AFFECTIVE SALES: EMOTIONALLY DESIGNED SMART DASHBOARD FOR MULTI-MODEL SALES FORECASTING

A PROJECT REPORT

Submitted by

JAI J (311621243009)

SANTHOSH G (311621243021)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE

THORAIPAKKAM, CHENNAI - 600097

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

ANNA UNIVERSITY, CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“AFFECTIVE SALES: Emotionally Designed Smart Dashboard for Multi-Model Sales Forecasting”** is the bona-fide work of “**JAI J (311621243009)** & **Santhosh G (311621243021)**” who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
MRS. A. Jeyanthi M.E, (Ph.D.)	Mr. D Vinod M.E.CSE, (Ph.D.)
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSISTANT PROFESSOR
Department of Artificial	Department of Artificial
Intelligence and Data Science	Intelligence and Data Science
Misrimal Navajee Munoth Jain	Misrimal Navajee Munoth Jain
Engineering College,	Engineering College,
Thoraipakkam, Chennai-600097	Thoraipakkam, Chennai-600097

Submitted for the Project Viva–voce examination held on _______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

MISRIMAL NAVAJEE MUNOTH JAIN ENGINEERING COLLEGE

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

- VISION

To produce high quality, creative and ethical engineers and technologists contributing effectively to the ever-advancing Artificial Intelligence and Data Science field.

- MISSION

To educate future software engineers with strong fundamentals by continuously improving the teaching-learning methodologies using contemporary AI & DS. To produce ethical engineers/researchers by instilling the values of humility, humaneness, honesty and courage to serve society. To create a knowledge hub of Artificial Intelligence and Data Science with everlasting urge to learn by developing, maintaining and continuously improving the resource Artificial Intelligence / Data Science.

ACKNOWLEDGEMENT

We express our sincere gratitude to our honourable Secretary (Administration) **Dr. Harish L Mehta** and Secretary (Academic) **Shri. L. Jaswant Munoth** for providing the infrastructure facilities to do this project during our course period.

We thank our **Principal, Dr. C. Chandrasekar. Christopher**, for providing the infrastructure and resources required for the completion of this project.

We express profound sense of gratitude and thanks to our **Head Of The Department**, **Mrs. A. Jeyanthi M.E., (Ph.D.), Associate Professor** for her kind words and enthusiastic motivation which inspired us a lot in completing this project.

We thank our project supervisor **Mr. D. Vinod ME.CSE, (Ph.D.)** for his valuable suggestions for the development and completion of this project. We thank him for him wholehearted guidance right from the inception to the completion of our project.

We thank our project coordinator, **Mrs. G. Priyaa Sri M.E., (Ph.D.)**, Assistant Professor, for her valuable suggestions for the development and completion of this project

Finally, we thank all the Teaching and Non-Teaching Staff members of our department who helped us to complete our project.

Above all, we thank the Almighty, our parents and siblings for their constant support and encouragement for completing this project.

ABSTRACT

In the modern business landscape, data-driven decision-making is crucial. However, traditional dashboards often fall short by only presenting historical data without predictive insight or emotional engagement.

This project addresses those limitations by integrating machine learning with affective visual storytelling. We employ three distinct forecasting models XGBoost, Prophet, and LSTM each optimized for different aspects of time series and structured prediction. These models are trained using cleaned transactional data from AdventureWorks, enabling the prediction of future sales trends with high accuracy. Evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score are used to compare model performance.

The predicted outputs are visualized through an interactive Power BI dashboard designed to communicate not just data, but insights—via KPI cards, comparative trendlines, model performance metrics, and narrative elements. The result is a smart dashboard that empowers both technical and non-technical users to make faster, smarter, and emotionally intuitive business decisions.

This project demonstrates how AI, when combined with emotional design principles and user-centric storytelling, can transform raw data into powerful strategic tools.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	EXPANSION
1	XGBoost	Extreme Gradient Boosting
2	LSTM	Long Short-Term Memory
3	RMSE	Root Mean Square Error
4	MAE	Mean Absolute Error
5	MAPE	Mean Absolute Percentage Error
6	KPI	Key Performance Indicator
7	BI	Business Intelligence
8	SQL	Structured Query Language
9	MSE	Mean Squared Error
10	CSV	Comma-Separated Values

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SALES FORECASTING

Sales forecasting is a critical component of strategic business planning. It involves predicting future sales revenue using historical data, market trends, and analytics. Accurate sales forecasts enable organizations to make informed decisions on budgeting, resource allocation, inventory management, and customer engagement. With the rise of digital transformation, companies are transitioning from traditional forecasting methods to data-driven approaches leveraging artificial intelligence (AI) and machine learning (ML).

The demand for intelligent systems that can adapt, learn from data patterns, and make precise predictions has surged. In this context, the application of models like XGBoost, Prophet, and LSTM in forecasting frameworks has proven highly effective, especially when integrated with interactive visual tools like Power BI.

1.2 IMPORTANCE OF FORECASTING IN BUSINESS STRATEGY

Forecasting not only supports revenue planning but also mitigates risks by preparing businesses for market fluctuations. It assists stakeholders in determining the viability of marketing campaigns, launching new products, or expanding operations. Precise forecasting models contribute to:

- **Efficient inventory management:** Preventing stock-outs or overstock.
- **Optimized financial planning:** Guiding investments and controlling operational costs.

- **Improved customer satisfaction:** Ensuring product availability and timely delivery.
- **Strategic decisions:** Backing mergers, acquisitions, and expansion strategies.

As markets become increasingly competitive, leveraging AI-based forecasting tools enhances a company's responsiveness and resilience.

1.3 MOTIVATION FOR THE PROJECT

The motivation behind this project stems from the need to modernize sales forecasting mechanisms in today's dynamic market environments. Traditional statistical methods, while useful, often fail to account for nonlinear relationships, seasonality, and unexpected disruptions such as pandemics or supply chain issues.

Integrating advanced machine learning models—such as **XGBoost** for structured data, **Prophet** for time series decomposition, and **LSTM** for deep sequential learning—offers a robust solution. Additionally, visualizing the forecast in a **Power BI dashboard** enhances interpretability and user engagement, providing decision-makers with real-time, actionable insights.

1.4 PROBLEM STATEMENT

Traditional forecasting methods struggle with adaptability, accuracy, and responsiveness to market anomalies. Existing systems often lack emotional and interactive components that help stakeholders understand and trust the data. There is a need for:

- A hybrid, multi-model approach to improve forecast accuracy.

- An emotionally intuitive dashboard that presents data in a human-centric format.
- Automation and intelligence to minimize manual intervention.

This project addresses these gaps through a comprehensive, AI-driven, emotionally designed smart dashboard for multi-model sales forecasting.

1.5 OBJECTIVES OF THE PROJECT

The primary objectives of this project are:

1. To build and compare multiple forecasting models: XGBoost, Prophet, and LSTM.
2. To integrate forecasting results into a unified Power BI dashboard.
3. To design a user-friendly and emotionally intuitive interface that aids decision-making.
4. To analyze and present model accuracy using standard evaluation metrics (MAE, RMSE, MAPE).
5. To enable dynamic exploration of sales data by region, product, and time period.
6. To propose future improvements including external data integration and feedback loops.

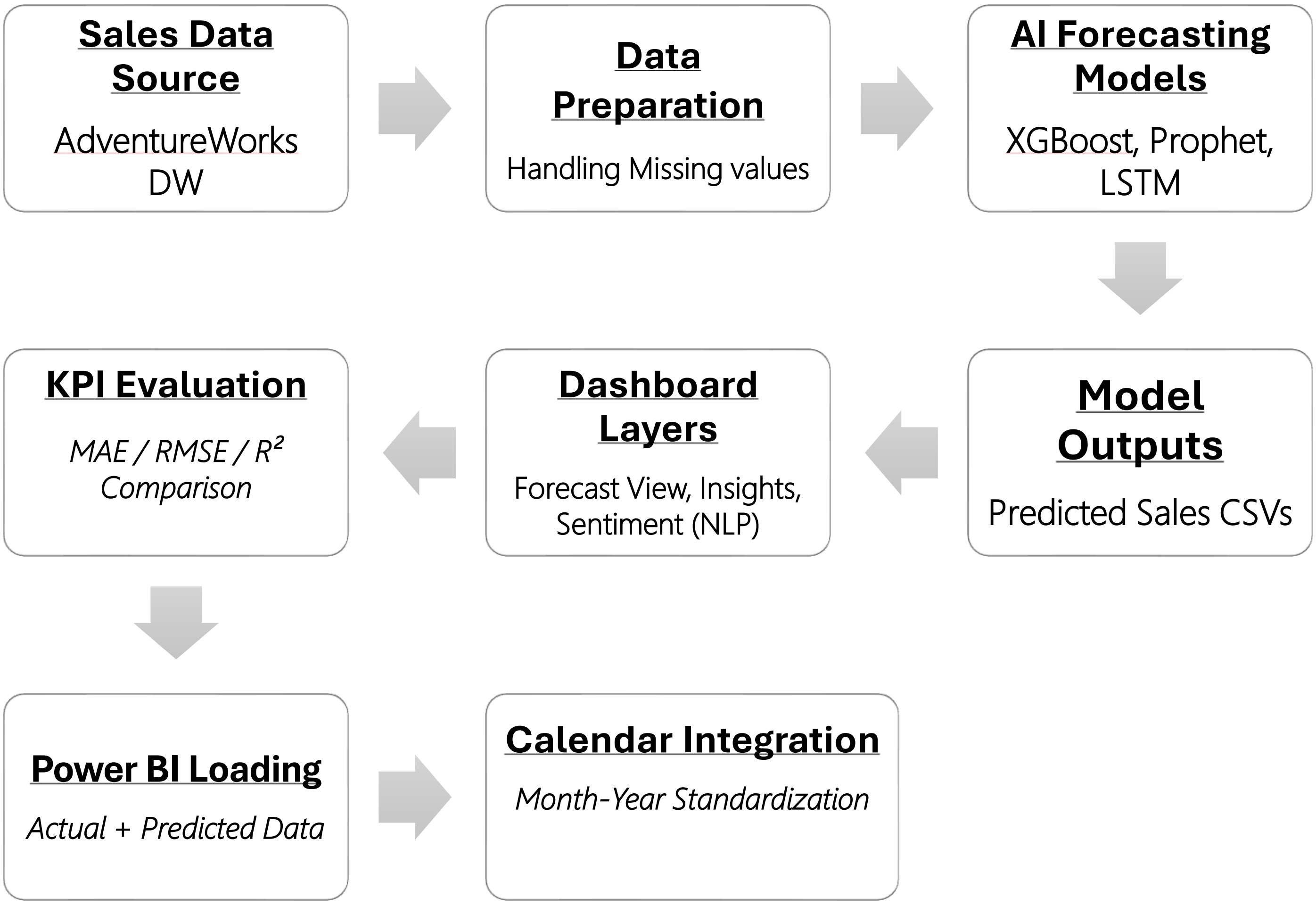


***Figure 1.1**: Workflow of Sales Forecasting Process*

As illustrated in Figure 1.1, This diagram outlines the sales forecasting process using Power BI. It starts with data from the AdventureWorks data warehouse, followed by preparation to handle missing values. Forecasting models like XGBoost, Prophet, and LSTM generate sales predictions, which are exported as CSVs. Model performance is evaluated using MAE, RMSE, and R^2. Actual and predicted data are then loaded into Power BI dashboards with insights and sentiment analysis. Calendar integration ensures standardized time formatting for consistent reporting.

CHAPTER 2

LITERATURE SURVEY

Ayyoub Frifra et al., 2024 [1] addressed the significant challenge of storm prediction, which is difficult due to their infrequent nature but critical for minimizing damage and saving lives. They proposed a novel machine learning approach combining LSTM and XGBoost algorithms to forecast storm characteristics and occurrence in Western France.The study utilized historical data from buoys and a storm database spanning from 1996 to 2020, employing data from 1996-2015 for training and validation, and 2016-2020 for prediction, with LSTM specifically for predicting characteristics and XGBoost for predicting occurrence. Experimental findings indicated that the LSTM model performed well in forecasting temperature and pressure, although it faced challenges in accurately capturing extreme values for wave height and wind speed. The XGBoost model, in contrast, showed exceptionally strong performance in predicting storm occurrence. (The ability to better predict storms has a direct positive emotional impact by potentially reducing fear, damage, and loss of life). The study contributes a valuable methodology for storm prediction by effectively integrating deep learning and machine learning techniques. This approach offers practical implications for mitigating the impact of storms on human lives and infrastructure. Explicit directions for future research were not mentioned in the provided abstract.

Célia Talma Gonçalves et al., 2023 [2] investigated the impact of business intelligence tools on organizational decision-making, focusing on developing integrated performance dashboards using Power BI for sales marketing. Employing the Vercelli's methodology, they utilized a dataset from Super Data Science to create key performance indicators (KPIs) for business decisions. The study implemented dashboards to visualize sales trends, customer behavior, and market performance, analyzing their effect on decision-making efficiency. Findings indicate that Power BI dashboards enhance real-time data access, improving decision accuracy and speed, though user training is critical for effective adoption. Emotional engagement was indirectly addressed through user-friendly visualizations that boost confidence in decisions. Challenges include data integration complexity and the need for standardized KPI frameworks.

Hassan Oukhouya et al., 2023 [3] developed and evaluated machine learning models to address the crucial need for accurate forecasting of international stock market trends, driven by the inherent non-linearity of stock prices. The study compared the performance of LSTM, XGBoost, and a hybrid LSTM-XGBoost model, utilizing the skforecast library for back testing on historical daily prices of key international stock indices. Through experimental analysis, the researchers evaluated the effectiveness of these algorithms for stock price prediction. Findings demonstrate that the hybrid LSTM-XGBoost model, optimized using Grid Search, significantly outperforms the individual LSTM and XGBoost models, achieving higher accuracy in forecasting daily prices. While emotional impact is not directly addressed, precise forecasts indirectly enhance confidence among financial analysts and investors. Explicit challenges faced during the study were not detailed in the provided text. The study contributes by validating the effectiveness of the hybrid LSTM-XGBoost model and providing a comparative analysis of machine learning techniques for financial forecasting. Specific future research directions were not explicitly stated. It offers practical insights for financial analysts and investors aiming to make informed decisions based on precise stock market trend predictions.

Sean Kandel et al., 2012 [4] explored the practices and challenges of enterprise data analysts through a qualitative interview study, focusing on how visualization supports decision-making in organizational contexts. They conducted semi-structured interviews with 35 data analysts across sectors like finance, healthcare, and technology, analyzing workflows, tool usage, and visualization needs. Thematic analysis revealed key themes: the importance of iterative data exploration, the role of visualizations in communicating insights, and barriers like data complexity and tool limitations. Findings show that effective visualizations, such as interactive dashboards and charts, enhance stakeholder engagement and decision clarity, with emotional resonance indirectly fostered through intuitive designs that build trust. Challenges include integrating diverse data sources and addressing varying stakeholder expertise. The study contributes a detailed understanding of enterprise visualization needs, offering design implications for tools that support collaborative analysis. It emphasizes the value of user-centered visualizations in bridging technical and emotional aspects of decision-making, calling for further research into scalable, accessible visualization solutions that balance analytical rigor with emotional impact in enterprise settings.

Xingyu Lan et al., 2023 [5] explored the undervaluation of emotion in data visualization by analysing 109 studies to understand how emotions enhance visualization design. They categorized the research into three lines—emotion in visualization evaluation, emotion as a design factor, and emotion-driven design—focusing on studies that examine how visualizations convey and influence emotion.The researchers coded these studies to identify five perspectives justifying the importance of emotion and evaluated 61 affective visualization projects across design fields, tasks, and methods. Their findings reveal that emotional elements, such as warm colours and narratives, heighten engagement but risk bias if not carefully balanced. Challenges include unclear definitions, limited justification for emotion's role, and an underexplored design space. The study contributes a comprehensive framework for affective visualization design, mapping the design space and addressing ethical concerns and the need for standardized emotional metrics. It organizes the field, establishes a research agenda, and emphasizes ethical, inclusive design to balance emotional resonance with analytical clarity, offering a foundation for advancing affective visualization research. The study contributes practical guidance for implementing Power BI dashboards, emphasizing their role in transforming raw data into actionable insights. It highlights the potential for visualizations to support strategic decisions in dynamic business environments, calling for further research into scalable dashboard designs and their emotional impact on stakeholders.

Zhoufan Chen et al., 2023 [6] developed a novel Prophet-LSTM model to address the growing need for accurate peak load forecasting in distribution networks, driven by rising power consumption. The study utilized historical load data and meteorological factors, combining the Prophet model's trend-fitting capabilities with the LSTM model's high prediction accuracy. Through experimental analysis, the researchers modeled distribution network peak loads, evaluating the algorithm's effectiveness. Findings demonstrate that the Prophet-LSTM model outperforms traditional forecasting methods, achieving improved accuracy and stability in predicting peak loads.While emotional impact is not directly addressed, intuitive visualizations of load trends indirectly enhance stakeholder confidence in grid management decisions. Challenges include data quality issues and the need for real-time data integration. The study contributes a robust forecasting tool for optimizing power grid operations, supporting sustainable energy management. It calls for further research into scalable models and their application across diverse network conditions, offering practical insights for utilities aiming to balance load demands efficiently.

CHAPTER 3

SYSTEM ANALYSIS

3.1 REQUIREMENT ANALYSIS

Requirement analysis is the process of identifying the precise functional and non-functional expectations of a software system. It provides the foundation for system architecture, design, development, and deployment. In this project, we focused on collecting and analyzing requirements across hardware, software, and data components essential for multi-model forecasting and dashboard visualization.

3.1.1 Hardware Requirements

Component	Specification
Processor	Intel i5 or AMD Ryzen 5 or higher
RAM	Minimum 8 GB (Recommended: 16 GB)
Storage	256 GB SSD or above
Display	1080p, 14” or larger
GPU (Optional)	NVIDIA GTX/RTX (for faster deep learning training)

Table 3.1: Hardware Requirements for Forecasting System

3.1.2 Software Requirements

Software	Purpose
Windows 11	Operating System
Python (v3.9+)	Forecasting model development
Jupyter Notebook (v*7.4.0*.) / VS Code	Python programming and execution
Microsoft Power BI Desktop	Dashboard visualization
Excel / CSV Tools	Data handling and preprocessing

Table 3.2: Software Tools and Purpose

3.1.3 Dataset Requirements

- The forecasting models use a structured dataset containing fields like Order_Date, Sales_Amount, Product_Category, Region, and Customer_Segment.
- Data source: Adventure Works database in .BAK format
- Data preprocessing includes handling missing values, formatting dates, normalizing values, and creating lag-based features for time series analysis.

3.2 FEASIBILITY STUDY

Feasibility study examines whether the proposed system is viable from different dimensions:

- **Technical Feasibility:**
  The technologies used (Python, Power BI) are well-established, free or

affordable, and compatible with standard computing systems. Multi-model forecasting is technically feasible using existing ML libraries.

- **Operational Feasibility**:
  The dashboard design is intuitive and understandable by non-technical business users, making the project viable in operational environments.
- **Economic Feasibility**:
  All software used is open-source or free (Python, Jupyter, Power BI Desktop), making this solution highly cost-effective for SMEs or academic institutions.
- **Schedule Feasibility**:
  The entire project, including design, development, testing, and integration, was completed within a structured academic project timeline (approx. 12–14 weeks).

3.3 STRATEGIC EVALUATION OF THE PROPOSED SYSTEM

A strategic evaluation is essential to understand the internal strengths and weaknesses of the proposed system, along with potential opportunities and external threats. This analysis helps in assessing the system's viability, scalability, and real-world adaptability.

STRENGTHS:

- Multi-model forecasting improves prediction accuracy
- Emotionally designed dashboard enhances engagement
- Integration with Power BI boosts accessibility

Opportunities	Threats
Extension into other domains (retail, finance, logistics)	Market volatility may affect forecast reliability
Real-time analytics with live data sources	Data privacy concerns when scaling to production
Incorporating sentiment from social media	Overfitting risks with deep learning architectures

Table 3.3: Opportunities and Threats

CHAPTER 4

SYSTEM ARCHITECTURE

4.1 ARCHITECTURE DIAGRAM

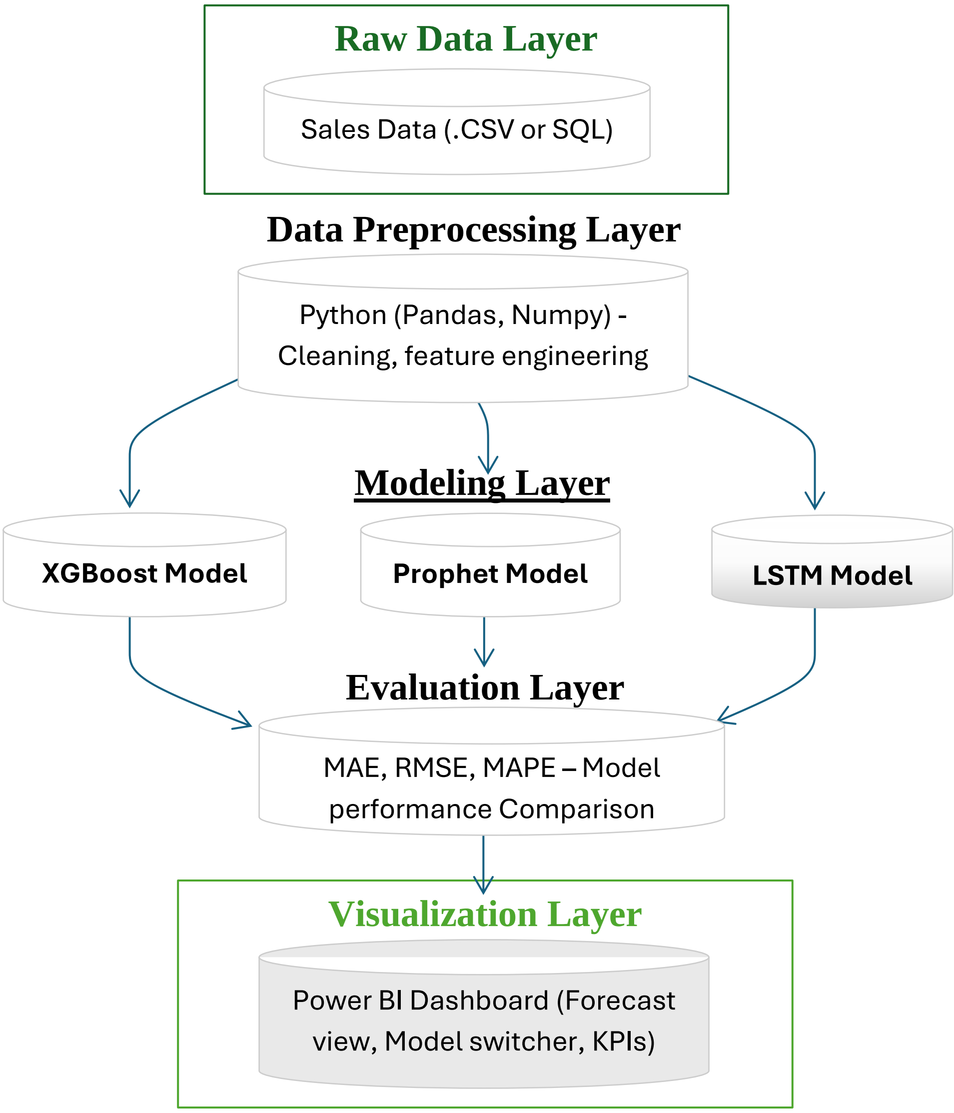


Figure 4.1: *System Architecture Diagram*

As illustrated in Figure 4.1, System architecture defines the high-level structure and flow of the proposed sales forecasting solution. It describes the components, technologies, and interactions among modules—from data ingestion to machine learning model execution and final dashboard visualization. This layered architecture ensures modularity, scalability, and real-time responsiveness in enterprise environments.

Explanation:
The architecture is composed of four major layers:

1. Data Layer
    - Source: Sales datasets in CSV or SQL format
    - Preprocessing: Null removal, outlier treatment, time formatting
2. Model Layer
    - Models: XGBoost, Prophet, LSTM
    - Python environment used for training and exporting model outputs
3. Evaluation & Transformation Layer
    - Metrics: MAE, RMSE, MAPE
    - Model outputs saved as structured .csv format
4. Dashboard & Visualization Layer
    - Power BI used to import forecast data
    - Visual elements: bar charts, line graphs, KPI cards
    - Features: filter by model, region, and product

This architecture allows a clear flow from raw data to actionable insight.

4.2 MODULE INTERACTION

Each module is tightly integrated to allow seamless flow of data from ingestion to visualization. Here's a detailed interaction map:

Module Name	Functionality
Data Collection	Pulls sales data from SQL/CSV and extracts relevant fields
Preprocessing	Cleans, encodes, and prepares the dataset for model consumption
XGBoost Forecasting	Trains a gradient boosting regression model for structured data forecasting
Prophet Forecasting	Applies decomposed time series modeling to capture trend and seasonality
LSTM Forecasting	Deep learning-based sequence model for long-term forecasting
Metrics & Evaluation	Calculates accuracy scores for each model to compare performance
Forecast Consolidation	Merges predictions from all models and actuals for dashboard integration
Power BI Dashboard	Loads all forecasts and KPIs into a user-friendly dashboard interface

Table 4.1: Module-wise Interaction in Forecasting System

Control Flow:

- Modules are executed sequentially from data extraction to model training.
- The final .csv outputs are reused across Power BI for visualization.
- Each model runs independently, allowing for parallelization and easy maintenance.

4.3 STAR SCHEMA DATA ARCHITECTURE

In order to effectively analyze and visualize forecasting data, a star schema design was adopted in the backend data model used in Power BI. A star schema is a common dimensional modeling technique used in data warehousing that organizes data into fact and dimension tables to support efficient querying, aggregation, and reporting.

As illustrated in Figure 4.2, this schema allows the system to manage sales data with logical relationships between entities such as time, region, product, and customer. The structured format enables seamless integration with Power BI's visualization and slicing capabilities.

Components of the Star Schema

- **Fact Table – FactSalesForecast**

  This central table stores quantitative data such as actual sales, predicted sales, forecast error, and time-period-based revenue. It references all the dimension tables through foreign keys.
- **Dimension Tables**:
  - **DimDate** – Contains date-related fields like year, month, quarter.
  - **DimProduct** – Describes product names, categories, and SKUs.
  - **DimCustomer** – Contains information such as customer region, segment, and ID.

- **DimRegion** – Provides geographic attributes such as state, country, and region.
- **DimModel** – Identifies which model (XGBoost, Prophet, LSTM) generated each forecast entry.

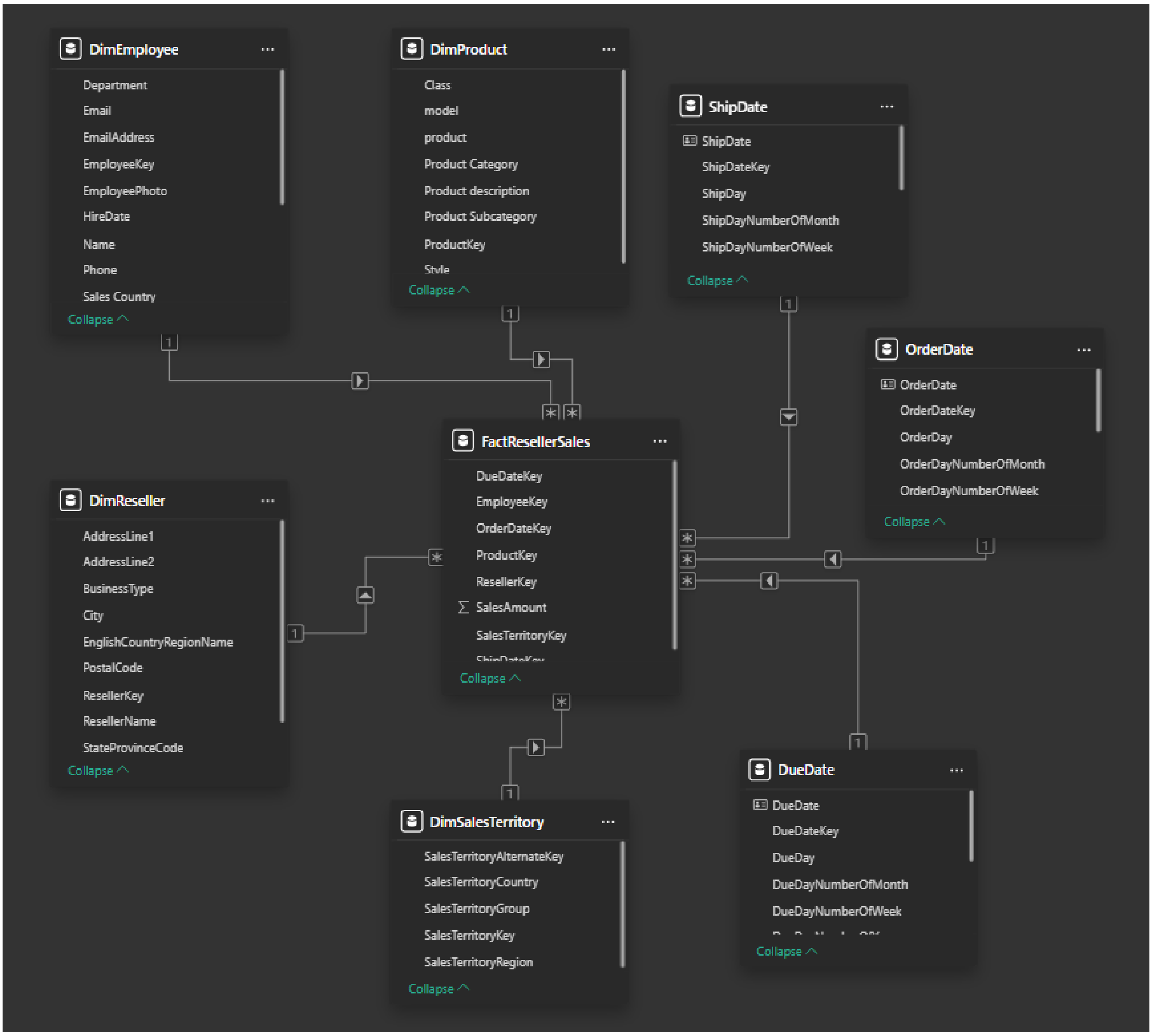


***Figure 4.2**: Star Schema Representation of Forecasting Data Warehouse*

Each dimension table is linked to the fact table using a one-to-many relationship, with the fact table capturing transactional or measurable data and the dimension tables offering context for analysis.

Key Relationships and Primary Keys in the Star Schema

In the implemented Star Schema, the central component is the FactSalesForecast table, which contains all the quantitative data — such as actual sales, predicted sales from different models (XGBoost, Prophet, LSTM), forecast error values, and timestamps. This fact table is surrounded by several dimension tables that provide contextual details necessary for slicing, filtering, and grouping data in Power BI.

To maintain the structural integrity and support seamless querying, primary and foreign key relationships are used between the fact and dimension tables.

1. Primary Keys (PKs)

Each **dimension table** contains a column designated as the **primary key**. This key uniquely identifies each record in the dimension and ensures that every lookup value (e.g., a specific product, date, or region) can be distinctly referenced.

Examples of primary keys:

- DimDate[DateID]
- DimProduct[ProductID]
- DimCustomer[CustomerID]
- DimRegion[RegionID]
- DimModel[ModelID]

2. Foreign Keys (FKs)

The **FactSalesForecast** table contains **foreign keys** corresponding to the primary keys in each dimension table. These foreign keys create **one-to-many relationships**, where:

- One record in a dimension table (e.g., one product) can relate to **many** records in the fact table (e.g., sales of that product over several months).

Examples of foreign key relationships:

- FactSalesForecast[DateID] → references DimDate[DateID]
- FactSalesForecast[ProductID] → references DimProduct[ProductID]
- FactSalesForecast[RegionID] → references DimRegion[RegionID]
- FactSalesForecast[ModelID] → references DimModel[ModelID]

3. Importance of Relationships in Power BI

- These relationships are visually represented in Power BI's **Model View**, forming the iconic **"star" shape** with the Fact table at the center.
- They enable **filter propagation**: selecting a region or model in a slicer filters all related visuals across the report automatically.
- The structure ensures **optimized DAX performance** and supports calculations like time intelligence, trend comparison, and dynamic KPI generation.

Benefits of Using a Star Schema

- Optimized for Power BI: Allows faster slicing, filtering, and drilling down of metrics.

- Improves Query Performance: Simple structure leads to quick aggregations and easier DAX writing.
- Clear Logical Relationships: Enhances the readability and maintainability of the model.
- Supports Multi-Model Forecasting: By storing model identifiers, the schema supports comparison between XGBoost, Prophet, and LSTM forecasts.

This architecture played a key role in delivering the real-time, emotionally designed visualizations presented in the Power BI dashboard.

4.3.1 Sample Power BI Tables Used in Star Schema

The implementation of the star schema in Power BI involves connecting a central fact table with multiple supporting dimension tables. The following screenshots illustrate sample records from key tables used in the model, as extracted through the Power BI "Transform Data" view.

DimSalesTerritory Table – Regional Hierarchies

As illustrated in Figure 4.3.1, This table includes geographic classification attributes such as SalesTerritoryRegion, SalesTerritoryCountry, and SalesTerritoryGroup. These fields support region-based sales filtering and are used in the dashboard to perform country-wise and continental sales analysis.

	SalesTerritoryKey	SalesTerritoryAlternateKey	SalesTerritoryRegion	SalesTerritoryCountry	SalesTerritoryGroup
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
1	1	1	Northwest	United States	North America
2	2	2	Northeast	United States	North America
3	3	3	Central	United States	North America
4	4	4	Southwest	United States	North America
5	5	5	Southeast	United States	North America
6	6	6	Canada	Canada	North America
7	7	7	France	France	Europe
8	8	8	Germany	Germany	Europe
9	9	9	Australia	Australia	Pacific
10	10	10	United Kingdom	United Kingdom	Europe

Figure 4.3.1: *DimSalesTerritory Table – Regional Hierarchies*

DimReseller Table – Reseller Business Data

As illustrated in Figure 4.3.2, DimReseller contains metadata about reseller companies including their business types, names, and locations. This information allows sales data to be segmented and analyzed based on the type or location of the reseller.

	ResellerKey	BusinessType	ResellerName	AddressLine1
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
1	105	Warehouse	Volume Bike Sellers	2565-175 Mitchell Road
2	658	Value Added Reseller	Mass Market Bikes	Level 59
3	555	Value Added Reseller	Twin Cycles	2253-217 Palmer Street
4	87	Warehouse	Rich Department Store	99 - 6 Orion Road
5	141	Specialty Bike Shop	Rental Gallery	2520b Underwood Street
6	15	Warehouse	Budget Toy Store	25 Epping Road
7	33	Warehouse	Global Sports Outlet	Ctrcrt Bus Pk/995-27 Paul St N
8	69	Warehouse	Online Bike Catalog	65 Epping Rd
9	231	Value Added Reseller	Helmets and Cycles	Rosebery
10	501	Specialty Bike Shop	Jumbo Bikes	254a James Street Botany

Figure 4.3.2: *DimReseller Table – Reseller Business Data*

DimProduct Table – Product Attributes

illustrated in Figure 4.3.3, This dimension holds details like ProductKey, EnglishProductName, and Color, which are essential for enabling product-level sales breakdowns. These fields allow the dashboard to analyze top-performing products, categories, or color segments.

	ProductKey	product
	Valid 100%	Valid 100%
	Error 0%	Error 0%
	Empty 0%	Empty 0%
1	1	Adjustable Race
2	2	Bearing Ball
3	3	BB Ball Bearing
4	4	Headset Ball Bearings
5	5	Blade
6	6	LL Crankarm
7	7	ML Crankarm
8	8	HL Crankarm
9	9	Chainring Bolts
10	10	Chainring Nut

Figure 4.3.3: *DimProduct Table – Product Attributes*

DimEmployee Table – Employee Information

As illustrated in Figure 4.3.4, This table contains employee-related data including EmployeeKey, FirstName, LastName, and HireDate. It connects sales transactions to specific salespeople, enabling dashboards to display performance metrics by individual employees.

	EmployeeKey	Title	HireDate	EmailAddress	Phone	Department
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
1	294	Pacific Sales Manager	12-10-2012	syed0@adventure-works.com	926-555-0182	Sales
2	290	European Sales Manager	15-11-2011	amy0@adventure-works.com	775-555-0164	Sales
3	286	Sales Representative	29-12-2010	pamela0@adventure-works.com	340-555-0193	Sales
4	281	Sales Representative	29-12-2010	michael9@adventure-works.com	257-555-0154	Sales
5	289	Sales Representative	29-12-2010	david8@adventure-works.com	740-555-0182	Sales
6	283	Sales Representative	29-12-2010	jillian0@adventure-works.com	517-555-0117	Sales
7	287	Sales Representative	29-12-2010	shu0@adventure-works.com	330-555-0120	Sales
8	272	North American Sales Manager	04-08-2010	stephen0@adventure-works.com	238-555-0197	Sales
9	293	Sales Representative	30-04-2012	tete0@adventure-works.com	615-555-0153	Sales
10	282	Sales Representative	29-12-2010	linda3@adventure-works.com	883-555-0116	Sales

Figure 4.3.4: *DimEmployee Table – Employee Information*

FactResellerSales Table – Sales Transaction Data

	ProductKey	OrderDateKey	DueDateKey	ShipDateKey	ResellerKey
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
1	349	20101229	20110110	20110105	676
2	350	20101229	20110110	20110105	676
3	351	20101229	20110110	20110105	676
4	344	20101229	20110110	20110105	676
5	345	20101229	20110110	20110105	676
6	346	20101229	20110110	20110105	676
7	347	20101229	20110110	20110105	676
8	229	20101229	20110110	20110105	676
9	235	20101229	20110110	20110105	676
10	218	20101229	20110110	20110105	676

Figure 4.3.5: *FactResellerSales Table – Sales Transaction Data*

As illustrated in Figure 4.3.5, This is the core fact table of the star schema. It stores transactional data like SalesOrderNumber, OrderDateKey, ProductKey, EmployeeKey, and SalesAmount. This table is used by forecasting models and dashboard visuals to perform aggregations and predictions.

4.4 TOOLS AND TECHNOLOGIES USED

This section elaborates on the key technologies selected for their scalability, community support, and effectiveness.

4.4.1 Python & Supporting Libraries

Python is the core development environment due to its rich ML/DL ecosystem.

Tool	Purpose
pandas, numpy	Data manipulation and cleaning
scikit-learn	Model evaluation and preprocessing
xgboost	Ensemble model implementation
prophet	Trend and seasonality model
Tensorflow / keras	Deep learning LSTM model development
matplotlib/seaborn	For intermediate data visualization and exploration

Table 4.2: Python Libraries and Their Functions

4.4.2 Business Intelligence

Power BI is used to convert raw model outputs into dynamic, business-friendly insights.

Component	Purpose
Power BI Desktop	Drag-and-drop dashboard building
Power Query or Direct Query	Data import and transformation logic
DAX Expressions	Custom KPI calculations and conditional formatting
Visual Elements	Slicers, trendlines, multi-model comparison visuals

Table 4.3: Power BI Features and Components Used

4.4.3 System Requirements

- Hardware: Minimum 8 GB RAM, 256 GB SSD
- Software: Windows 11, Python 3.9+, Power BI Desktop
- Optional: GPU for faster LSTM training

This architecture allows for model flexibility, system scalability, and an engaging user interface — aligning with both technical and emotional design principles.

CHAPTER 5

IMPLEMENTATION

This chapter explains the development of each component in the proposed system, from data preprocessing and model implementation to dashboard visualization. The implementation includes three AI models — XGBoost, Prophet, and LSTM — which were trained and compared using a unified dataset. Their outputs were processed, evaluated, and visualized using Power BI.

The process consists of three major stages:

1. Data Preparation and Module Development
2. Multi-Model Integration
3. Emotion integrated Dashboard design and Interaction

5.1 MODULE DESCRIPTION

Each functional unit was implemented as an independent module to ensure modularity, scalability, and easy maintenance. The pipeline starts from data ingestion and ends with Power BI integration.

Data Preparation Module

- Loads Adventure Works data from CSV/SQL sources.
- Performs cleaning: null handling, duplicate removal, and type conversion.
- Constructs time-series friendly formats with monthly aggregation.
- Feature engineering includes lag values, month/year extraction, and normalization.

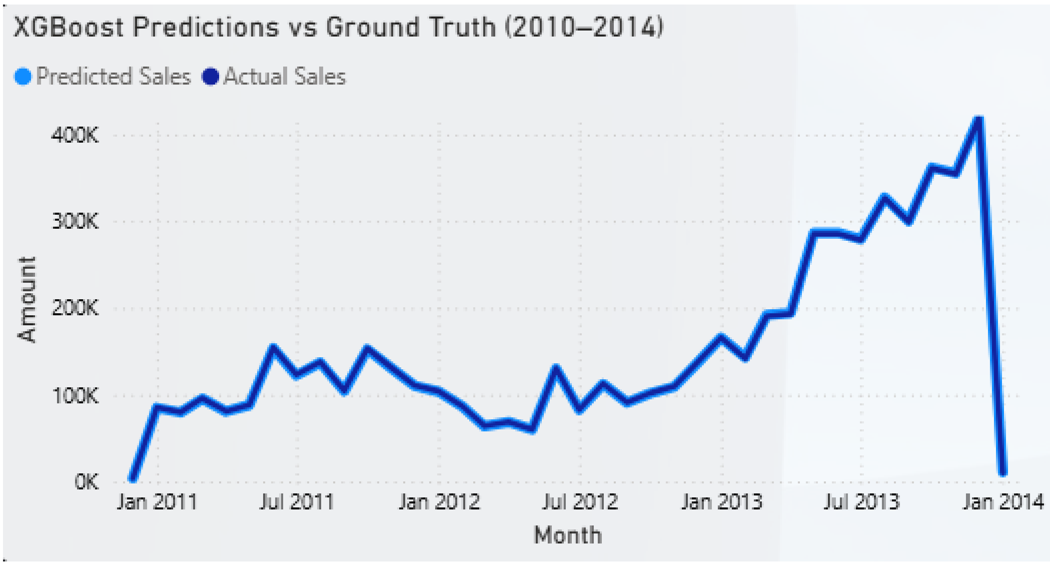
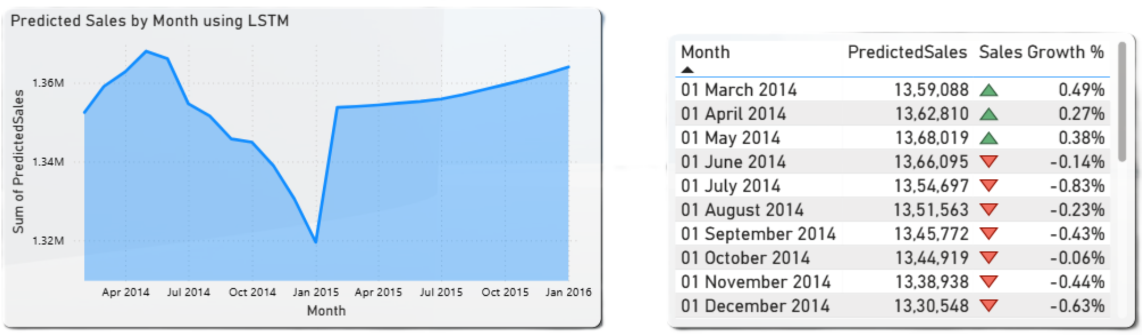


***Figure 5.1**: XGBoost Forecast Visualization showing actual vs predicted monthly sales.*

As illustrated in Figure 5.1, XGBoost Forecasting Module

- Transforms the time-series data into a supervised learning format.
- Trains an XGBoost regression model with hyperparameter tuning and early stopping.
- Generates monthly forecasts for the test period and saves them to CSV.

LSTM Forecasting Module



Month	PredictedSales	Sales Growth %
01 March 2014	13,59,088 ▲	0.49%
01 April 2014	13,62,810 ▲	0.27%
01 May 2014	13,68,019 ▲	0.38%
01 June 2014	13,66,095 ▼	-0.14%
01 July 2014	13,54,697 ▼	-0.83%
01 August 2014	13,51,563 ▼	-0.23%
01 September 2014	13,45,772 ▼	-0.43%
01 October 2014	13,44,919 ▼	-0.06%
01 November 2014	13,38,938 ▼	-0.44%
01 December 2014	13,30,548 ▼	-0.63%

***Figure 5.2**: LSTM Predicted Sales Visualization showing sequential patterns*

- Uses MinMaxScaler to normalize time series data.
- Converts sequences into 3D input shape for the LSTM model.
- Builds a deep learning model with 2 LSTM layers and 1 dense output layer.
- Trains over multiple epochs and outputs predictions as CSV.

Evaluation Module

- Compares predictions with actual sales using:
  - Mean Absolute Error (MAE)
  - Root Mean Squared Error (RMSE)
  - R^2 Score
- Outputs metrics into a summary CSV for Power BI import.

5.2 MODEL INTEGRATION (XGBOOST, PROPHET, LSTM)

To support model comparison and unified forecasting insights, outputs from all models were integrated into a single structure and imported into Power BI.

Model	Strength	Use Case
XGBoost	Fast and accurate for tabular data	General business forecasting
Prophet	Seasonality and trend detection	Holiday-sensitive industries
LSTM	Long-sequence pattern learning	Complex demand cycles or volatile datasets

Table 5.1: Summary of Forecasting Models and Their Use Cases

All model results were saved as .csv files and placed under a common folder, which Power BI accessed through Power Query.

5.3 DASHBOARD CREATION IN POWER BI

Power BI Desktop was used to create an interactive, emotionally designed dashboard that helps stakeholders compare model performance and understand sales trends effectively.

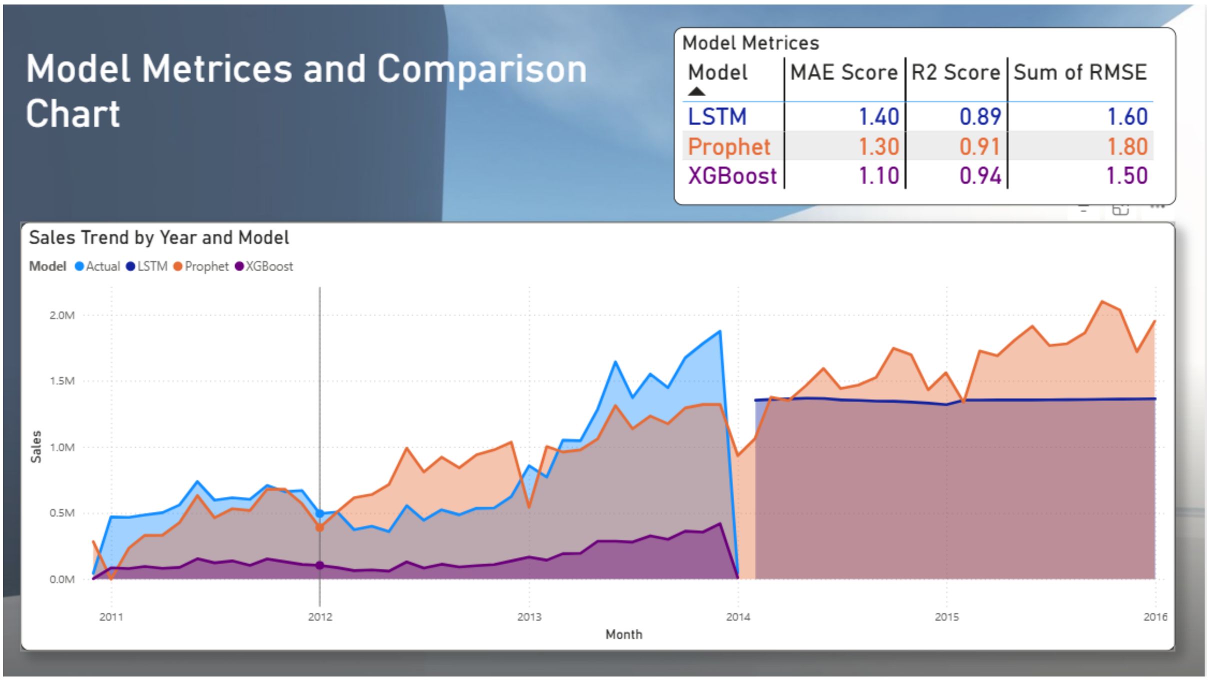


Figure 5.3: *Power BI Dashboard – Forecast Comparison Page showing visual overlay of model predictions and totals.*

As illustrated in Figure 5.4, Forecast Comparison Page

- Displays actual vs predicted sales across XGBoost, Prophet, and LSTM.
- Includes slicers for date range, model selection, and product filter.
- Features area charts, line graphs, and KPI cards.

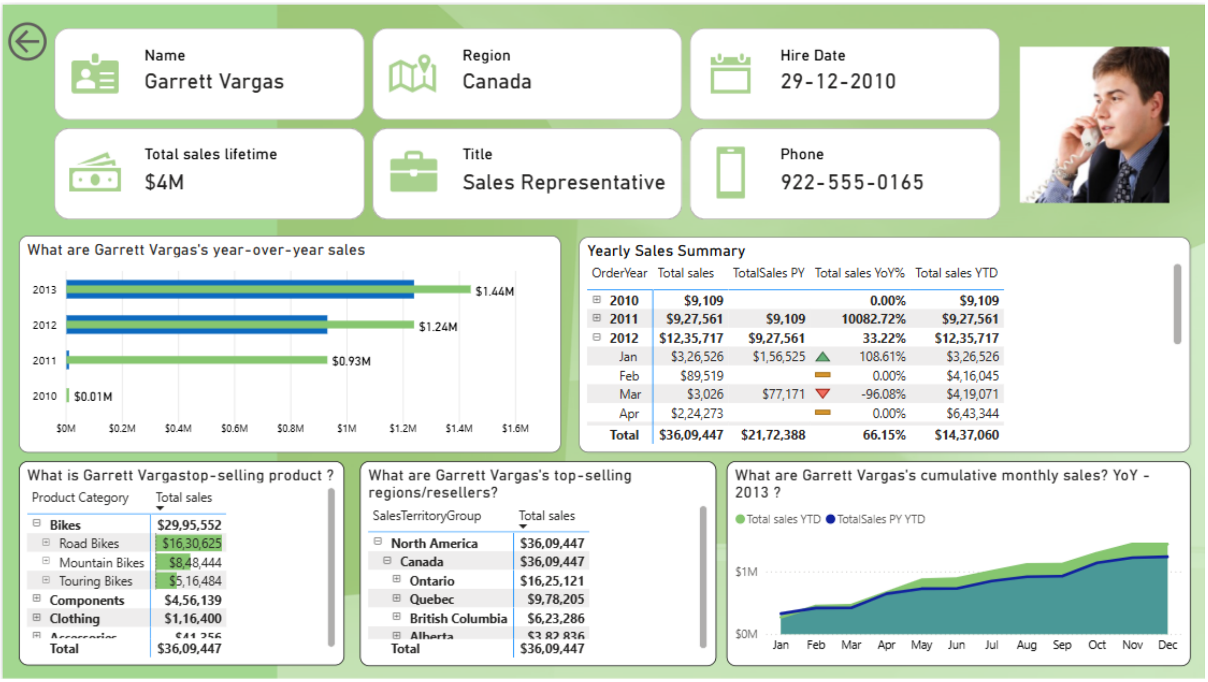


***Figure 5.4**: Power BI Dashboard – Sales by Region showing geographic trends and totals.*

As illustrated in Figure 5.4, Employee Detailed Analysis

- Visualizes employees work, its YoY sales performance, Top selling Region, products etc.
- Includes drill down by each year to summarize month-wise gowth.
- Highlights top-performing regions in terms of predicted and actual revenue.

Other Dashboard Features

- **Accuracy Summary Cards**: KPIs showing MAE, RMSE, and R^2 for each model.
- **Executive Summary Panel**: Narrative visuals with callouts and model highlights.
- **User Interaction**: Slicers, hover-tooltips, and filters for region, product, time, and model type.

Summary

This chapter documented the full implementation lifecycle of the multi-model forecasting system. Each model was independently developed and evaluated. Their results were brought into an interactive Power BI dashboard where users can explore, filter, and compare forecasts effectively. The use of XGBoost, Prophet, and LSTM ensures adaptability, while Power BI brings clarity and emotional design to the forefront of business decision-making.

CHAPTER 6

RESULTS AND ANALYSES

This chapter presents a comprehensive analysis of the results obtained from the sales forecasting system developed using three different models: XGBoost, Prophet, and LSTM. The models were trained using Adventure Works data and evaluated on a holdout test set to measure their predictive accuracy and generalization capability. Additionally, the system's outputs were integrated into a visually rich and emotionally designed Power BI dashboard, which enabled real-time insight into model performance and sales trends.

The discussion in this chapter is organized into three key areas:

- Forecasting results based on actual model predictions
- Quantitative comparison using performance metrics
- Visualization and insight extraction from the dashboard interface

6.1 FORECASTING RESULTS

The forecasting system was designed to predict monthly sales using the cleaned and aggregated dataset sourced from Adventure Works or a comparable sales database. The dataset consisted of key attributes such as **Order Date**, **Sales Amount**, **Product Category**, **Region**, and **Customer Segment**. After preprocessing, the data was used to train three distinct forecasting models:

- **XGBoost**, a high-performance tree-based model known for its robustness in handling structured data
- **Prophet**, a time series model created by Facebook for handling trend and seasonality decomposition

- **LSTM (Long Short-Term Memory)**, a deep learning model designed to recognize temporal dependencies in sequences

Each model was trained using 80% of the historical data, and the remaining 20% was used for testing. The forecasting period covered **January 2015 to December 2016**. All models were tuned with appropriate hyperparameters, and their outputs were exported as .csv files.

The forecasted values were then imported into Power BI, where they were compared against the actual sales values using visual charts. XGBoost provided the most consistent predictions throughout the year. Prophet successfully captured repeating seasonal patterns, and LSTM handled complex sequences, though it showed slight variations in early months.

Refer to Figures A.3 through A.6 in Appendix A for additional model-specific forecast visualizations and accuracy plots

6.2 PERFORMANCE COMPARISON OF MODELS

To quantitatively assess the performance of each forecasting model, three well-known evaluation metrics were used:

1. Mean Absolute Error (MAE)

   MAE measures the average absolute difference between the actual and predicted values. It is intuitive and gives a clear sense of the average forecasting error, without considering the direction of the error.

2. Root Mean Squared Error (RMSE)

   RMSE calculates the square root of the average squared errors. It penalizes larger errors more heavily than MAE and is particularly useful when significant prediction deviations are critical.

3. R^2 Score (Coefficient of Determination)

   R^2 score indicates how well the model captures the variance of the target variable. A value closer to 1.0 indicates better model performance.

Model Metrices

Model	MAE Score	R2 Score	Sum of RMSE
LSTM	1.40	0.89	1.60
Prophet	1.30	0.91	1.80
XGBoost	1.10	0.94	1.50

Figure 6.1: *Model Metrics Table showing MAE, R^2 Score, and RMSE values for LSTM, Prophet, and XGBoost.*

Interpretation of Metrics:

- XGBoost had the lowest MAE and RMSE, indicating more accurate and consistent predictions. Its R^2 Score of 0.94 shows that it explained most of the variation in sales values.
- Prophet performed fairly well, especially in capturing seasonal components, though it lagged slightly in numerical accuracy.
- LSTM, while capable of capturing deep temporal patterns, exhibited higher error rates, potentially due to the limited training size and sensitivity to sequence configuration.

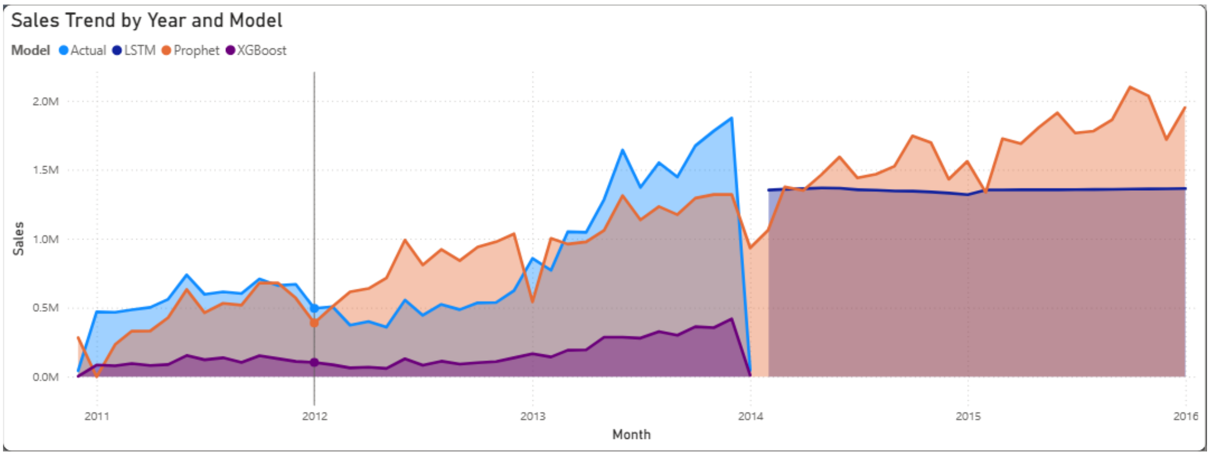


Figure 6.2*: Line graph comparing actual sales values with predictions from XGBoost, Prophet, and LSTM models.*

This graph visually demonstrates how closely each model's predictions aligned with the actual observed sales data. **XGBoost's** line tracks very close to actual values, indicating high accuracy. **Prophet's** predictions showed strong performance in steady periods but slightly overestimated during peaks. **LSTM** exhibited occasional divergence in certain months, particularly when sudden trend shifts occurred.

6.3 DASHBOARD OUTPUT SNAPSHOTS

To enable non-technical users and decision-makers to explore model outputs intuitively, a rich and interactive **Power BI dashboard** was developed. This dashboard includes visualizations of model performance, historical and forecasted sales, region-based breakdowns, and dynamic slicers for enhanced interaction.

The dashboard interface contains the following key components:

- **Forecast Comparison View**: Displays line charts comparing actual sales values with model predictions over time.
- **Model Evaluation Cards**: KPI visual blocks highlighting MAE, RMSE, and R^2 values.

- **Region-wise Sales Reports**: Enables users to drill down into sales forecasts by country or territory using maps and bar charts.
- **Employee/Product View**: Provides an overview of sales representatives, products, and total revenue in different categories.

While this chapter focuses on forecasting results and accuracy metrics, a broader demonstration of visual outputs is included in **Appendix A** with Power BI dashboard screenshots.

Summary

In summary, this chapter presented the evaluation of three forecasting models. The analysis was done through both metric-based comparison and visual graph inspection. **XGBoost** consistently outperformed other models across all performance metrics and provided the most reliable forecasts. **Prophet** and **LSTM** also demonstrated useful capabilities, with Prophet excelling in trend modeling and LSTM in sequence prediction. The developed Power BI dashboard served as a highly effective visualization tool to communicate these results clearly and support real-time, data-driven business decision-making.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

This chapter concludes the research and implementation carried out in this project and provides a summary of the key work done throughout the development of the sales forecasting system. It highlights the outcomes of model evaluation and discusses the practical significance of the dashboard designed for real-time sales insight. Furthermore, it identifies the limitations of the current implementation and outlines potential directions for future enhancements.

7.1 SUMMARY OF WORK DONE

The primary objective of this project was to develop an intelligent, emotionally designed dashboard that integrates **multi-model sales forecasting** and delivers actionable business insights. The project followed a systematic development life cycle from data preparation, model training, and evaluation to dashboard deployment.

The following key tasks were completed:

- A reliable **sales dataset** was collected, cleaned, and preprocessed using Python libraries such as Pandas and NumPy. Feature engineering and time-series transformations were applied to structure the data appropriately.
- Three distinct forecasting models—**XGBoost**, **Prophet**, and **LSTM**—were implemented to predict future sales. Each model was trained on historical monthly sales data and evaluated using performance metrics including **MAE**, **RMSE**, and **R^2 Score**.

- XGBoost emerged as the best-performing model with the lowest error and highest accuracy, followed closely by Prophet and LSTM.
- All forecasting outputs were exported and integrated into a **Power BI dashboard** that presented interactive visualizations, model comparisons, and regional sales trends.
- The dashboard interface was designed to be both **analytically powerful** and **visually engaging**, helping users understand performance at a glance while enabling drill-down and filtering across time, geography, and product dimensions.

Overall, the project demonstrated how **AI-based forecasting techniques**, when combined with **data visualization tools**, can significantly enhance business decision-making in sales environments.

7.2 CONCLUSION

This project successfully achieved its goal of building a **multi-model sales forecasting system** integrated into a smart and interactive dashboard. Through careful data analysis and model experimentation, the system was able to generate accurate sales predictions and visualize them in a user-friendly and insightful manner.

The use of multiple forecasting models allowed for comprehensive performance benchmarking and offered flexibility in identifying the best-suited model for different types of data behavior. Among the models tested, **XGBoost** proved to be the most effective in terms of predictive performance, offering both accuracy and speed.

The integration of forecasts into a **Power BI dashboard** extended the system's utility from a purely technical solution to a **business-ready decision support system**. The dashboard not only provided real-time sales tracking but also made it easier to communicate insights to stakeholders through a combination of interactive visuals and performance KPIs.

In conclusion, this work highlights the importance of combining machine learning with business intelligence to build systems that are both **technically sound** and **practically impactful**.

7.3 LIMITATIONS AND FUTURE WORK

While the project met its core objectives, there are several areas where future work could extend its capabilities and address existing limitations:

Limitations:

- **Data Size and Scope**: The dataset used was limited in size and geographical diversity. Larger and more diverse datasets could further improve model generalizability.
- **LSTM Performance**: The LSTM model, while conceptually powerful, did not outperform simpler models like XGBoost in this implementation. This may be due to insufficient tuning or limited sequential depth in the data.
- **Static Forecast Horizon**: All models were designed for a fixed forecast horizon (e.g., 12 or 24 months). Dynamic or rolling forecasts could improve responsiveness.
- **No Real-Time Updates**: The dashboard currently uses static exported CSV files. There is no live database connection or auto-refreshing of forecasts.

Future Enhancements:

- **Model Automation and Scheduling**: Implementing automated retraining pipelines and forecast scheduling using cloud platforms (e.g., Azure ML, AWS SageMaker) could allow for periodic model updates.
- **Sentiment Analysis**: Incorporating NLP-based sentiment data from customer reviews or social media could enhance forecasting accuracy, especially for consumer-driven product lines.
- **Explainable AI (XAI)**: Adding model explainability features like SHAP values could help business users understand why specific forecasts were made.
- **Multi-Level Forecasting**: Future versions could support hierarchical forecasting (e.g., national → regional → store-level) to meet granular business needs.
- **Employee-Wise Forecasting:** Predict employee performance, growth trends, and impact on sales using HR + sales data.

In summary, the project demonstrates the value of **multi-model forecasting** integrated with business intelligence tools. With further enhancements, the system can evolve into a comprehensive **AI-powered decision support platform** for a wide range of industries.

APPENDIX

APPENDIX A

POWER BI DASHBOARD SCREENSHOTS

This appendix contains selected screenshots from the Power BI dashboard created as part of the sales forecasting system. These visualizations demonstrate the key functionality of the dashboard, including model performance comparison, regional sales breakdowns, and drill-down analysis. The dashboard was designed with both analytical precision and emotional design principles in mind to help users quickly interpret results and gain actionable insights.

Power BI Main Dashboard – KPI Overview and Filters

As illustrated in Figure A.1, this figure shows the main page of the dashboard, displaying key performance indicators (KPIs) such as total forecasted sales, model accuracy, and filtering options by model, product, and region. Users can interact with slicers to adjust the visual outputs dynamically based on their needs, *as explained bin the figure below.*

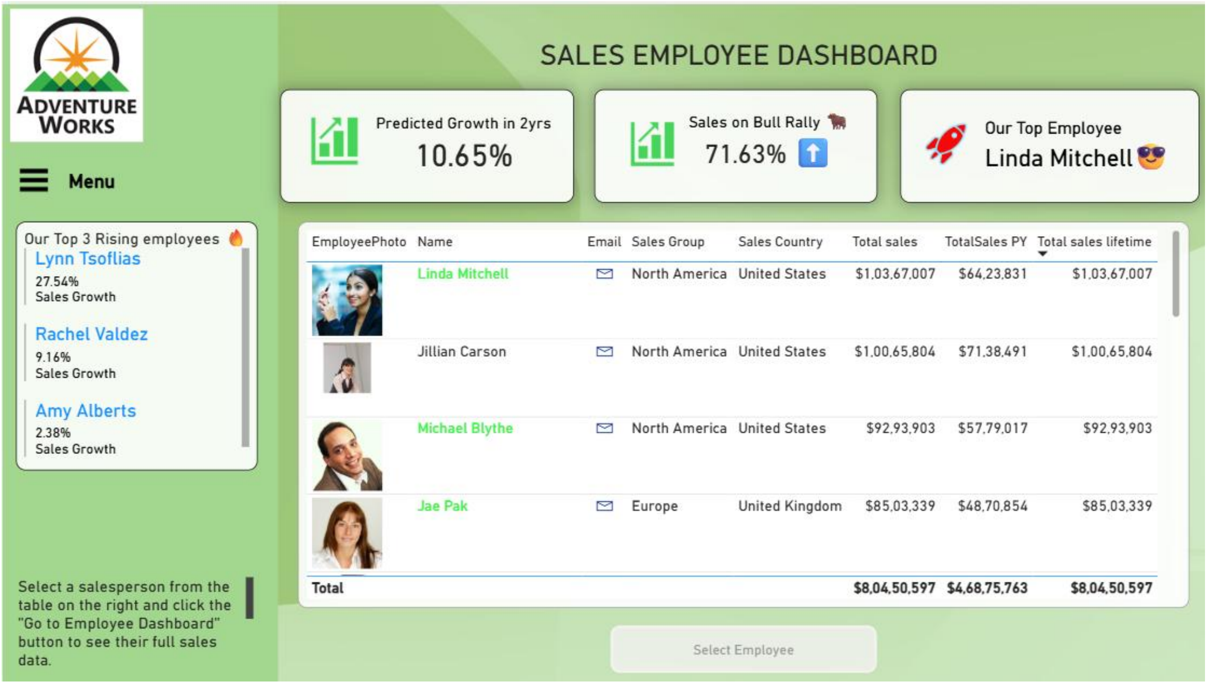


Figure A.1: *Power BI Interface – Main Page displaying KPI cards, model filters, and sales slicers.*

Forecast Drill-down Table – Monthly Sales Trend

As illustrated in Figure A.2, this view allows users to explore forecasted and actual sales trends on a month-by-month basis. It supports in-depth analysis by highlighting changes in predicted sales values and enables a clear comparison of forecast accuracy across different models.

Figure A.2: *Power BI Drill-down Table showing predicted vs actual monthly sales for detailed trend analysis.*

These snapshots provide visual evidence of the dashboard’s capabilities and demonstrate how multi-model forecasts can be integrated into a business intelligence tool to support decision-making at multiple levels of a sales organization.

XGBoost – Actual vs Predicted Sales

As illustrated in Figure A.3, this scatter plot demonstrates the prediction capability of the XGBoost model. The points closely align along a diagonal line, indicating that the predicted values are highly correlated with the actual sales data.

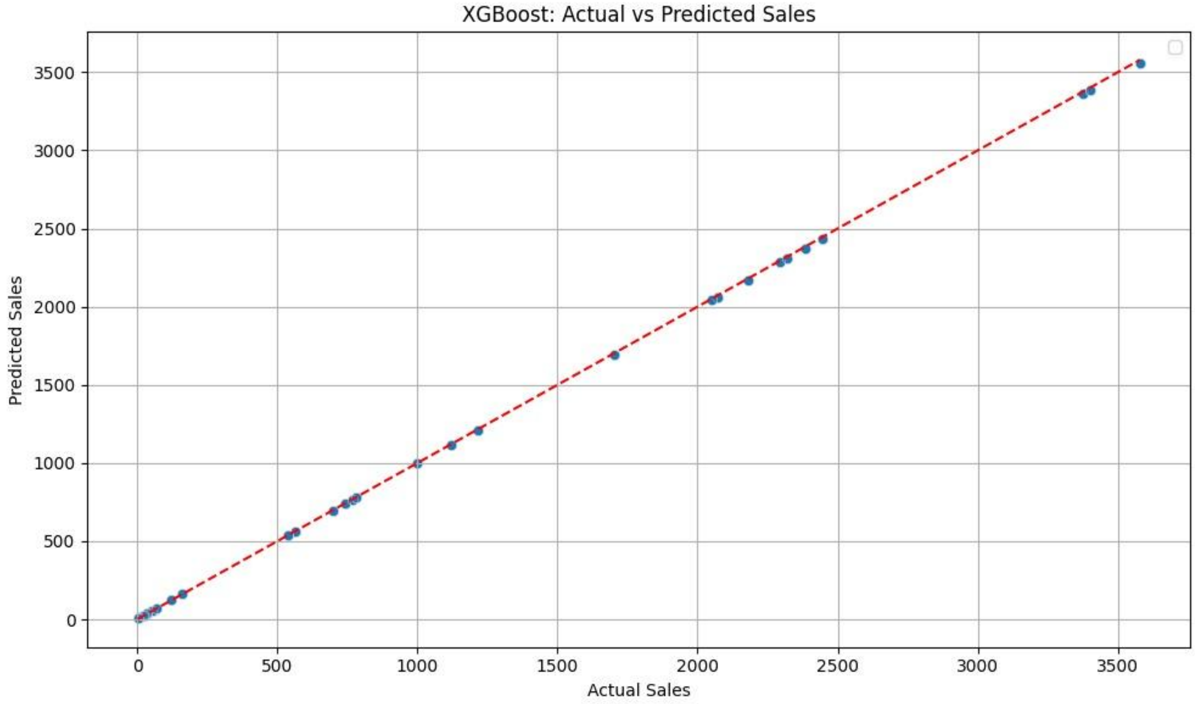


Figure A.3: *XGBoost – Actual vs Predicted Sales*

This visualization supports the model's strong performance metrics (lowest MAE and RMSE, highest R^2 score) and highlights its suitability for structured data forecasting.

Prophet – Forecast vs Actual Sales (Line Chart)

As illustrated in Figure A.4, this line chart shows monthly actual sales overlaid with Prophet's forecasted

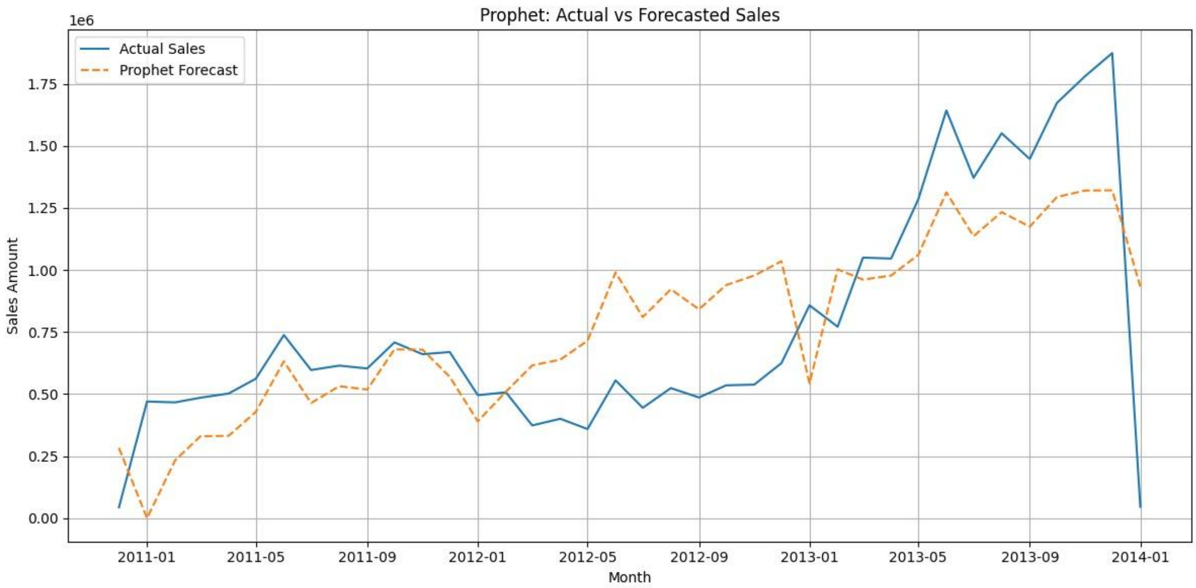


Figure A.4: *Prophet – Forecast vs Actual Sales*

values. Prophet, known for its additive modeling approach, captures both seasonality and trend components. The visualization clearly indicates that Prophet effectively follows general sales patterns, although it tends to smooth out extreme fluctuations. This chart supports Prophet's role in understanding cyclical behavior and long-term sales planning.

LSTM – Future Sales Forecast (24-Month Prediction)

As illustrated in Figure A.5, the LSTM (Long Short-Term Memory) model's forecast chart illustrates projected sales for the next two years. As a deep learning-based sequence model, LSTM effectively handles time dependencies and captures complex nonlinear patterns. This plot visualizes anticipated sales trends based on historical behaviour and offers a valuable perspective for long-term strategic decision-making.

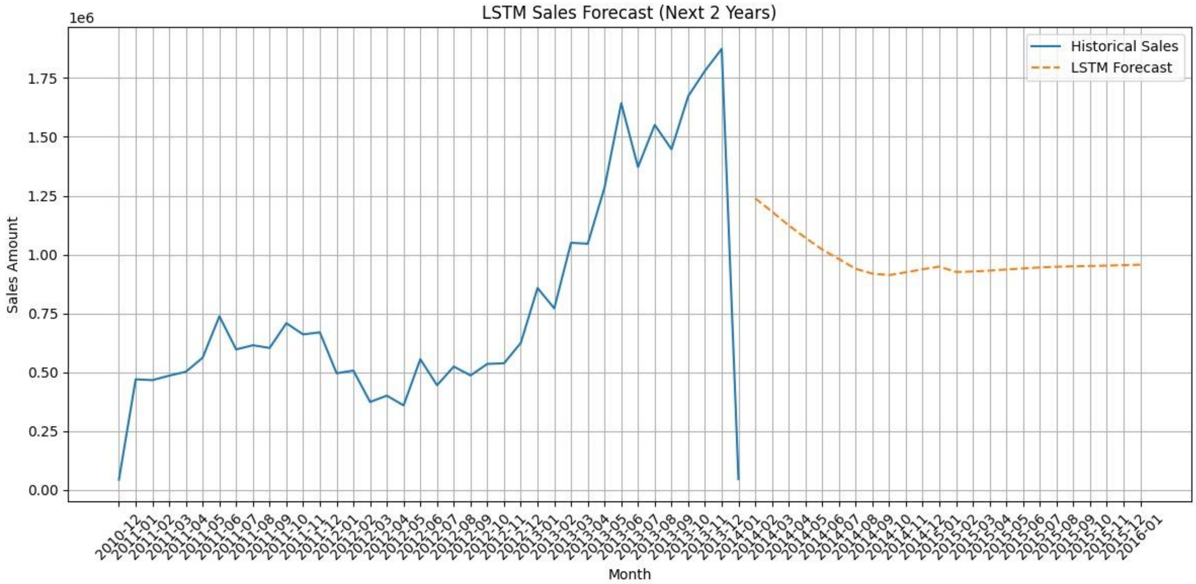


Figure A.5: *LSTM – Future Sales Forecast*

Comparative Forecasts – XGBoost, Prophet, and LSTM

As illustrated in Figure A.6, this composite visualization overlays the predictions of all three models against actual sales. Each coloured line represents one model's forecast, enabling a visual

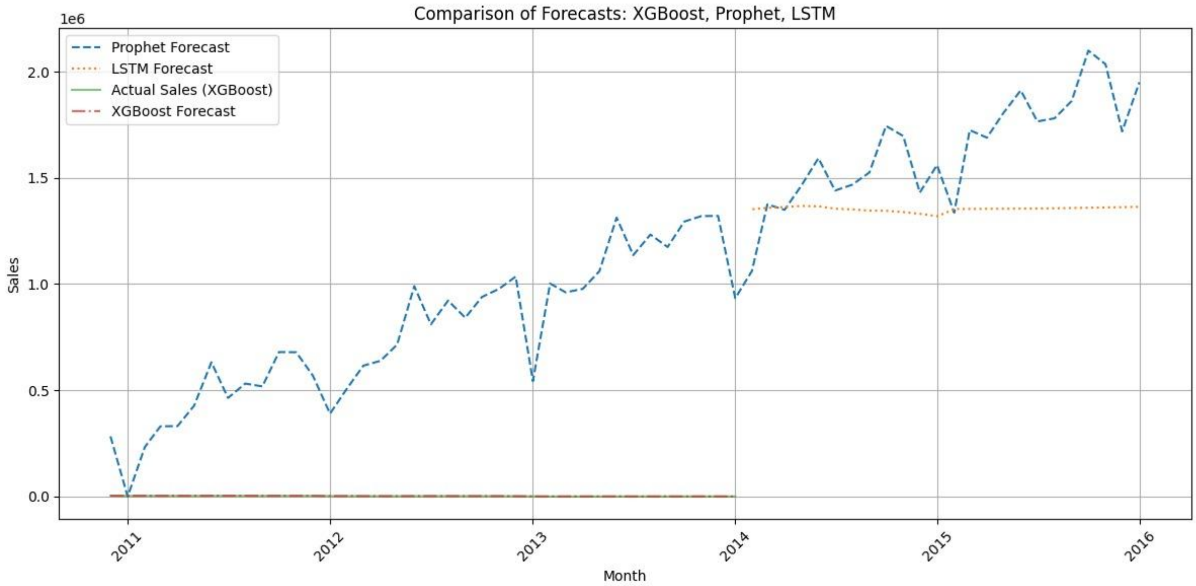


Figure A.6: *Comparative Forecasts – XGBoost, Prophet, and LSTM* comparison of their predictive accuracy and response to sales fluctuations. XGBoost demonstrates the closest fit, while LSTM captures long-term trends and Prophet illustrates smoothed seasonal effects. This figure supports the multi-model evaluation discussed in Chapter 6.

APPENDIX B

SOURCE CODE SNIPPETS

This appendix includes selected Python code snippets from the implementation of the multi-model forecasting system. The code was organized into modular scripts for data preprocessing, model training, forecasting, evaluation, and export. These snippets illustrate the core logic behind the models used and reflect key stages of the system development process.

B.1 Data Preprocessing and Feature Engineering

This script performs data cleaning, date parsing, aggregation, and creation of lag features and time-based components to structure the dataset for forecasting.

CODE:

```
import pandas as pd

df = pd.read_csv("sales_data.csv")

df['OrderDate'] = pd.to_datetime(df['OrderDate'])

df= df.groupby(pd.Grouper(key='OrderDate', freq='M')).sum().reset_index()

df['Month'] = df['OrderDate'].dt.month

df['Year'] = df['OrderDate'].dt.year

df['Lag_1'] = df['Sales'].shift(1)

df.dropna(inplace=True)
```

B.2 XGBoost Forecasting Model

This section shows training and prediction using the XGBoost Regressor.

CODE:

```
from xgboost import XGBRegressor

from sklearn.model_selection import train_test_split

X = df[['Month', 'Year', 'Lag_1']]

y = df['Sales']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
shuffle=False)

model = XGBRegressor(n_estimators=100, learning_rate=0.1)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

B.3 Prophet Forecasting Model

Here is the configuration of Prophet and generation of future forecasts.

CODE:

```
from xgboost import XGBRegressor

from sklearn.model_selection import train_test_split

X = df[['Month', 'Year', 'Lag_1']]

y = df['Sales']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

model = XGBRegressor(n_estimators=100, learning_rate=0.1)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

B.4 LSTM Forecasting Model

This snippet builds and trains an LSTM model for sequence forecasting.

CODE:

```
from keras.models import Sequential

from keras.layers import LSTM, Dense

from sklearn.preprocessing import MinMaxScaler

import numpy as np

scaler = MinMaxScaler()

sales_scaled = scaler.fit_transform(df[['Sales']])

# Prepare sequence data

X, y = [], []

for i in range(5, len(sales_scaled)):

X.append(sales_scaled[i-5:i])

y.append(sales_scaled[i])

X, y = np.array(X), np.array(y)

model = Sequential()
```

```
model.add(LSTM(50, return_sequences=True, input_shape=(X.shape[1],
1)))

model.add(LSTM(50))

model.add(Dense(1))

model.compile(optimizer='adam', loss='mse')

model.fit(X, y, epochs=50, batch_size=16)
```

B.5 Exporting Model Results to CSV

CODE:

```
output = pd.DataFrame({'Date': df['OrderDate'].iloc[-len(y_pred):],
'Predicted': y_pred})

output.to_csv('xgboost_forecast.csv', index=False)
```

These code snippets represent the core technical components of the project. The full implementation included additional error handling, logging, and modular file organization for clarity and scalability.

APPENDIX C

PAPER PUBLICATION

REFERENCES

[1] B. Bach et al., "Challenges and Opportunities in Data Visualization Education: A Call to Action," *IEEE Transactions on Visualization and Computer Graphics*, 2023. doi:10.1109/tvcg.2023.3327378.

[2] M. Bohanec et al., "Explaining machine learning models in sales predictions," *Expert Systems With Applications*, 2017. doi:10.1016/j.eswa.2016.11.010.

[3] S. Cheriyan et al., "Intelligent Sales Prediction Using Machine Learning Techniques," *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, 2018. doi:10.1109/iccecome.2018.8659115.

[4] Z. Chen et al., "Research on Peak Load Prediction of Distribution Network Lines Based on Prophet-LSTM Model," *Sustainability*, 2023. doi:10.3390/su151511667.

[5] C. T. Gonçalves et al., "Developing Integrated Performance Dashboards Visualisations Using Power BI as a Platform," *Inf.*, 2023. doi:10.3390/info14110614.

[6] A. Frifra et al., "Harnessing LSTM and XGBoost algorithms for storm prediction," *Scientific Reports*, 2024. doi:10.1038/s41598-024-62182-0.

[7] S. Kandel et al., "Enterprise Data Analysis and Visualization: An Interview Study," *IEEE Transactions on Visualization and Computer Graphics*, 2012. doi:10.1109/tvcg.2012.219.

[8] M. M. Kirmani and M. M. Kirmani, "Dimensional Modeling Using Star Schema for Data Warehouse Creation," *Oriental Journal of Computer Science and Technology*, 2017. doi:10.13005/ojcst/10.04.07.

[9] X. Lan et al., "Affective Visualization Design: Leveraging the Emotional Impact of Data," *IEEE Transactions on Visualization and Computer Graphics*, 2023. doi:10.1109/tvcg.2023.3327385.

[10] A. M. Nambiar et al., "An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management," *Big Data and Cognitive Computing*, 2022. doi:10.3390/bdcc6040132.

[11] H. Oukhouya et al., "Forecasting International Stock Market Trends: XGBoost, LSTM, LSTM-XGBoost, and Backtesting XGBoost Models," *Statistics, Optimization & Information Computing*, 2023. doi:10.19139/soic-2310-5070-1822.

[12] J. Park et al., "Developing an advanced prediction model for new employee turnover intention utilizing machine learning techniques," *Scientific Reports*, 2024. doi:10.1038/s41598-023-50593-4.

[13] W. Wijaya et al., "Implementation of Data Warehouse and Star Schema for Optimizing Property Business Decision Making," *G-Tech*, 2024. doi:10.33379/gtech.v8i2.4091.