Sales Forecasting and Demand Prediction – Project Document

1. Project Proposal

Overview

The Sales Forecasting and Demand Prediction project builds a machine learning solution to forecast product sales and demand from historical data. Accurate forecasts support inventory optimization, staffing, and marketing decisions. The project spans data acquisition and EDA through feature engineering, model development and optimization, and ends with deployment and monitoring to enable data- driven planning.

Objectives

- Develop robust forecasting models based on historical sales and contextual factors.
- Improve decision- making for inventory, staffing, and marketing through predictive analytics.
- Enable reproducible workflows from experimentation to deployment and monitoring (MLOps).
- Provide interactive visualizations and dashboards for exploratory analysis and consumption of forecasts.

Scope

- Data collection from open sources (e.g., Kaggle, UCI) or company databases; integration of sales, product, customer, promotions, seasonality, and external indicators (e.g., holidays, weather).
- Exploratory data analysis (EDA), preprocessing, and feature engineering (time- based, lag/rolling, promotional flags, category encodings).
- Model development using time- series and machine learning approaches (e.g., ARIMA/ETS, tree- based methods, LSTM) with hyperparameter tuning.
- Deployment as an API or web service (Flask/FastAPI) and dashboarding (e.g., Streamlit) with monitoring, alerts, and retraining policies.

2. Project Plan

Timeline (Gantt)

The following Gantt chart illustrates the tentative timeline for Milestones 1–4. Milestone 5 timeline is TBD and will be finalized later.

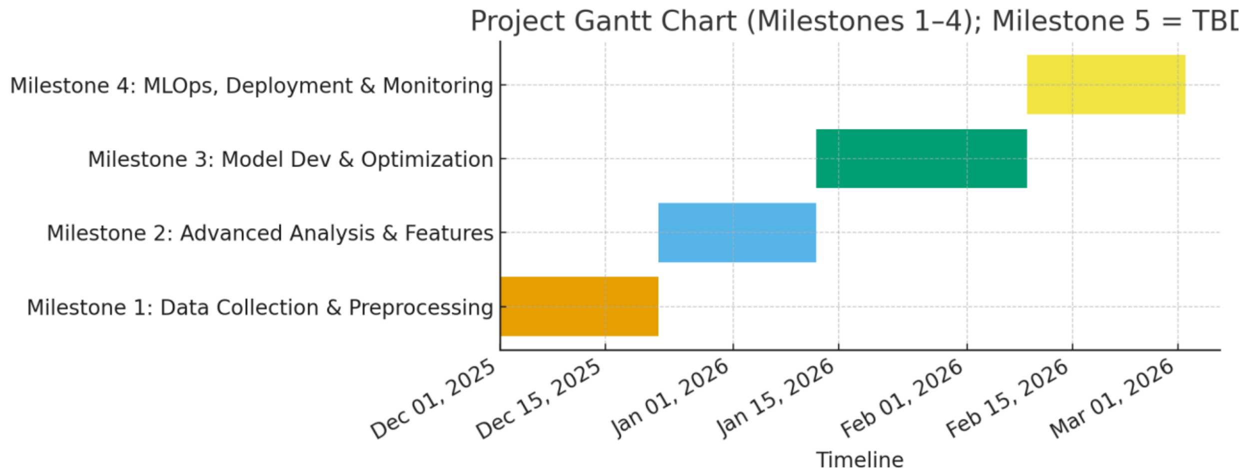

Figure 1: Project Timeline (Milestones 1–4). Milestone 5 timeline: TBD.

Milestones & Deliverables

Milestone	Duration	Start	End	Key Deliverables
1. Data Collection, Exploration, and Preprocessing	21 days	Dec 01, 2025	Dec 22, 2025	EDA Report; Interactive Visualizations; Cleaned Dataset
2. Advanced Data Analysis and Feature Engineering	21 days	Dec 22, 2025	Jan 12, 2026	Data Analysis Report; Enhanced Visualizations; Feature Engineering Summary

3. Model Development and Optimization	28 days	Jan 12, 2026	Feb 09, 2026	Model Evaluation Report; Model Code; Final Model
4. MLOps, Deployment, and Monitoring	21 days	Feb 09, 2026	Mar 02, 2026	Deployed Model; MLOps Report; Monitoring Setup
5. Final Documentation and Presentation				

Resource Allocation

Phase	Human Resources	Tools & Technologies	Compute / Infrastructure
Milestone 1	Data Engineer; Data Scientist	Python, Pandas, NumPy, Jupyter, SQL	Local + Cloud storage
Milestone	Data	Statsmodels,	Local compute /

2	Scientist; Analyst	Scikit- learn, Featuretools	GPU as needed
Milestone 3	ML Engineer; Data Scientist	Scikit- learn, XGBoost/LightGBM, TensorFlow/PyTorch	GPU/CPU instances for training
Milestone 4	ML Engineer; DevOps	MLflow/DVC, FastAPI/Flask, Streamlit/Dash, Prometheus	Cloud runtime (AWS/GCP/Azure), CI/CD
Milestone 5			

3. Task Assignment & Roles

Enter team details below. Cells are intentionally left blank for later assignment.

Team Member	Role	Assigned Tasks

4. Risk Assessment & Mitigation Plan

Key Risks

- Incomplete or missing data for certain products, time windows, or external indicators.
- Data quality issues: missing values, duplicates, outliers, inconsistent schemas/IDs.
- Model risk: concept drift and seasonal shifts reducing forecast accuracy over time.
- Operational risk: deployment issues, API downtime, or insufficient monitoring/alerts.

Mitigation Strategies

- Establish a repeatable ingestion pipeline; document data gaps and assumptions.
- Impute or remove missing values; deduplicate and cap/treat outliers; align entity keys.
- Use rolling/expanding windows and periodic retraining; track drift with accuracy thresholds.
- Implement robust MLOps with observability and alerting; provide fallback/baseline forecasts.

Normalization Statement & Implications

Normalization (e.g., standardization or min- max scaling) has been applied during preprocessing to stabilize model training and improve numerical conditioning. This reduces the influence of scale differences across features and can improve convergence for gradient- based models. Implications: model coefficients are learned on normalized scales and should be inverse- transformed for interpretation; monitoring must use the same transformation; drift detection should consider both raw and normalized metrics.

5. KPIs (Key Performance Indicators)

- Forecast Accuracy (%):
- Mean Absolute Error (MAE):
- Root Mean Squared Error (RMSE):
- Response Time (API):
- System Uptime (%):
- User Adoption Rate (% active users):

6. Milestone Details

Milestone 1: Data Collection, Exploration, and Preprocessing

Objectives & Tasks

- Collect sales and demand data from open sources (Kaggle, UCI) or company databases with product, customer, promotions, seasonality, holidays, weather indicators.
- Conduct EDA to assess trends, seasonality, promotional effects; compute summary statistics; address missing values, duplicates, and outliers.
- Preprocess and engineer features: time- based (month/week/day), product categories, promotion flags; encode categorical variables; normalize numerical features; create lag features.

Deliverables

- EDA Report
- Interactive Visualizations / EDA Notebook
- Cleaned Dataset (ready for forecasting)

Milestone 2: Advanced Data Analysis and Feature Engineering

Objectives & Tasks

- Perform time- series analysis: trends, seasonality, cycles; run stationarity tests (e.g., ADF).

- Correlation analysis across features (sales, promotions, holidays, weather); create rolling averages, lags, seasonal/holiday components.
- Transformations: scaling/encoding; aggregations (e.g., monthly totals); integrate external factors to boost accuracy.
- Develop advanced visualizations/dashboards explaining demand patterns and factor impacts.

Deliverables

- Data Analysis Report
- Enhanced Visualizations / Dashboards
- Feature Engineering Summary

Milestone 3: Machine Learning Model Development and Optimization

Objectives & Tasks

- Model selection: ARIMA/ETS and ML models (Random Forest, Gradient Boosting, LSTM) suited for seasonality, trend, and external regressors.
- Time- aware training/validation (rolling/blocked CV); guard against leakage.
- Evaluate with MAE, MSE, RMSE, R^2; generate residual plots to diagnose fit and error patterns.
- Hyperparameter tuning (Grid/Random Search); compare models and select best based on accuracy and applicability.

Deliverables

- Model Evaluation Report
- Model Code
- Final Model (ready for deployment)

Milestone 4: MLOps, Deployment, and Monitoring

Objectives & Tasks

- MLOps implementation: manage experiments and versions (MLflow/DVC); log metrics, parameters, and artifacts.
- Model deployment as API/web service (Flask/FastAPI) for real- time or batch forecasting; optional cloud deployment for scalability.
- Monitoring: track forecast accuracy and drift; establish alerting for performance degradation.
- Retraining strategy: periodic updates based on new data, seasonality, and external changes.

Deliverables

- Deployed Model (API/service)
- MLOps Report
- Monitoring Setup & Runbook

Milestone 5: Final Documentation and Presentation

Objectives & Tasks

-
-

-

Deliverables

- Final Project Report
- Final Presentation

7. Additional Notes

- This document uses consistent heading styles, spacing, and bullet formatting for readability.
- Milestone 5 details and specific KPI values are intentionally left blank for later input.
- Adjust the Gantt timeline dates as needed once the official start date is confirmed.