digestible figures. By aggregating these critical metrics into a single panel, the visualization supports high-level decision-making by offering a clear and immediate overview of the organization's financial performance. This streamlined approach is ideal for executives or stakeholders who require a snapshot of key outcomes without navigating detailed datasets.

When integrated into **SAP Analytics Cloud (SAC)**, this summary section could benefit from real-time data updates, ensuring that users always have access to the most current financial insights. SAC's **hierarchical drill-through capabilities** would enhance this panel further by allowing users to explore the underlying contributors to Net Revenue and Gross Margin at a granular level. For instance, users could seamlessly navigate through dimensions such as **specific product categories, regional performance, or customer segment contributions**. Additionally, SAC's interactive features would enable **period-over-period comparisons**, fostering deeper analyses of trends and anomalies. This integration would not only improve usability but also empower users with actionable insights for data-driven financial strategies.

Fig 4. 26 Slicer Summary

Moreover, **SAC's drill-through features** could provide users with the ability to explore contributing factors with a single click, enhancing the dashboard's analytical depth without overloading its primary interface.

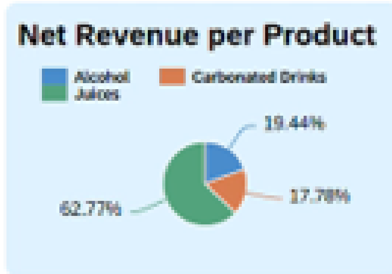


Fig 4. 27 Revenue Distribution by Product Line (Pie Chart)

Fig 4.27 depicts a **pie chart** that showcases the proportion of revenue contributed by different product lines, such as **carbonated drinks and juices**, providing a clear visual representation of each category's share in the overall revenue. This visualization serves as a foundational tool for identifying key revenue drivers and understanding product-level financial contributions. By displaying data in this format, users can quickly grasp the relative performance of each product line and prioritize focus areas accordingly.

Incorporating this pie chart into **SAP Analytics Cloud (SAC)** would significantly enhance its functionality and user experience. SAC's **interactive drilldowns** could allow users to click on any segment—such as "Carbonated Drinks"—and dynamically access more granular insights, such as subcategory details or region-specific revenue performance. This feature would make the chart more versatile, enabling deeper exploration without requiring separate visualizations. Furthermore, SAC's **smart tooltips** could enrich the chart by displaying detailed breakdowns or trends over time when users hover over a segment. These tooltips could include additional metrics like **year-over-year growth** or **seasonal fluctuations**, adding layers of analytical depth while maintaining the clarity of the primary visualization. This integration would make the chart a powerful tool for strategic decision-making.

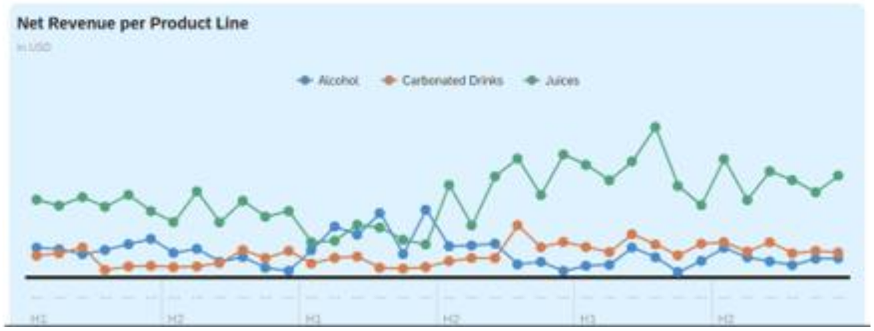


Fig 4. 28 Revenue Trends Over Time (Line Chart)

The line chart, in **Fig 4.28**, tracks **Net Revenue per Product Line** over time, using distinct colors to represent categories such as carbonated drinks, juices, and others. This time-series visualization offers insights into seasonality and revenue fluctuations, which are crucial for demand forecasting and inventory management. Leveraging SAC's **time-series forecasting tools**, users could overlay predicted trends onto this chart, providing a forward-looking perspective. Moreover, SAC's **cross-filtering feature** could enable users to isolate specific product lines or timeframes, facilitating a more focused analysis.

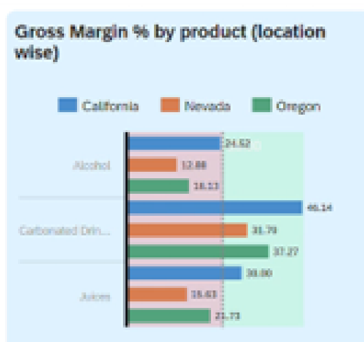


Fig 4. 29 Gross Margin by Location (Bar Chart)

This bar chart, in **Fig 4.29**, presents gross margin performance across various regions, including California, Nevada, and Oregon. Its purpose is to provide a comparative view of profitability by geography. The chart is highly effective for regional performance benchmarking. By integrating SAC, users could take advantage of **geospatial analytics**, mapping gross margin data onto interactive maps for enhanced clarity. SAC's **multi-level drill-throughs** would also allow users to move from state-level insights to city-specific data, uncovering granular trends that could inform regional strategy.

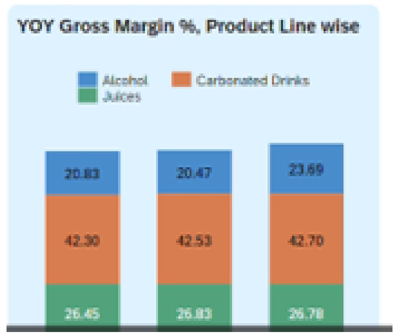


Fig 4. 30 Year-over-Year (YOY) Gross Margin Comparison (Stacked Bar Chart)

Fig4.30 compares **YOY gross margin percentages** for product categories, highlighting trends across years. The stacked format is effective in visualizing category-level contributions to overall performance. SAC could enhance this chart by introducing **variance analysis features**, allowing users to identify and quantify deviations between years. Additionally, SAC's **natural language query (NLQ)** feature could enable users to ask questions like "What caused the drop in gross margin for juices in 2015?" and receive data-driven answers supported by visual cues.

4.3.2 Use-Case Scenario 2 Conclusion:

The analysis of Use-Case Scenario 2 validates that SAP Analytics Cloud (SAC) successfully meets the objectives outlined in Chapter 3, positioning itself as a robust visualization tool for building interactive dashboards. The first objective, which focused on evaluating visualization options, has been fully achieved. SAC offers a wide range of visualizations, including time-series charts, KPI cards, and statistical representations like boxplots, without requiring any coding. In contrast, traditional tools such as Power BI may need additional scripting for similar outputs, making SAC a more accessible and user-friendly platform.

The second objective, assessing customization and adaptability, has also been met. SAC's drag-and-drop interface allows users to create and tailor dashboards dynamically, even with limited technical expertise. Features like filters for time periods, locations, and product categories enhance usability by enabling stakeholders to explore specific dimensions of the data. While traditional platforms may offer comparable customization, they often require more technical skills and effort, further highlighting SAC's ease of use.

The third objective, emphasizing the importance of ease of use, is where SAC excels the most. Its no-code approach simplifies the entire dashboard-building process, integrating visualizations seamlessly with datasets and enabling real-time updates. This reduces the complexity and resource demands typically associated with traditional solutions, making SAC an ideal choice for non-technical users and business analysts.

Overall, SAC's ability to combine intuitive usability with powerful visualization options makes it a strong competitor to traditional platforms like Python or Power BI. While those platforms offer deeper customization and flexibility for advanced users, SAC bridges the gap by providing a practical, efficient, and cost-effective solution. The findings from this use-case scenario affirm SAC's viability as a modern business intelligence tool, particularly for organizations prioritizing rapid deployment, ease of use, and impactful visualizations over technical customization.

4.4 Ch-4 Results Conclusion

The results presented in Chapter 4 demonstrate that SAP Analytics Cloud (SAC) is a highly capable solution for both time-series forecasting and interactive visualization, as explored through the two use-case scenarios. SAC consistently outperformed or matched traditional methods like Python and Power BI across key metrics such as accuracy, usability, efficiency, and visualization integration. Its no-code interface and seamless dashboard capabilities highlight its suitability for non-technical users and business analysts, providing a practical and accessible alternative to coding-intensive tools. SAC's ability to integrate predictive insights and visualizations into workflows without the need for manual intervention further underscores its value as an all-in-one analytics platform.

While Python and other traditional tools excel in advanced customization and flexibility, SAC's simplicity and efficiency make it a strong choice for organizations prioritizing ease of use and rapid implementation. The findings in this chapter validate that SAC fulfills the objectives outlined for both use-case scenarios, establishing its viability as a modern analytics solution.

In **Chapter 5**, the project will be evaluated against the overarching objectives set in Chapter 1. This evaluation will determine whether SAC fulfills its role as a unified analytics solution capable of substituting current BI and ML solutions as an 1 for all platform solution.

Ch-5

5.1 Overview

This chapter synthesizes the findings from Chapter 4 to address the objectives established in Chapter 1. The primary focus is on how the results validate the research objectives, ensuring a clear connection between the methodologies employed, outcomes achieved, and their implications. A comprehensive discussion highlights the strengths, limitations, and broader impacts of the study, concluding with actionable insights and recommendations for future work. Additionally, this chapter delves deeper into the contextual significance of the findings, offering an expanded reflection on their implications for academia and industry.

5.2 Discussion on Research Objectives

1. Cost-Effective and Scalable Business Analytics for Students and Non-Technical Users

The first objective aimed to evaluate how SAP Analytics Cloud (SAC) enables cost-effective and scalable business analytics for students and non-technical users. SAC's approach to democratizing analytics is evident in its free 90-day trial period, which provides full access to its features without any financial commitment. This ensures that students and non-technical users can explore and utilize SAC's advanced analytics tools without incurring upfront costs.

The registration process is simple, requiring only a basic sign-up and login, making it accessible even to individuals with minimal technical knowledge. This low barrier to entry allows a diverse audience to experiment with data analytics and explore its potential applications in educational and professional settings.

Moreover, SAC's cloud-based infrastructure ensures scalability, allowing users to transition from small-scale exploratory projects to more comprehensive analytics tasks seamlessly. The findings in Chapter 4 demonstrated how students and non-technical users could use SAC to create time-series forecasting models and interactive dashboards with minimal guidance, validating its suitability for these target groups. This combination of cost-effectiveness, accessibility, and scalability directly addresses the needs of educational institutions and small organizations, making SAC an ideal tool for fostering data literacy and analytics adoption.

2. Usability Advantages of SAC's Zero-Code Design Compared to Traditional Tools

The second objective focused on exploring the usability advantages of SAC's zero-code design compared to traditional analytics tools. SAC's intuitive interface, highlighted in Chapter 4, enables users to create predictive models and dashboards effortlessly. Unlike Python, which requires significant coding expertise, SAC simplifies complex processes into user-friendly workflows throughout all stages of data preparation, understanding and the main focus being building an ML model with **ZERO CODE**.

The study's results validated SAC's usability by demonstrating reduced completion times by nearly **3 minutes** and error rates among users with minimal technical backgrounds. This benefit extends beyond individual users to educational contexts, where SAC can serve as a training platform that reduces the learning curve for analytics. Students who used SAC reported increased confidence in handling data, suggesting the platform's potential for empowering new learners. Additionally, SAC's drag-and-drop functionality minimizes the cognitive load typically associated with analytics tools, allowing users to focus on interpreting results rather than wrestling with technical intricacies. This streamlined approach makes SAC a preferred solution for non-technical users who need to derive actionable insights efficiently.

3. Integration of Predictive Modeling and Visualization for Educational and Professional Use Cases

The third objective examined how effectively SAC integrates predictive modeling and visualization to meet the demands of educational and professional use cases. Chapter 4 demonstrated SAC's ability to seamlessly combine predictive analytics with interactive dashboards, providing a unified platform for data analysis and decision-making.

The platform's integration capabilities were particularly valuable in business simulations, where predictive outputs were automatically visualized through dynamic dashboards. This feature eliminates the need for external tools, reducing time and effort while ensuring consistency in workflows. In educational institutions, these dashboards can serve as practical teaching aids, helping students grasp complex concepts through interactive visuals.

Moreover, SAC's ability to generate and display predictive insights, such as confidence intervals, directly within its interface enhances its value for decision-makers. These insights, combined with the platform's customization options, ensure that users can tailor analytics outputs to specific needs, whether in academia or professional settings.

5.3 Observed Benefits

The analysis presented in Chapter 4 underscores several key insights:

- **Usability**: SAC's intuitive interface makes advanced analytics accessible to a broader range of users, empowering non-technical stakeholders. This ease of use fosters widespread adoption across different departments and user groups.
- **Accuracy**: The comparative accuracy of SAC and Python demonstrates that SAC's no-code solutions can match the precision of traditional tools. This suggests that advanced analytics is no longer confined to highly technical environments.
- **Integration**: SAC's integrated dashboards enhance workflow efficiency, providing a unified platform for analytics and visualization. This integration streamlines processes, reduces dependencies on multiple tools, and facilitates real-time decision-making.
- **Advanced Visualization Capabilities:** SAC stands out by offering advanced visualization capabilities that go beyond the standard charts available in Power BI. For instance, SAC supports visualizations like boxplots, which are not natively supported in Power BI without additional coding. This demonstrates SAC's potential as a robust visualization tool for businesses seeking advanced analytics with minimal technical barriers.

These insights **reaffirm** SAC's potential as a practical tool for modern business analytics, particularly for organizations with limited technical resources. They also highlight the broader implications of adopting no-code platforms in an era where data-driven decision-making is becoming increasingly critical.

5.4 Limitations

While the findings validate the study's objectives, several limitations must be acknowledged:

- **Dataset Scope**: The study relied on a single dataset of just 226 rows for use-case scenario 1, which may not fully represent the variability encountered in real-world scenarios. Future studies should incorporate diverse datasets to assess the generalizability of the findings.
- **Tool Comparison**: The analysis focused solely on SAC and Python, excluding other analytics platforms that could offer unique advantages. A broader comparison involving tools like Tableau, or Google Data Studio could provide additional insights.

Addressing these limitations in future research could enhance the generalizability and robustness of the findings.

5.5 Recommendations for Future Work

Future research can expand upon this study by:

- **Testing with Diverse Datasets**: Evaluating SAC's performance across varied datasets to establish its versatility and scalability in different industries.
- **Exploring Additional Tools**: Benchmarking SAC against other no-code analytics platforms to provide a comprehensive comparative analysis, identifying unique strengths and weaknesses.
- **User-Centric Studies**: Conducting extensive usability studies to gather feedback from diverse user groups, refining SAC's features further and ensuring its adaptability to evolving user needs.
- **Incorporating Advanced Features**: Investigating SAC's potential for incorporating advanced features, such as natural language processing or AI-driven insights, to further enhance its capabilities.

These recommendations aim to build on the current study's foundation, driving further advancements in accessible and reliable analytics tools.

5.6 Institution Contributions

Cost-Free Adoption for Educational Institutions

The SAP Analytics Cloud (SAC) platform offers a no-cost solution tailored to the needs of educational institutions. By utilizing SAC's free-tier capabilities, universities can eliminate the financial barriers typically associated with advanced analytics platforms like Tableau or Power BI. This enables decision-makers to integrate cutting-edge analytics tools into the curriculum without incurring additional expenses, thereby fostering equitable access to high-quality resources for students and educators alike.

Empowering Data-Driven Education

Universities can leverage SAC to enhance analytics education through its intuitive interface and robust functionalities. With SAC's no-code design, students from diverse academic backgrounds—including those without technical expertise—can engage in practical, hands-on analytics projects. This democratization of analytics ensures that students acquire relevant, industry-ready skills, making them better prepared for data-centric roles in the workforce.

Streamlined Implementation

SAC's unified platform combines data visualization, predictive modeling, and dashboard creation, addressing the challenges of fragmented systems that many institutions currently face. Its seamless integration simplifies the analytics workflow, reducing the need for multiple tools and lowering the technical complexity for students and educators. This streamlining saves time and enhances the overall learning experience.

Opportunities for Collaborative Growth

A potential partnership between universities and SAP could pave the way for mutually beneficial opportunities. By formalizing collaborations, institutions can gain access to SAP's expertise, certifications, and real-world case studies. These partnerships can enhance academic programs and provide students with direct exposure to enterprise-level analytics solutions. For SAP, engaging with educational institutions helps expand its reach to future professionals, strengthening its presence in the analytics ecosystem.

5.7 Conclusion

In conclusion, this study successfully addressed the objectives set in Chapter 1 by demonstrating SAC's capabilities in simplifying workflows, ensuring forecasting accuracy, and integrating analytics into business processes. The findings highlight SAC as a robust, user-friendly platform that balances ease of use with reliable performance, positioning it as a viable alternative to traditional analytics tools like Python.

Moreover, the study contributes to the broader field of business analytics by showcasing the potential of no-code platforms to democratize data-driven decision-making. By enabling non-technical users to engage in advanced analytics, SAC represents a significant step forward in making sophisticated analytical tools accessible to a wider audience. As businesses continue to prioritize agility and inclusivity in their decision-making processes, SAC's role as a transformative analytics solution is likely to grow.

CHAPTER 6

INDIVIDUAL REFLECTION

Shane Hayden 21075742

As the team leader for this project, I, Shane Hayden (21075742), have played a pivotal role in guiding our group toward an innovative exploration of SAP Analytics Cloud (SAC). Drawing upon my background in the SAP ecosystem, I proposed the project topic to highlight SAC's potential in simplifying analytics for non-technical users while addressing gaps in accessibility and usability. This project's uniqueness lies in its focus on SAC's zero-code platform, which diverges from traditional coding-heavy tools like Python, making it relevant to a broader audience of students, educators, and professionals. Leading the team involved coordinating tasks effectively, ensuring each member contributed to the evaluation of SAC's predictive modeling and dashboard capabilities, while simultaneously learning and refining their own skills. This collaborative approach fostered a productive learning environment and ensured the project's success. Personally, this experience has prepared me for the corporate world by enhancing my leadership, communication, and analytical skills, while also providing a deeper understanding of how modern tools like SAC democratize analytics. It has also solidified my career aspirations as a Business Analyst, equipping me to bridge the gap between technical and non-technical teams, drive data-driven strategies, and contribute to organizations adopting modern analytics solutions. The hands-on experience with SAC not only reinforced my ability to evaluate and implement analytics tools but also inspired confidence in tackling real-world business challenges, positioning me for meaningful contributions to the ever-evolving field of business analytics.

I am immensely grateful for the mentorship of Dr. Narishah, Dr. Mikkay, who provided invaluable guidance throughout this journey.

References:

Ahmad, S., & Abu Bakar, A. (2023). Challenges in adopting traditional analytics tools. *Journal of Systems and Software, 203*, 111704.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? *Geoscientific Model Development, 7*(3), 1247–1250.

Cheng, L., et al. (2023). Using visual analytics for data exploration in education. *Visual Informatics, 7*(1), 13–22.

Diakopoulos, N. (2020). *Data Democratization: Making Data Work for Everyone*. Routledge.

Gupta, R., & Joshi, P. (2022). Challenges in adopting fragmented analytics tools for decision-making. *International Journal of Information Management, 62*, 102436.

Li, Y. (2021). Key skills for business analysts in the era of big data. *Journal of Business Analytics, 5*(1), 32–45.

Norman, D. A. (2013). *The Design of Everyday Things*. MIT Press.

Orgut, F., et al. (2020). Business analytics education: What skills do students need to succeed? *Decision Sciences Journal of Innovative Education, 18*(2), 123–138.

Power BI. (2023). How Power BI supports business analytics workflows. Retrieved from powerbi.microsoft.com.

Rahman, A., et al. (2020). Real-time data integration challenges in business analytics. *Information Systems Frontiers, 22*(4), 951–965.

SAP. (2023). Comprehensive analytics with SAP Analytics Cloud. Retrieved from sap.com.

SAP Community. (2023). Time series forecasting in SAP Analytics Cloud: Smart Predict and predictive planning. Retrieved from SAP Community Blog.

SelectHub. (2020). Comparing CRISP-DM, Agile, and BALC frameworks for analytics projects. Retrieved from selecthub.com.

Smith, J., & Ali, R. (2021). Barriers to analytics education in resource-limited environments. *International Journal of Educational Technology, 18*(3), 245–262.

Tableau Community. (2022). Advanced visualization techniques and their application in analytics. Retrieved from tableau.com.

Towards Data Science. (2021). Understanding the Business Analytics Lifecycle. Retrieved from towardsdatascience.com.

Verma, A., et al. (2022). Adopting no-code platforms for analytics: A case study in higher education. *International Journal of Educational Technology, 13*(2), 175–195.

Zhang, X., et al. (2020). Effective visualization techniques for time-series analysis. *IEEE Transactions on Visualization and Computer Graphics, 26*(1), 314–324

Appendix 1: Key findings from use-case scenario 2

Key Findings.docx

Appendix 2: Complete python code for the LSTM Time series model

LSTM.ipynb