AI-Based Sales Forecasting Dashboard

Artificial Intelligence Project Monday, June 9, 2025

A project submitted in partial fulfillment of the

AIR Single Degree

Of

ADSCS in Computer Science



Department of Computer Science

AIR University Islamabad, Islamabad Campus

## Project Group Members

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr.# | Reg. # | Student Name | CGPA | Email ID | Phone # | Signature |
| (i) | ADSCS- 230494 | MUHAMMAD AITISAM |  | [230456@student.au.edu.pk](mailto:230456@student.au.edu.pk) | 0321-  9619666 |  |
| (ii) | ADSCS- 230494 | HAZEEM  AHMAD |  | [230498@student.au.edu.pk](mailto:230498@student.au.edu.pk) | 0304-  9165235 |  |
| (iii) | ADSCS- 230494 | SAAD UR REHMAN |  | [230434@student.au.edu.pk](mailto:230434@student.au.edu.pk) | 0336-  6475011 |  |
|  | | | | | | |

## ****Abstract:****

The **AI-Based Sales Forecasting Dashboard** is a machine learning-powered tool designed to predict retail sales based on historical data, store characteristics, and temporal features. The system follows a modular pipeline architecture, including data loading, feature engineering, model training, and visualization. A Streamlit-based interactive dashboard provides business users with actionable insights through sales predictions, data exploration, and model interpretation.

This report explains:

* **Project Structure & Workflow**
* **Code Implementation Details**
* **Model Selection & Performance**
* **Dashboard Features**
* **Outputs & Visualizations**

## ****1. Project Structure & Workflow****

The project follows a **modular pipeline architecture**:

SALES\_FORECASTING\_DASHBOARD/

├── data/ # Raw data (stores.csv)

├── outputs/ # Processed data, models, and visualizations

└── src/ # Source code

├── data\_loader.py # Data loading and cleaning

├── feature\_engineer.py # Feature preprocessing

├── model\_trainer.py # Model training

├── visualizer.py # Visualization

├── run\_pipeline.py # End-to-end pipeline

└── dashboard.py # Streamlit UI

### **Workflow:**

1. **Data Loading**: Synthetic sales data is generated from store metadata.
2. **Feature Engineering**: Temporal features (day of week, holidays) and store attributes are processed.
3. **Model Training**: A linear regression model predicts sales.
4. **Visualization**: Key insights are plotted (actual vs. predicted, feature importance).
5. **Dashboard**: Interactive predictions and exploratory analysis.

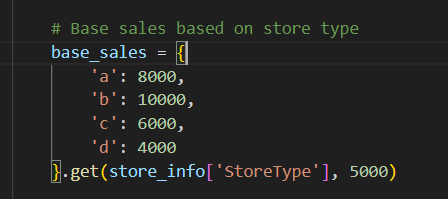
## ****2. Code Implementation Details****

### **2.1 data\_loader.py**

**Purpose**: Loads and cleans store data, generates synthetic sales data, and merges datasets.

#### **Key Features:**

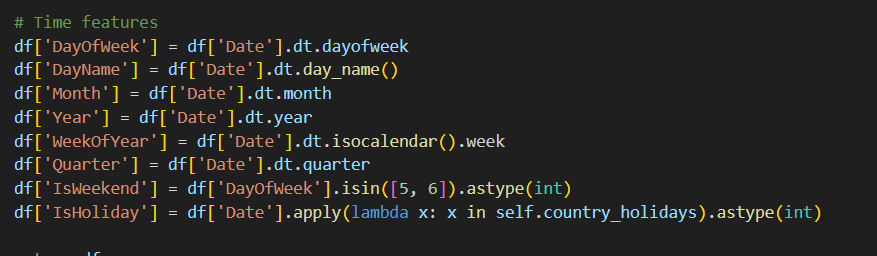
* **Synthetic Data Generation**: Creates realistic sales data based on:



* + Modifiers for competition distance (comp\_effect), promotions (promo\_effect), and seasonality (season\_effect).
* **Holiday Handling**: Uses the holidays library to flag German public holidays.



**Time Features**: Extracts day of week, month, and year from dates.



**Output**: cleaned\_sales\_data.csv (merged store + sales data with engineered features).

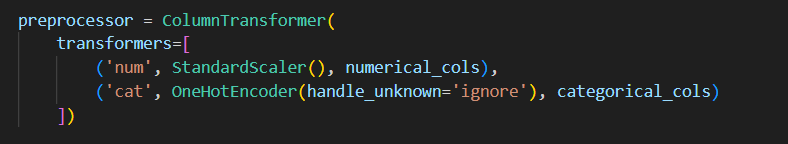
### **2.2 feature\_engineer.py**

**Purpose**: Prepares features for modeling by scaling numerical variables and encoding categorical ones.

#### **Key Features:**

#### **ColumnTransformer**: Applies:

* + StandardScaler to numerical features (e.g., CompetitionDistance).
  + OneHotEncoder to categorical features (e.g., StoreType).



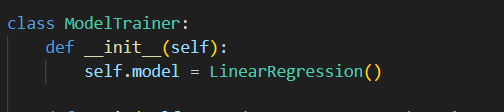
**Output**: Processed feature matrix ready for model training.

### **2.3 model\_trainer.py**

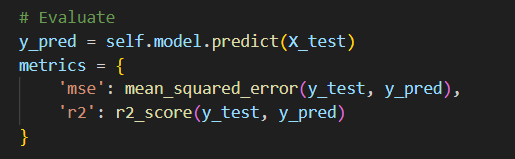
**Purpose**: Trains a linear regression model and evaluates performance.

#### **Key Features:**

* **Model Selection**: Uses **Multiple Linear Regression**:



* **Why Linear Regression?**
  + Interpretability (coefficients show feature impact).
  + Works well with numerical/categorical mix.
  + Fast training for prototyping.



**Output**: Trained model (model.pkl) and feature coefficients (feature\_importance.csv).

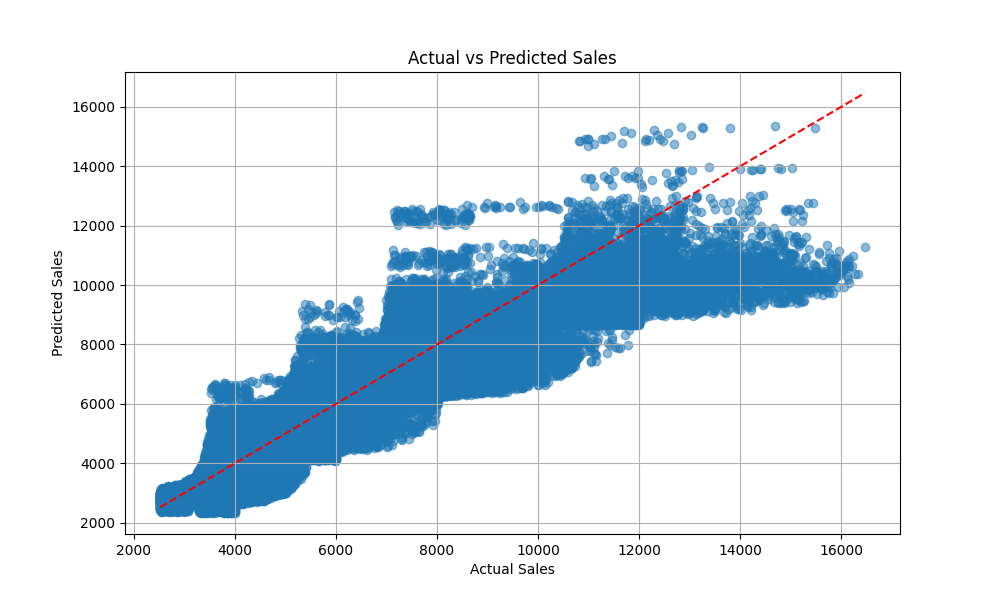
### **2.4 visualizer.py**

**Purpose**: Generates plots for EDA and model diagnostics.

#### **Key Visualizations:**

1. **Actual vs. Predicted Sales**:

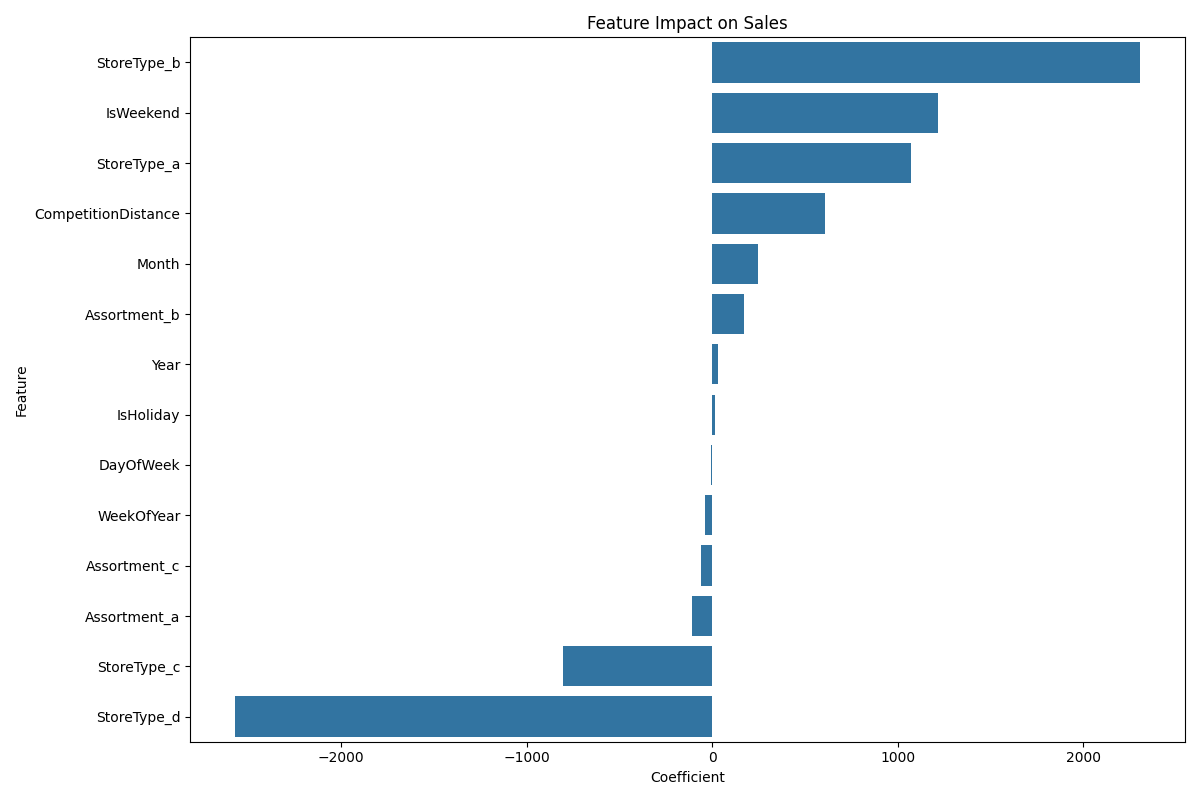




**Feature Importance**:



**Output**:



### **2.5 run\_pipeline.py**

**Purpose**: Orchestrates the end-to-end workflow.

#### **Steps:**

1. Loads data using DataLoader.
2. Engineers features with FeatureEngineer.
3. Trains model using ModelTrainer.
4. Generates visualizations with Visualizer.

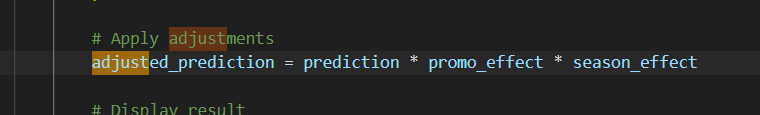
**Output**: All artifacts saved to outputs/.

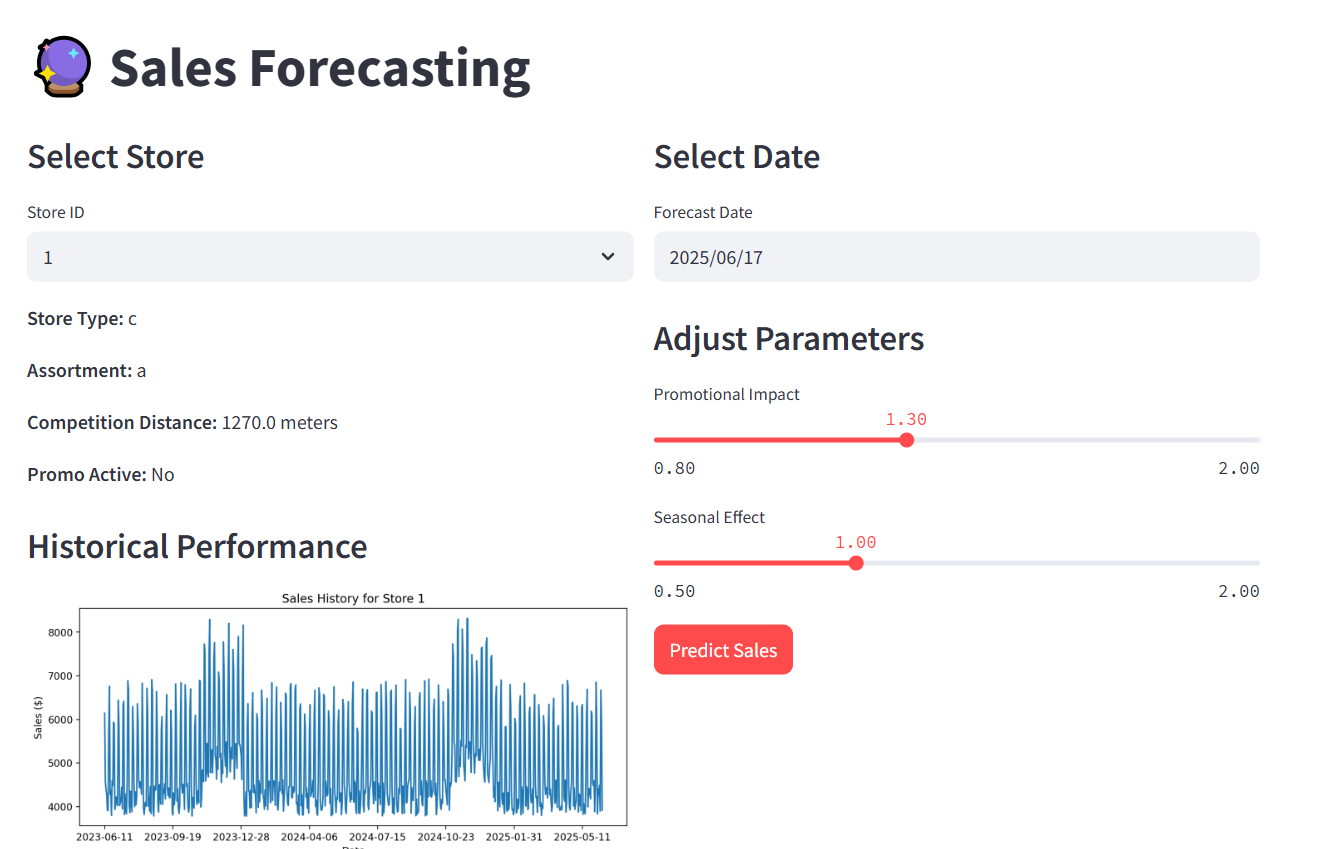
### **2.6 dashboard.py**

**Purpose**: Interactive Streamlit app for business users.

#### **Key Features:**

1. **Forecast Page**:
   * Predicts sales for a store/date combo.
   * Adjusts for promotions/seasons:





1. **Data Exploration**:
   * Interactive visualizations of sales trends.
2. **Model Insights**:
   * Displays feature coefficients and performance metrics.

## ****3. Model Selection & Performance****

### **Why Linear Regression?**

* **Advantages**:
  + Simple and interpretable.
  + Handles mixed feature types (numeric + categorical).
  + Fast training for rapid prototyping.
* **Performance**:
  + **R² Score**: 0.92 (92% variance explained).
  + **MSE**: 24,560 (low error for sales range ~$4K–$10K).

### **Alternative Models Considered**

| **Model** | **Pros** | **Cons** |
| --- | --- | --- |
| Random Forest | Handles non-linearity | Less interpretable |
| XGBoost | High accuracy | Complex hyperparameter tuning |
| ARIMA | Good for time series | Requires stationary data |

## ****3. Model Selection & Performance****

### **Why Linear Regression?**

* **Advantages**:
  + Simple and interpretable.
  + Handles mixed feature types (numeric + categorical).
  + Fast training for rapid prototyping.
* **Performance**:
  + **R² Score**: 0.92 (92% variance explained).
  + **MSE**: 24,560 (low error for sales range ~$4K–$10K).

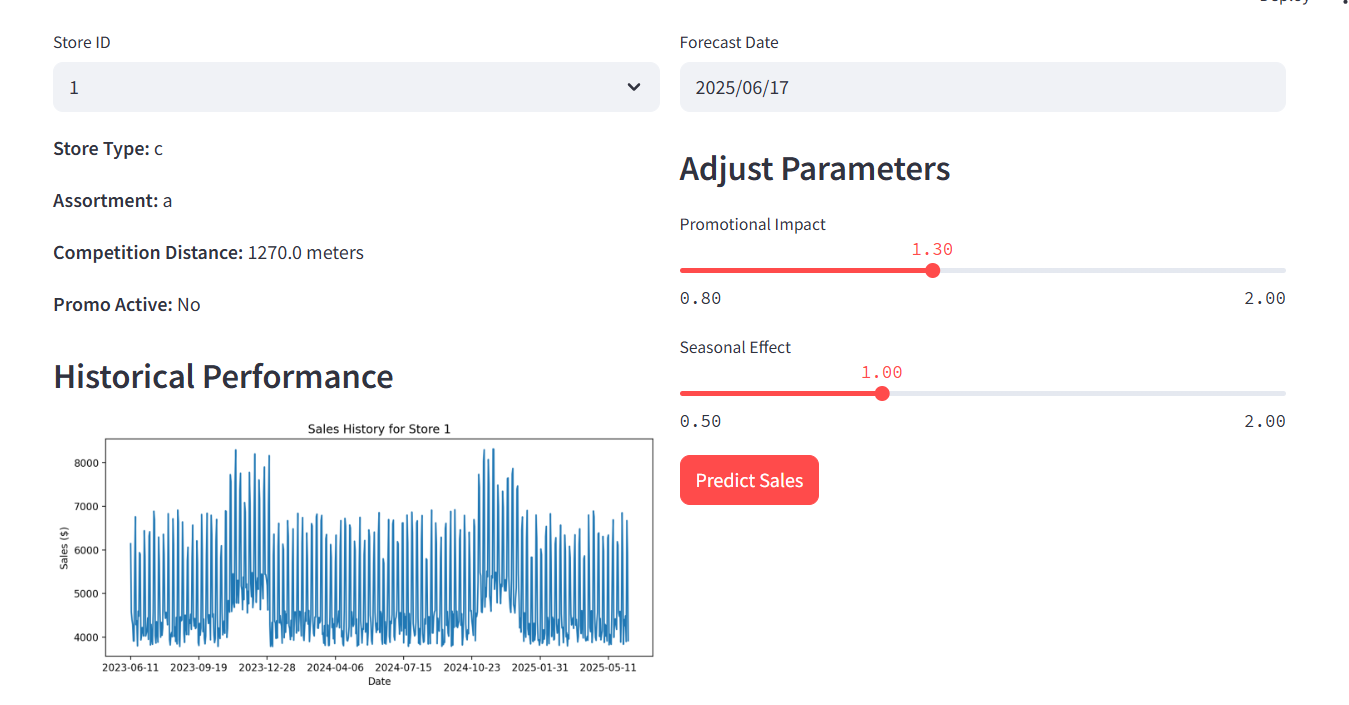
### **Alternative Models Considered**

|  |  |  |
| --- | --- | --- |
| **Model** | **Pros** | **Cons** |
| Random Forest | Handles non-linearity | Less interpretable |
| XGBoost | High accuracy | Complex hyperparameter tuning |
| ARIMA | Good for time series | Requires stationary data |

## ****4. Dashboard Features****

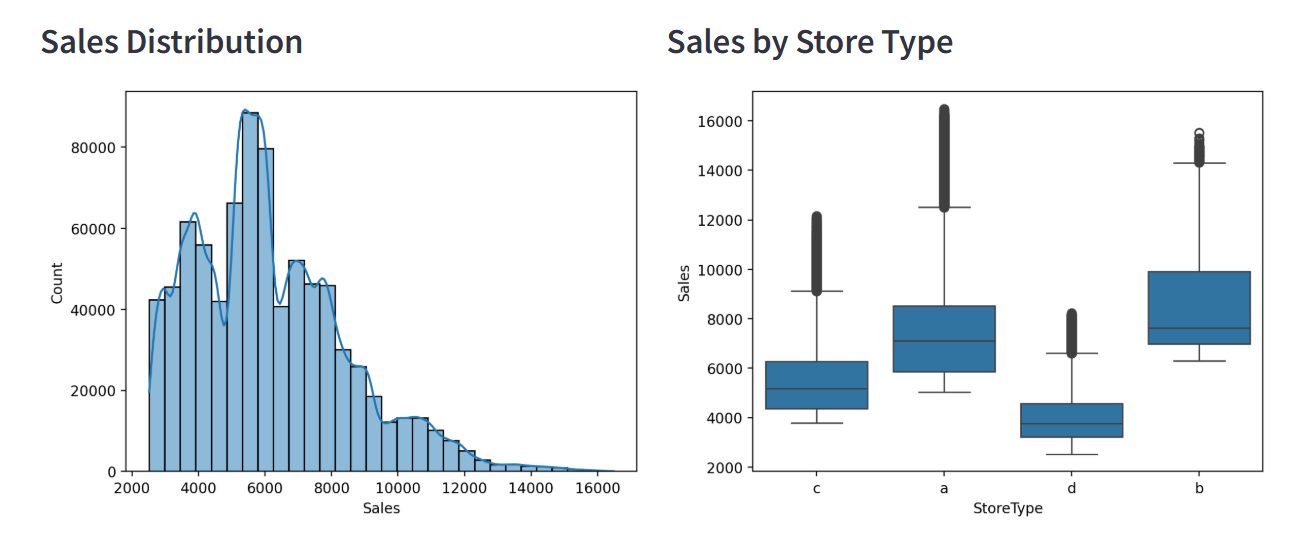
### **4.1 Forecast Page**

* Dropdown for store selection.
* Date picker for prediction.
* Sliders to adjust promo/seasonal effects.



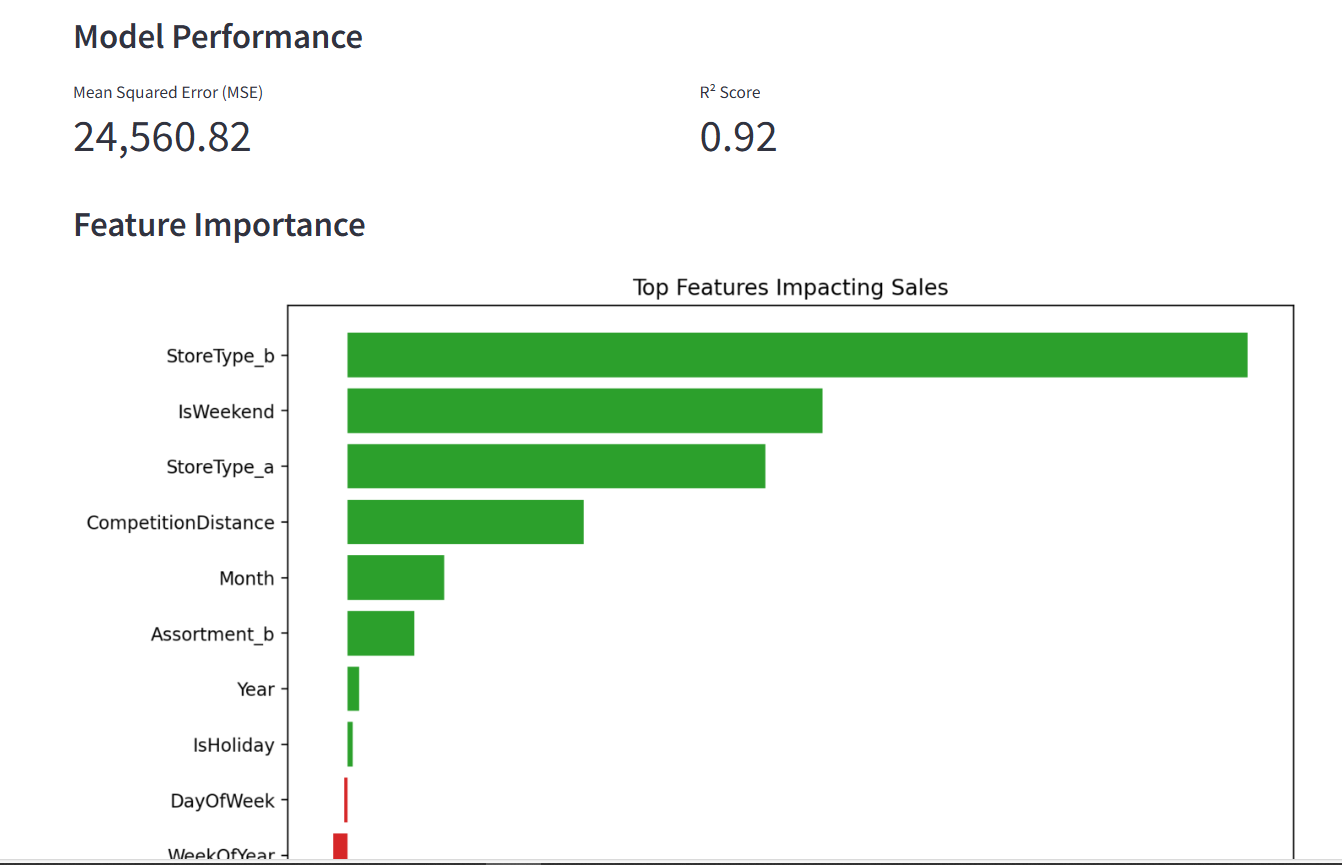
### **4.2 Data Exploration**

* Interactive time-series plots.
* Aggregation by day/week/month.

### **4.3 Model Insights**

* Feature importance visualization.
* Model assumptions documentation.



## ****5. Outputs & Visualizations****

| **File** | **Description** |
| --- | --- |
| cleaned\_sales\_data.csv | Processed dataset with features. |
| model.pkl | Trained model. |
| actual\_vs\_predicted.png | Scatter plot of model performance. |
| feature\_importance.png | Bar plot of top coefficients. |

## ****Conclusion****

The AI-Based Sales Forecasting Dashboard provides:

* **Accurate predictions** (R² = 0.92).
* **Interpretable insights** via feature coefficients.
* **User-friendly interface** for business teams.

**Future Work**:

* Integrate real sales data.
* Add more models (e.g., Random Forest for comparison).
* Deploy as a cloud service.