IBM Data Science Project:

Sales Forecasting and Optimization

Presented By:

- Abdallah Adel Abdallah 21063654
- Abdelrahman Adel Abdelkader 21057228
- Abdelrahman Badawy Ali 21074236
- Asmaa Muhammad Abdelhamid 21095177
- Maryam Taha Abdelaty 21067260
- Muhammed Ahmed Abdelmegeed 21063106

Project 2: Sales Forecasting and Optimization

Project Overview:

The Sales Forecasting and Optimization project aims to predict future sales for a retail or e-commerce business by using historical sales data. The project involves data collection, cleaning, exploration, time-series forecasting model development, optimization, and deployment. The end goal is to have a model that can generate accurate sales predictions to help businesses optimize inventory, marketing, and sales strategies.

Project Guidelines

1. Project Planning & Management

a) Project Proposal

- Overview:
  The project aims to develop a sales forecasting and optimization system for a retail business using historical sales data. The system will predict future sales, optimize inventory levels, and provide actionable insights to improve business performance.
- Objective:
  - Improve sales forecasting by building an AI model to predict sales trends.
  - Improve pricing strategy by suggesting the best prices to generate maximum revenue.
  - Improve inventory management to reduce costs and avoid stockouts.
  - Provide actionable demand and marketing planning insights.
- Scope:
  Data collection and preprocessing from historical sales records. Development of time-series forecasting models at multiple granularity levels (store-level, category-level, product-level) incorporating external factors visible such as weather conditions, promotional periods, and seasonality. Creation of a dashboard for visualization and reporting.

b) Project Plan

- Week 1-2: Data Collection, Exploration, and Preprocessing
  - Milestones: Completion of data collection and preprocessing.
  - Deliverables: Cleaned dataset, data exploration report.
  - Resource Allocation: Data Analyst, Data Engineer.

- Week 3-4: Data Analysis and Visualization
  - Milestones: Completion of data analysis and visualization.
  - Deliverables: Analyzed data insights, visualizations.
  - Resource Allocation: Data Analyst, Data Scientist.

- Week 5-6: Model Development and Training
  - Milestones: Trained model.
  - Deliverables: Trained model, model performance metrics.

- Resource Allocation: Data Scientist, Machine Learning Engineer.

- Week 7: Model Evaluation and Optimization
  - Milestones: Optimized model.
  - Deliverables: Optimized model, evaluation report.
  - Resource Allocation: Data Scientist, Machine Learning Engineer.

- Week 8: Dashboard Development and Deployment
  - Milestones: Deployed dashboard.
  - Deliverables: Deployed dashboard, user guide.
  - Resource Allocation: Frontend Developer, Backend Developer.

- Week 9: Testing, Final Documentation, and Presentation
  - Milestones: Final project completion.
  - Deliverables: Final report, presentation slides.
  - Resource Allocation: Project Manager, Data Analyst.

c) Task Assignment & Roles – Defined responsibilities for team members.

- Member 1: Manage timelines, deliverables, and communication between team members, risk assessment and help them.
- Member 2: Collect and explore historical sales data and preprocess it for analysis and model building.
- Member 3: Clean and preprocess the data further and visualize the relationships in the data.
- Member 4: Build models, incorporating external variables like weather conditions and promotional periods evident in the dataset.
- Member 6: Tests the system against historical data to ensure forecast accuracy and system reliability.
- All member: Dashboard & visualization implementation and Final Presentation.

d) Risk Assessment & Mitigation Plan

- Data Quality Issues: Missing or inconsistent data.
  - Mitigation: Perform thorough data cleaning and anomaly detection.
- Model Overfitting: Poor generalization to new data.
  - Mitigation: Use cross-validation and regularization techniques.
- Deployment Challenges: Issues with API or dashboard deployment.
  - Mitigation: Test deployment in a staging environment before going live.
- Stakeholder Misalignment: Misunderstanding of project goals.
  - Mitigation: Regularly communicate with stakeholders and gather feedback.

e) KPIs (Key Performance Indicators)

- Forecast Accuracy Metrics: Implement MAPE (Mean Absolute Percentage Error) or similar metrics to quantify prediction accuracy.
  OR Forecast Accuracy: Measures how close predictions are to actual sales.
- System Uptime: Availability of forecasting service and Response time of predictions

- Inventory Efficiency: Reduction in stockouts & overstock cases
- Revenue Growth: Impact of optimized pricing strategies on sales.
- User Adoption: Number of active users (e.g., store managers, inventory planners), and User satisfaction score.

2. Literature Review

- Feedback & Evaluation – Lecturer's assessment of the project.
- Suggested Improvements – Areas where the project can be enhanced.
- Final Grading Criteria – Breakdown of marks based on documentation, implementation, testing, and presentation.

3. Requirements Gathering

a) Stakeholder Analysis – Identifying key stakeholders and their needs.

- Business Owners: Need accurate sales forecasts to plan inventory and staffing. OR Need revenue growth insights.
- Supply Chain Team: Requires understanding of units ordered, competitor pricing, and seasonality trends for restocking.
- Inventory Managers: Need accurate forecasts to optimize stock.
- Marketing Department: Need insights on promotions and pricing strategies.
- Finance Department: Requires revenue forecasts for financial planning.
- IT Department: Needs data design reliability, and integration with existing ERP approaches because he is responsible for system integration and maintenance.

b) User Stories & Use Cases

User Storys:

- Business Owners: I want to visualize predicted sales patterns for my store during promotional periods (using the Holiday/Promotion indicator)
  - Use Case: to optimize staffing and plan inventory levels.
- Inventory Manager: I want to see projected inventory needs for the next 30 days by store and product category and receive alerts for low stock levels
  - Use Case: optimize stock levels, minimize carrying costs, and reorder products when inventory levels fall below a threshold.
- Marketing manager: I want to analyze the impact of promotions on sales so that I can optimize marketing campaigns.
  - Use Case: The system provides insights into sales trends during promotions.
- Finance Analyst: I want to forecast revenue by region and category
  - Use Case: To support budget planning processes.

c) Functional Requirements – List of features and functionalities.

- Inventory Management: Track current stock, units sold, and forecasted demand.
- Sales Tracking: Monitor sales trends by store, region, and category.
- Pricing Optimization: Compare competitor pricing and suggest optimal discount levels based on the historical performance of different discount rates.

- Dashboard: Visualize sales trends, forecasts, and inventory levels. Allow filtering by product category, region, and period.

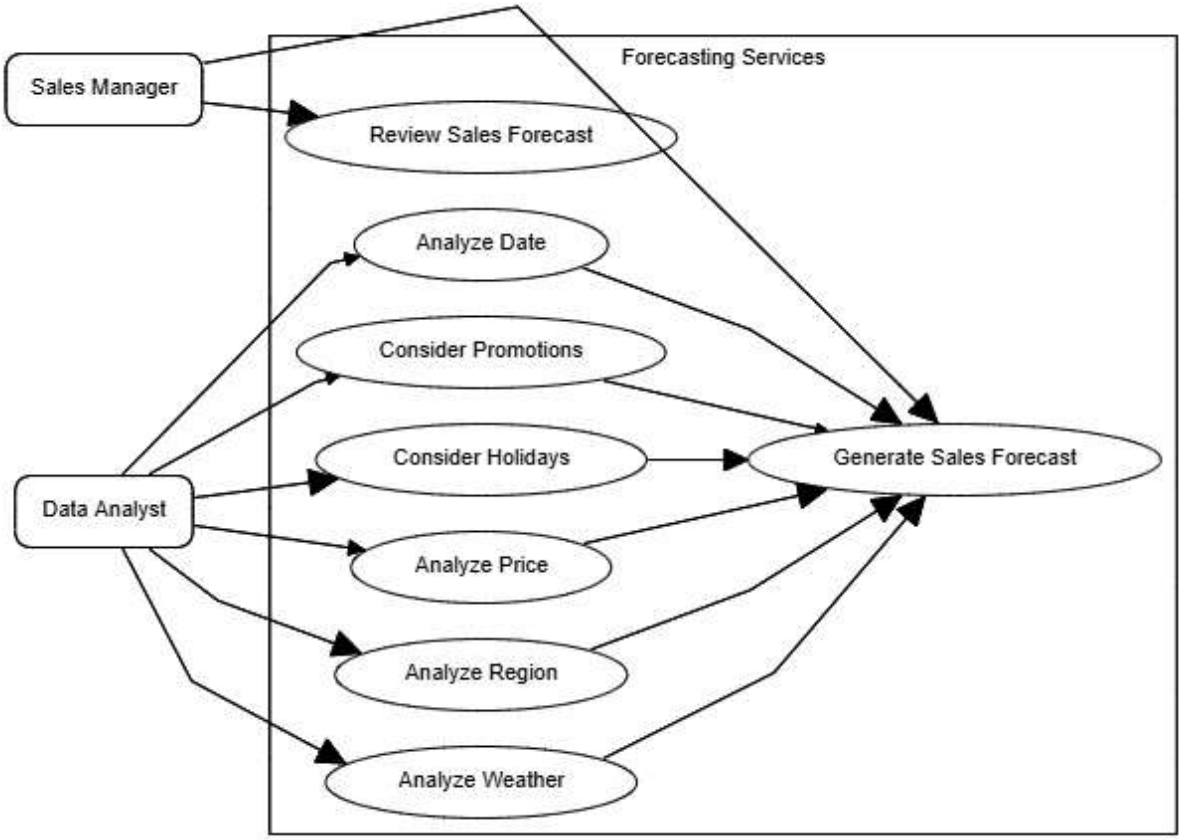
d) Non-functional Requirements

- Performance: The system should process large datasets efficiently with minimal lag.
- Security: Enforce role-based access and data encryption.
- Usability: Deliver an intuitive UI with easy data filtering and visualization.
- Reliability: Ensure high availability with minimal downtime.
- Scalability: Support multiple stores and increasing data volume.

4. System Analysis & Design

1) Problem Statement & Objectives – Define the problem being solved and project goals.

a) Use Case Diagram & Descriptions – Identify system actors and interactions.



b) Functional & Non-Functional Requirements – Clearly state system capabilities and constraints.

- **Functional Requirements**
  - Data Integrate with existing sales databases to collect historical data.
  - Implement multiple forecasting models to capture different sales patterns.
  - Provide interactive dashboards for forecast review and analysis.
  - Predict sales and Optimization for the next period.

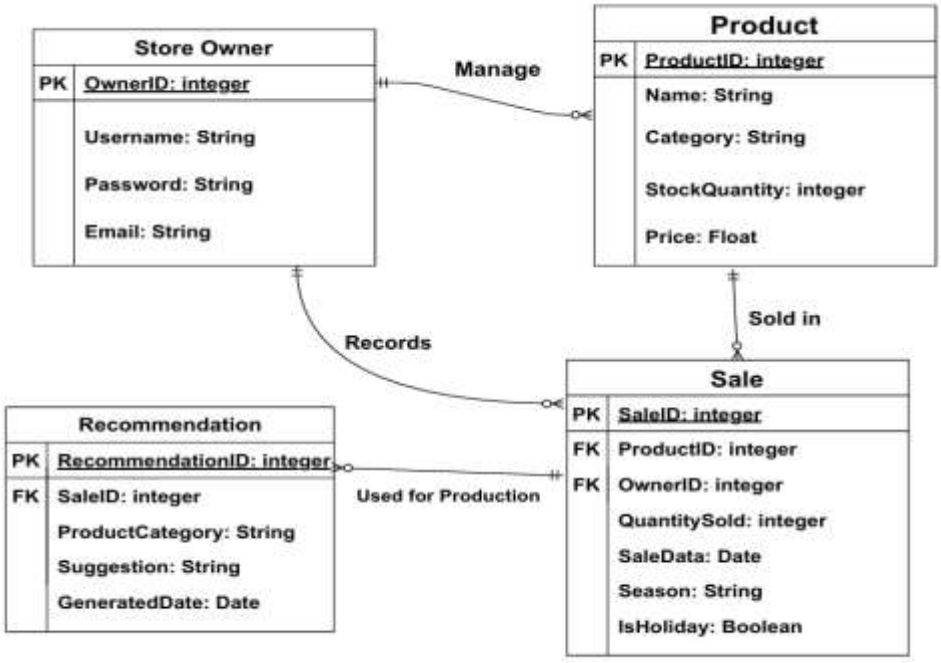
- **Non-Functional Requirements**
  - Performance: API response time < 1 second, dashboard load time < 3 seconds (for example).
  - Security: Data encryption, role-based access control.
  - Usability: Intuitive and mobile-responsive interface.
  - Scalability: Support growth in data volume and user base.

c) Software Architecture – High-level design outlining system components, interactions, and architecture style (e.g., MVC, Microservices).

- Data Ingestion Service: Collects and preprocesses data.
- Forecasting Service: Handles data ingestion, model training, and prediction.
- Dashboard Service: Provides a user interface for visualization.
- API Gateway: Manages communication between services (if is applicated)

2) Database Design & Data Modeling

- **ER Diagram (Entity-Relationship Diagram) – A well-defined ERD showcasing database structure and relationships.**



- **Logical & Physical Schema – Tables, attributes, keys, and normalization considerations.**

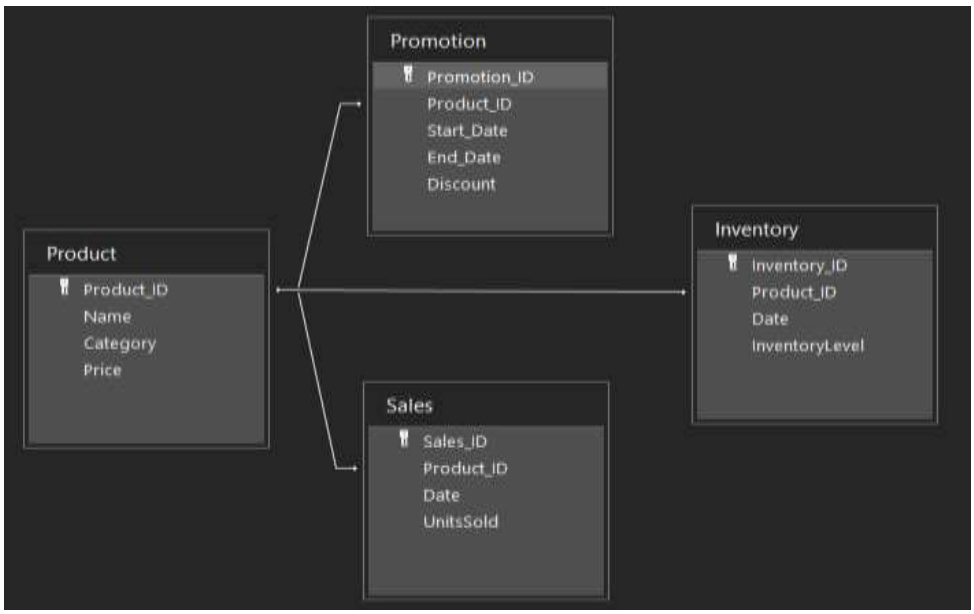


Figure 1. Logical Schema

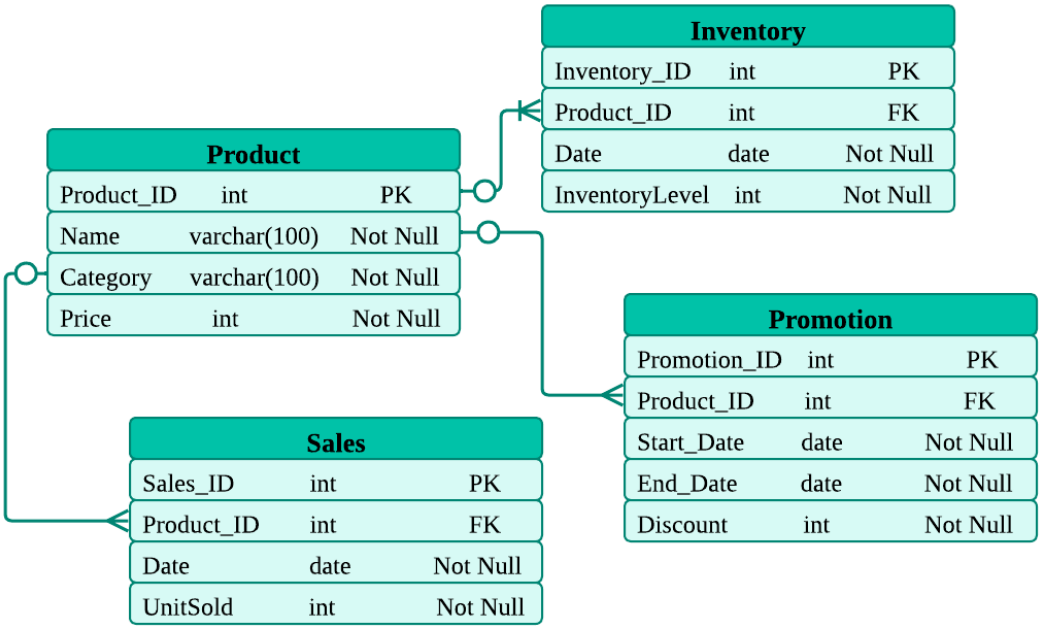
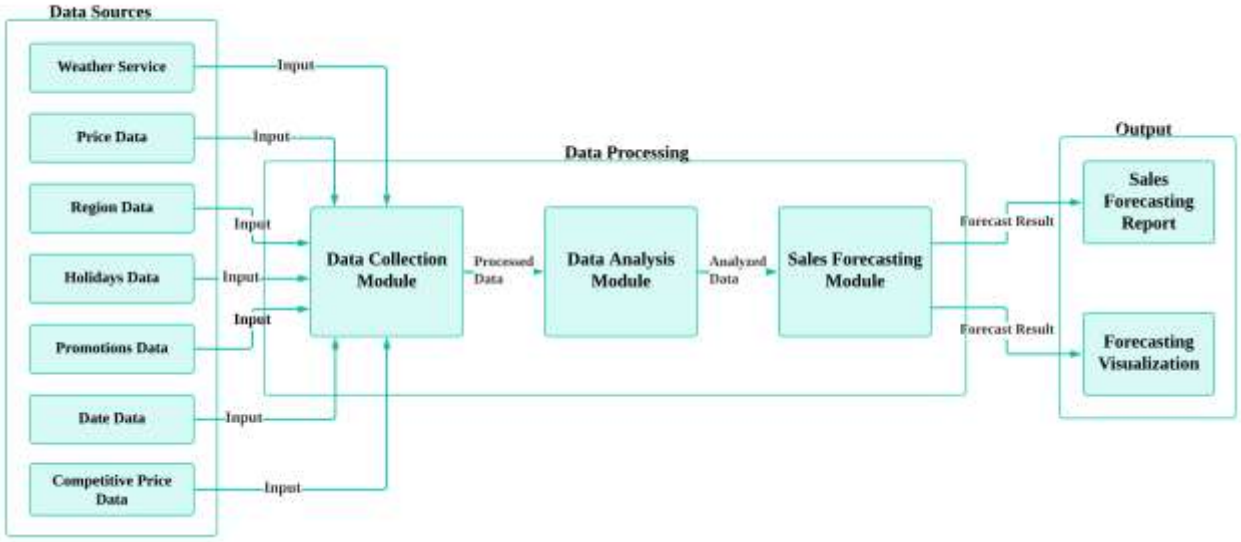


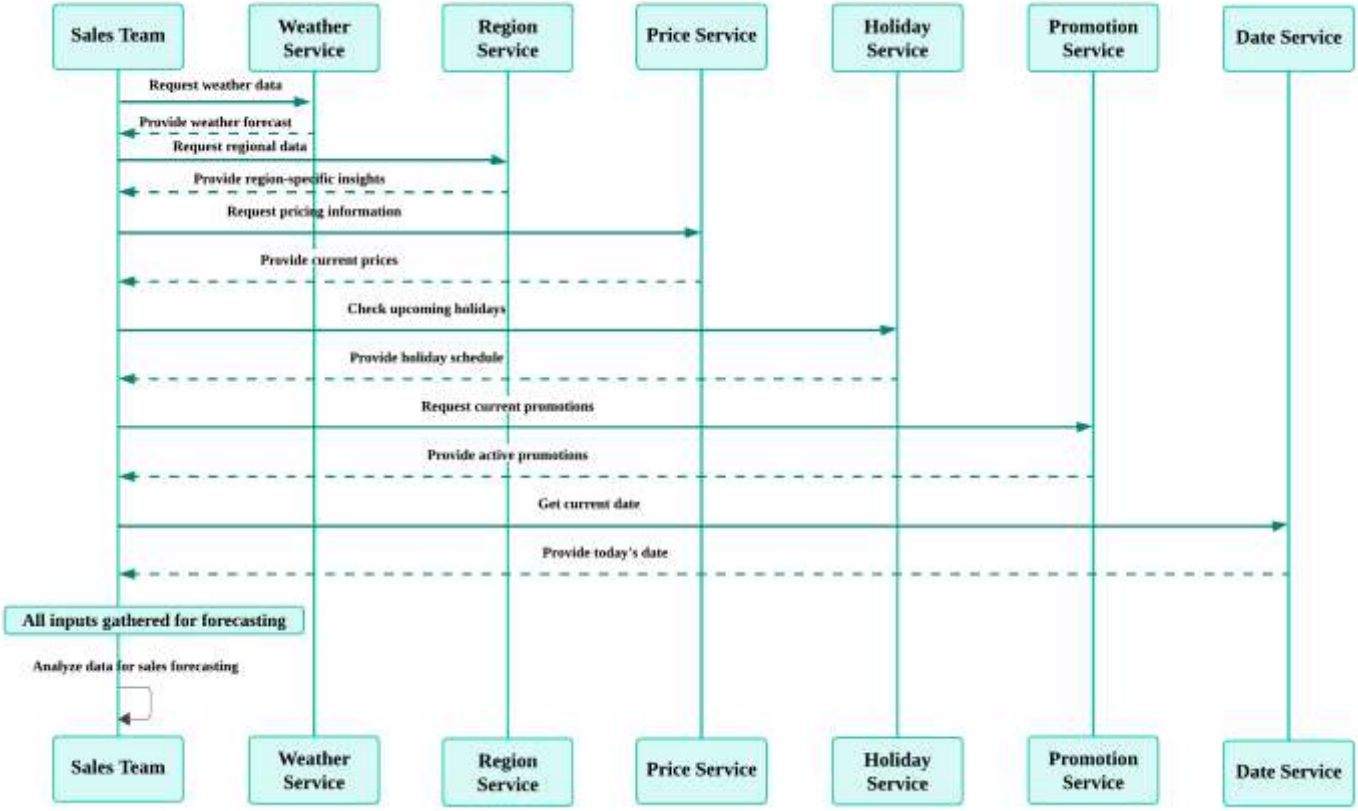
Figure 2. Physical Schema

3) Data Flow & System Behavior

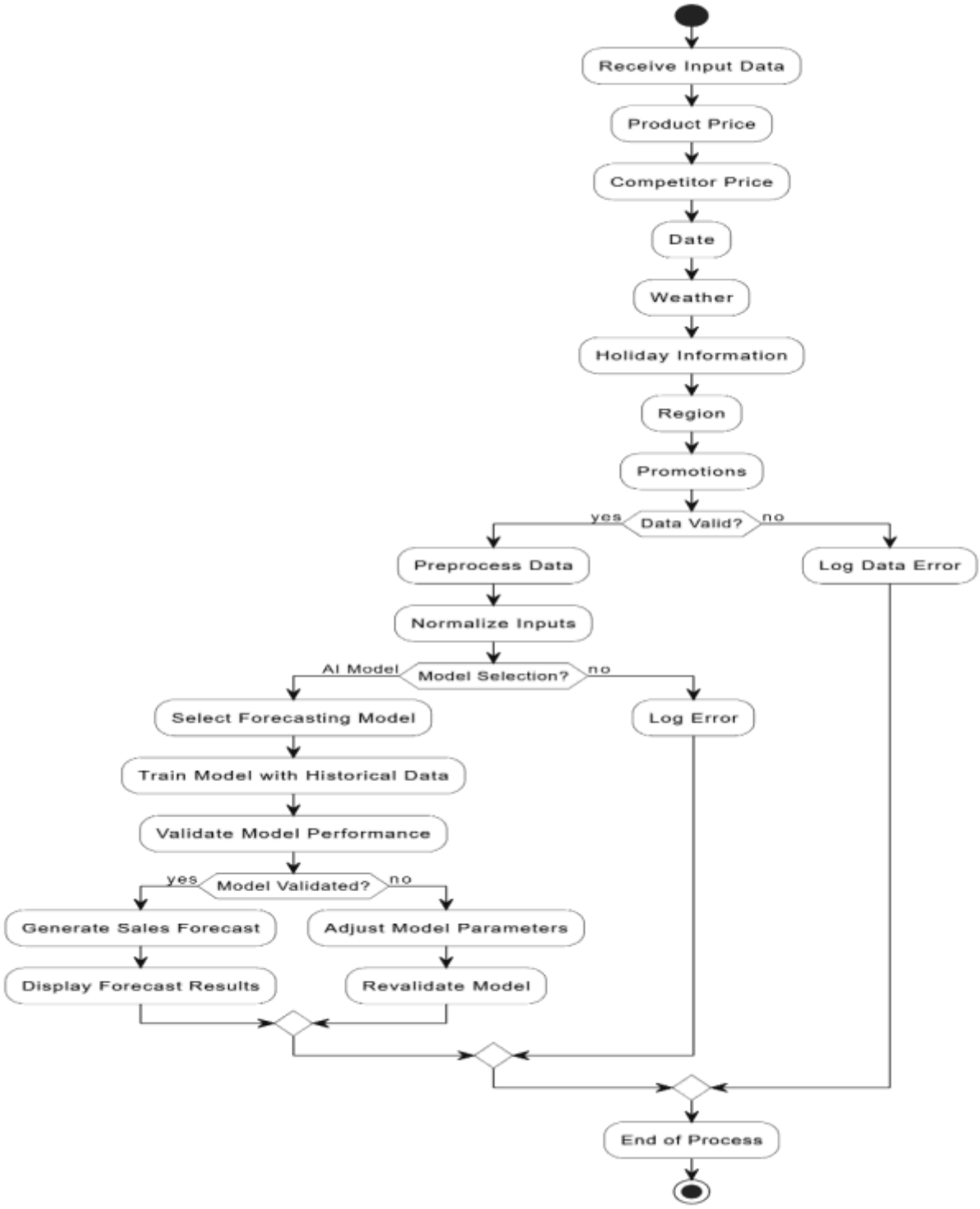
- **DFD (Data Flow Diagram) – Context-level and detailed levels showing how data moves through the system.**



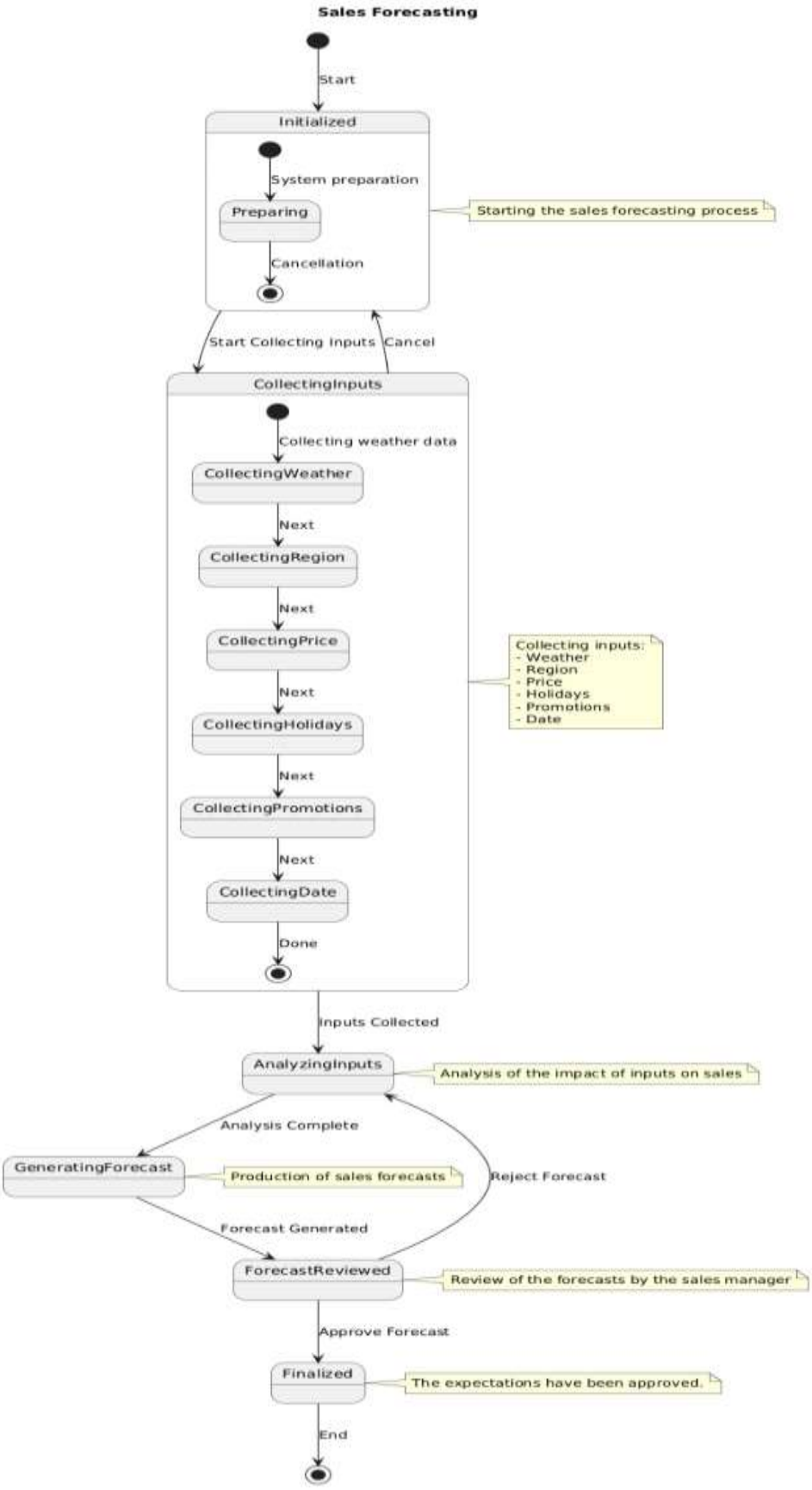
- **Sequence Diagrams – Process flow representation of key interactions between components.**



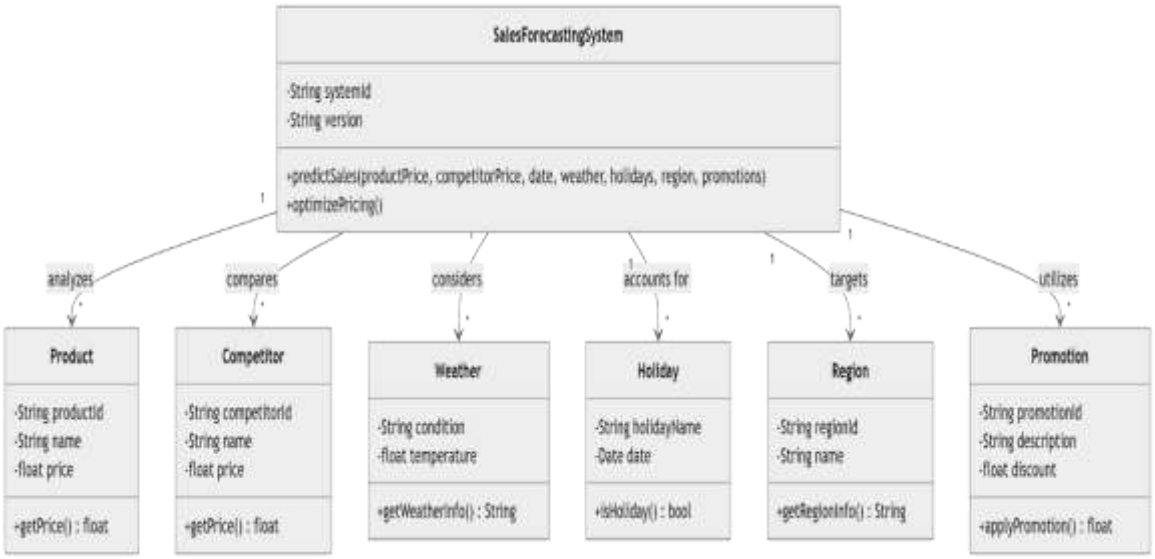
- **Activity Diagram – Visualizing the workflow of processes or user actions within the system.**



- **State Diagram – Represents different states of an object and how it transitions between them.**



- **Class Diagram – Defines the structure of the system by showing classes, attributes, methods, and relationships.**



4) UI/UX Design & Prototyping (not valid) ➔ We don't use it

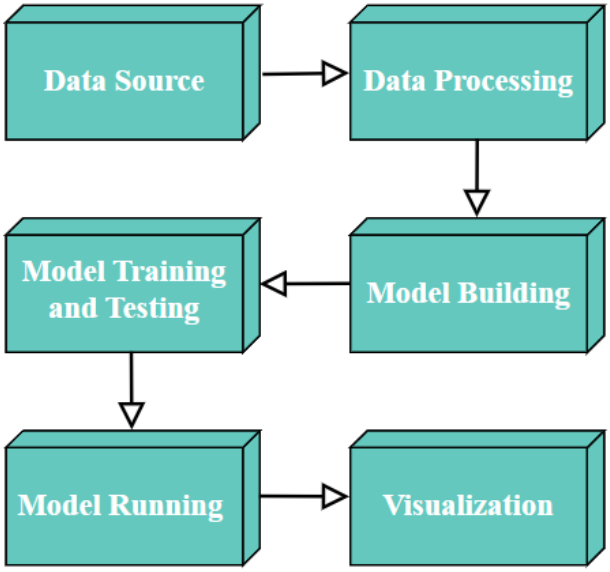
- Wireframes & Mockups – Screens and visual representations of the user interface.
- UI/UX Guidelines – Design principles, color schemes, typography, and accessibility considerations.

5) System Deployment & Integration

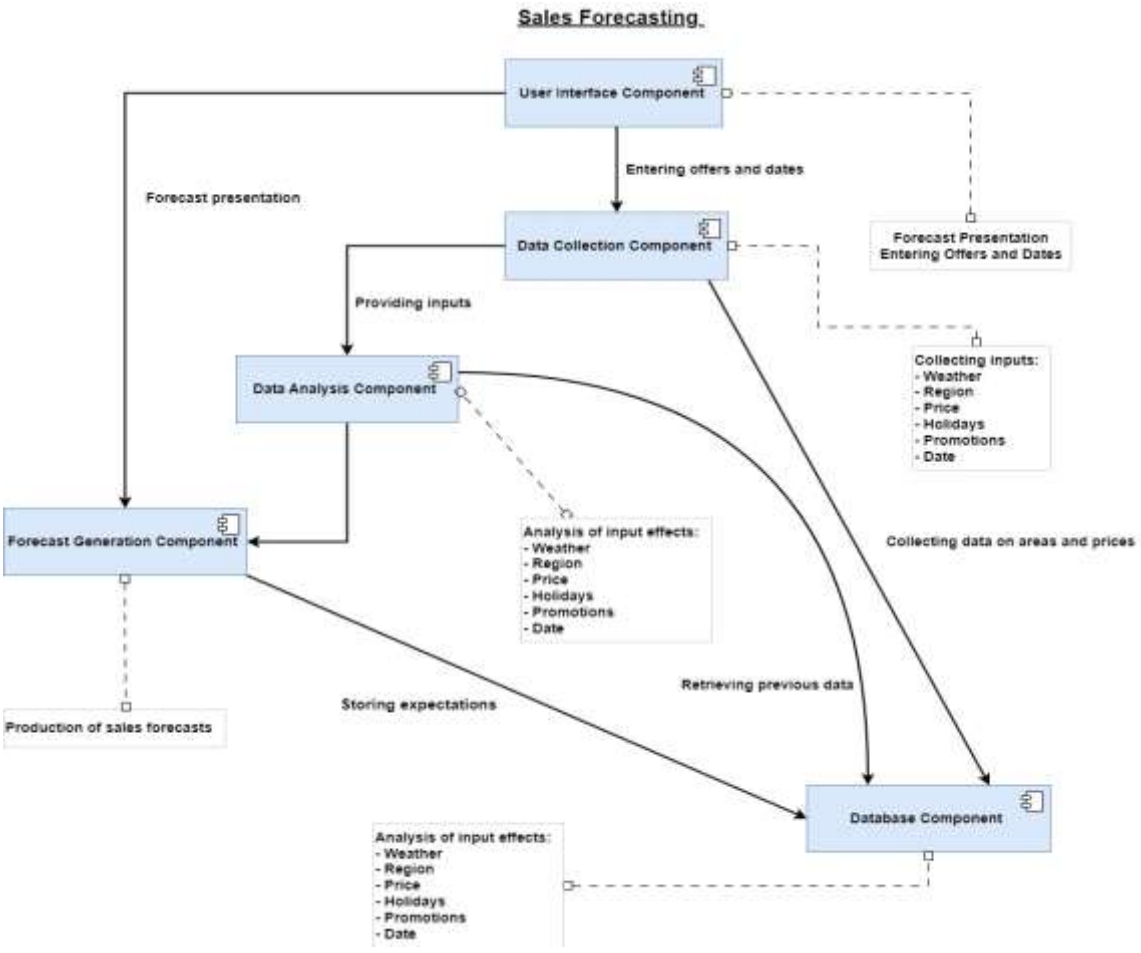
a) Technology Stack

- Data Processing & Machine Learning:
  - Programming Language: Python
  - Libraries: Pandas, NumPy, Scikit-learn, TensorFlow/PyTorch (if deep learning is used)
  - Visualization: Matplotlib, Seaborn, Plotly, Streamlit
- Deployment & Integration:
  - Environment: Jupyter Notebook or Google Colab for development
  - Containerization (if needed): Docker (optional, for easier deployment)
  - Cloud Services (if needed): Google Cloud AI Platform, AWS SageMaker, or Azure ML for large-scale deployment
  - API Framework: FastAPI (if API is used).

b) Deployment Diagram – Describes how software components are distributed across hardware.



c) Component Diagram – Shows high-level system components and their dependencies.



6) **Additional Deliverables (if applicable)** ➔ **We don't use it yet**

- **API Documentation – If the system includes APIs, provide documentation for endpoints and usage.**
- **Testing & Validation – Unit tests, integration tests, and user acceptance testing plan.**
- **Deployment Strategy – Hosting environment, deployment pipelines, and scaling considerations.**