

**“CortexX Demand Forecasts”**Technical Documentation & Project Specifications  
 *November 01, 2025*

|  |  |
| --- | --- |
| Sequence | Member Name |
| 1 | Muhammad Fathi Kamal |
| 2 | Amir Ayman Sobhy |
| 3 | Fares Essam Eldin Kamel |
| 4 | Mohamed Ashraf Abdel Samie |
| 5 | Abdullah Mostafa |

|  |  |
| --- | --- |
| Supervised by: | Dr: George Samuel Mahrous |
| Training company: | Amit Learning |

# Table of Contents

**1. Project Overview**

1.1 Purpose & Vision

1.2 Project Objectives

1.3 Scope & Deliverables

**2. Technical Architecture**

2.1 System Overview

2.2 Technology Stack

**3. Stakeholder Analysis**

**4. Data Flow Design**

**5. UI/UX Design**

5.1 Design Concept

5.2 User Flow

5.3 Wireframes

5.4 Sample Output

**6. Implementation Plan**

# 1. Project Overview

## 1.1 Purpose & Vision

CortexX is an enterprise-level sales forecasting and demand prediction platform designed to help businesses accurately predict sales, analyze trends, and optimize inventory and staffing decisions using advanced machine learning techniques. The platform transforms raw sales data into actionable insights through automated pipelines and interactive visualization.

## 1.2 Project Objectives

* Build a scalable, modular forecasting tool adaptable to various business domains
* Offer multiple ML models (XGBoost, LightGBM, Prophet, ensembles) for flexible forecasting
* Provide comprehensive exploratory data analysis and automated feature engineering
* Deliver an interactive Streamlit dashboard for real-time insights and reporting
* Support automated model training, retraining, and performance monitoring
* Ensure maintainability through clean code architecture and comprehensive documentation

## 1.3 Scope & Deliverables

### Functional Scope

* Data ingestion and pre-processing of sales and inventory data from CSV/Excel sources
* Automated feature creation for time-series forecasting including lag features and rolling statistics
* Training, evaluation, and comparison of sophisticated ML models with hyperparameter tuning
* Interactive visualization and reporting with export capabilities
* Flexible date handling and data validation for diverse business scenarios

### Key Deliverables

* Installation and setup scripts for seamless deployment
* Well-documented data pipeline and model training modules
* Real-time dashboard with data upload, analysis, and forecasting capabilities
* Exportable forecast reports and model performance documentation
* Comprehensive API documentation and user guides

# 2. Technical Architecture

## 2.1 System Overview

CortexX follows a modular microservices architecture with clear separation of concerns. The system is organized into distinct modules for data processing, feature engineering, model training, and visualization, enabling independent development and testing.

## 2.2 Technology Stack

|  |  |
| --- | --- |
| Component | Technologies |
| Backend & ML | Python, Scikit-learn, XGBoost, LightGBM, Prophet, Pandas, NumPy |
| Visualization | Streamlit, Plotly, Matplotlib, Seaborn |
| Data Processing | Pandas, NumPy, SciPy for statistical operations |
| Development | Git, Poetry for dependency management, Pytest for testing |
| Deployment | Docker, Streamlit Cloud compatible |

# 3. Stakeholder Analysis

Effective stakeholder management is crucial for project success. The following table outlines key stakeholders, their roles, and communication strategies.

|  |  |  |
| --- | --- | --- |
| Stakeholder | Role & Responsibilities | Communication Plan |
| Project Manager | Oversees project delivery and timeline | Weekly status meetings, email updates, milestone reviews |
| Data Scientist | Develops ML models and feature engineering | Daily stand-ups, JIRA tickets, Slack coordination |
| Software Developer | Implements dashboard and backend services | Code reviews, sprint demos, Slack channels |
| Business Analyst | Defines business requirements and success metrics | Requirement workshops, status reports, email communication |
| End Users | Utilize forecasting platform for business decisions | Training sessions, feedback collection, user support channels |

**Communication Strategy:** Scheduled meetings for formal updates, Slack channels for daily coordination, and email for official documentation and decision tracking.

# 4. Data Flow Design

CortexX employs a flexible file-based data ingestion system designed for rapid prototyping and deployment without complex database infrastructure requirements.

## Data Sources & Structure

* Primary data source: CSV files (retail\_store\_inventory.csv)
* Supported formats: CSV, Excel with flexible schema adaptation
* Key columns: Date, Store ID, Product ID, Category, Region, Inventory Level, Units Sold, Demand Forecast, Price, Discounts

## Data Processing Pipeline

1. Data Ingestion: Flexible CSV/Excel parsing with automatic date detection
2. Pre-processing: Handling missing values, outliers, and data validation
3. Feature Engineering: Automated creation of time-series features (lags, rolling statistics, seasonal patterns)
4. Model Training: Multiple algorithm support with automated hyperparameter tuning
5. Inference & Reporting: Forecast generation with confidence intervals and performance metrics

# 5. UI/UX Design

## 5.1 Design Concept

* Intuitive navigation through sidebar-based workflow progression
* Consistent visual language using Plotly with clean, business-appropriate themes
* Progressive disclosure of complexity - simple defaults with advanced options
* Real-time feedback and validation throughout user interactions

## 5.2 User Flow

1. Data Upload: Upload CSV/Excel or generate sample data for testing
2. Configuration: Auto-detection or manual selection of date and value columns
3. Exploration: Perform EDA with interactive visualizations and statistical summaries
4. Feature Engineering: Automated feature creation with customization options
5. Model Training: Configure and train forecasting models with selectable algorithms
6. Forecasting: Generate and visualize sales forecasts with confidence interval

## 5.3 Wireframes Overview

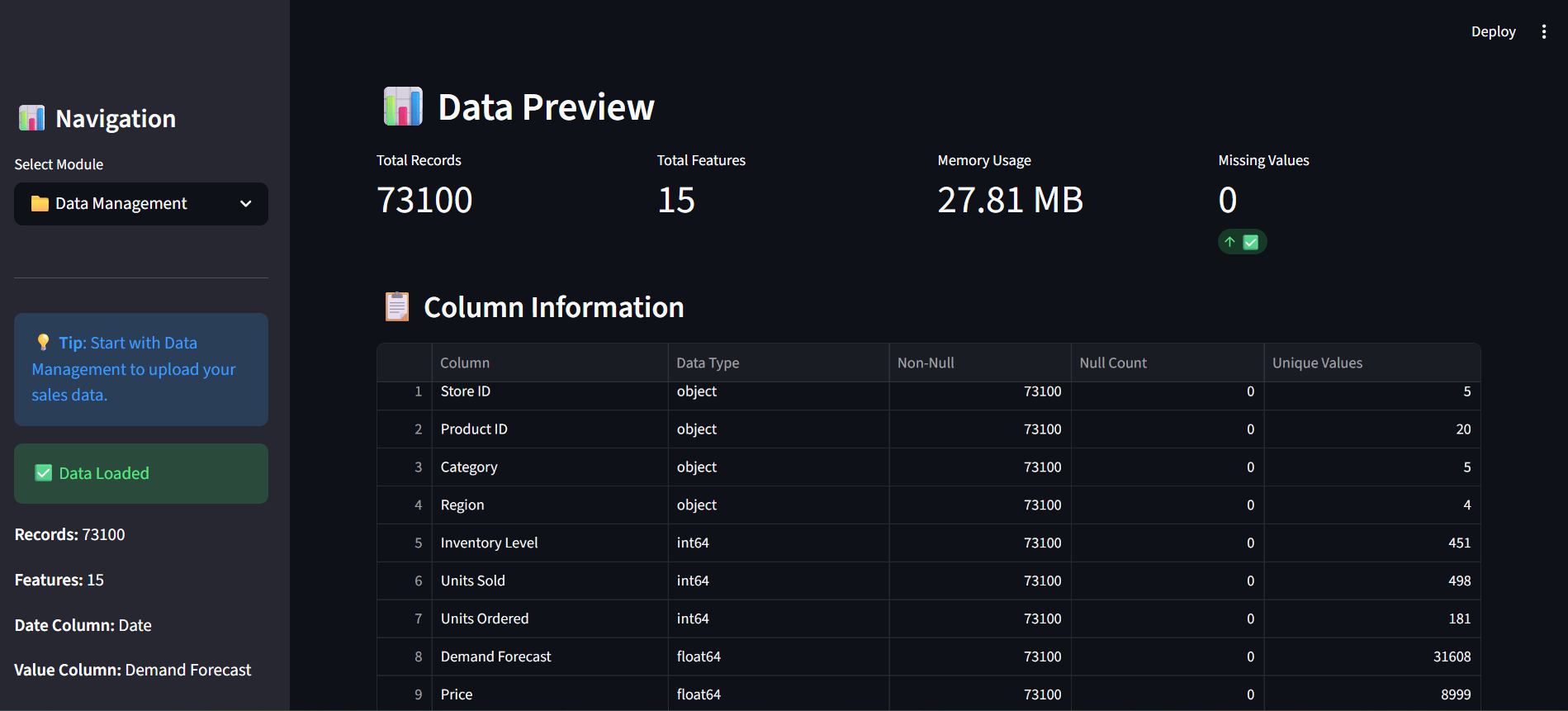
* Grouped navigation tabs for logical workflow progression
* Dynamic sidebar displaying data state and session information
* Interactive data preview tables with sorting and filtering capabilities
* Multiple Plotly graph types supporting detailed exploratory analysis
* Model configuration panels with preset and custom options
* Export functionality for forecasts, charts, and performance reports

### Design Rationale

* Streamlit enables rapid development cycles and delivers intuitive user interfaces without complex frontend development
* Plotly provides enterprise-grade interactive visualization suitable for business intelligence applications
* Modular codebase supports maintainability, extensibility, and collaborative development practices
* Responsive design ensures accessibility across different devices and screen sizes

# 5.4 Sample Output:





# 6. Implementation Plan

## Development Phases

|  |  |
| --- | --- |
| Phase | Deliverables |
| Phase 1: Core Infrastructure | Data pipeline, basic ML models, foundational dashboard |
| Phase 2: Advanced Features | Ensemble models, advanced EDA, performance optimization |
| Phase 3: Production Ready | Error handling, comprehensive testing, documentation |
| Phase 4: Deployment & Training | User training, deployment scripts, support materials |

## Success Metrics

* Model Accuracy: MAPE < 15% on validation datasets
* Performance: Dashboard load time < 3 seconds for standard datasets
* Usability: User satisfaction score > 4.0/5.0
* Reliability: System uptime > 99.5% in production environment

# GitHub Repo: <https://github.com/Elgeneral200/CortexX>