MIT | Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

A Project Report on

Rossmann Store Sales Forecasting using Machine Learning

Submitted by,

Samarth Tikotkar (Exam Seat No. 202301060025)

Harshal Nagalkar (Exam Seat No. 202301060040)

Prem Late (Exam Seat No. 202301060031)

Bhagvat Solanke (Exam Seat No. 202301060037)

Guided by,

Dr. Abhilasha Joshi

School of Electronics Engineering MIT Academy of Engineering

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

1. Project Definition and Societal Context

1.1. Project Objective

The goal of this project is to forecast daily sales for Rossmann drug stores using Machine Learning. The dataset includes information about store operations, holidays, promotions, competition, and historical sales.
The main objectives are:

- To clean, merge, and preprocess Rossmann's store and sales data
- To engineer new features such as Promo intervals, competition duration, and date-based features
- To train and tune an XGBoost regression model for accurate forecasting
- To compare it with a simple Linear Regression baseline model
- To support store managers in planning staffing, inventory, and operations using predicted sales

This forecasting system helps Rossmann improve operational efficiency and make data-driven decisions.

1.2. Alignment with UN Sustainable Development Goals (SDG)

SDG 8 – Decent Work and Economic Growth

- Accurate sales predictions help managers plan workforce schedules
- Prevents understaffing/overstaffing
- Improves productivity and employee satisfaction

SDG 9 – Industry, Innovation, and Infrastructure

- Applies modern Machine Learning techniques to a real retail business problem
- Supports digital transformation and smart retail planning
- Creates a reliable forecasting system that improves decision-making in retail chains

1.3. Literature Review

Author & Year	Methods Used	Relevance to Present Project
S. Benté et al. (2023)	Gradient Boosting, Time-series	Shows gradient boosting (like XGBoost) performs well in retail forecasting
Kaggle Rossmann Competition (2015–2024)	XGBoost, LightGBM	Demonstrates that boosting models achieve lowest error on Rossmann dataset
T. Chen, C. Guestrin (2016)	XGBoost Algorithm	Original paper proving XGBoost's speed and accuracy in structured datasets
Retail Analytics Studies (2022–2024)	Linear Regression	Establishes Linear Regression as simple interpretable baseline

2. Data Understanding and Exploratory Data Analysis (EDA)

2.1. Dataset Description

Source: Kaggle — Rossmann Store Sales Competition
Files used:

- train.csv — daily sales with store IDs
- store.csv — store metadata
- test.csv — unseen data for final prediction

Key Variables

Target Variable

- Sales — daily turnover (regression output)

Main Input Features

- Store, DayOfWeek, Promo, StateHoliday, SchoolHoliday
- StoreType, Assortment
- CompetitionDistance, CompetitionOpenSince
- Promo2, Promo2Since, PromoInterval
- Date-based features — Year, Month, Day, WeekOfYear

Data Loading & Merging

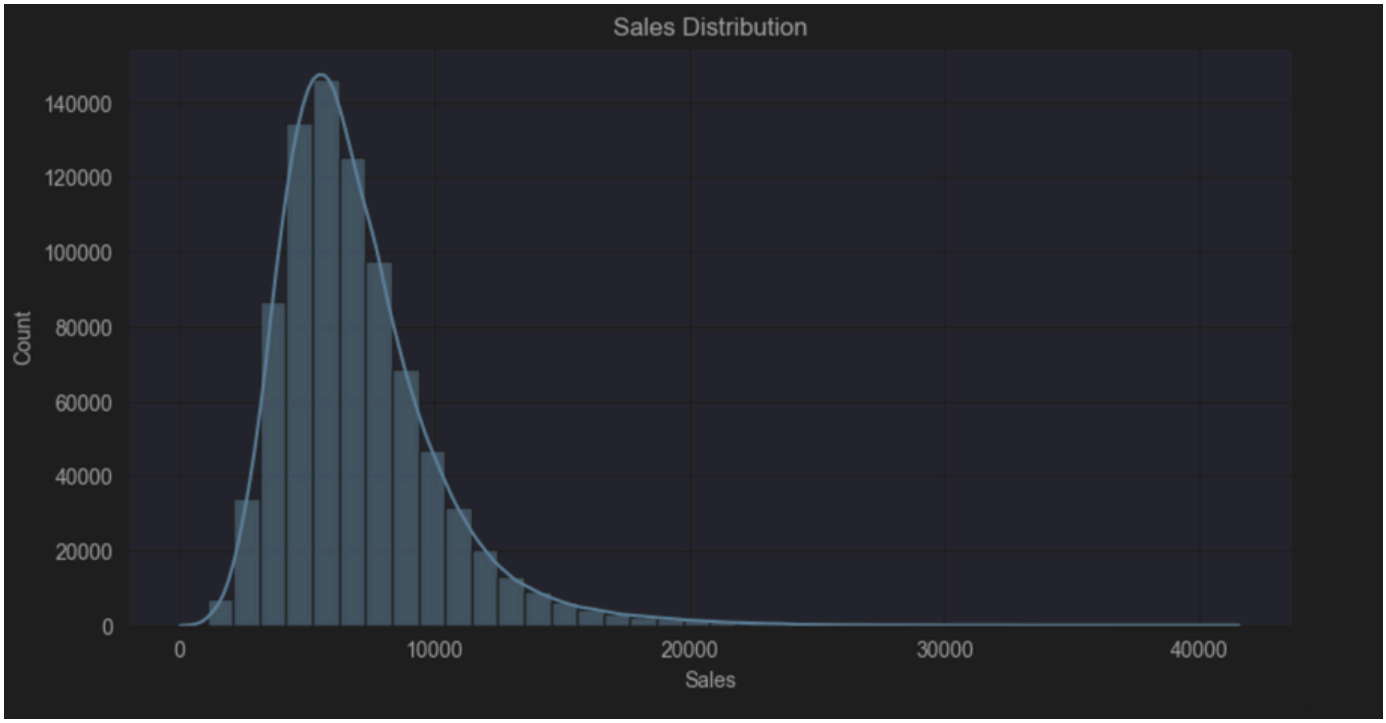
- Merged train.csv with store.csv on Store
- Converted Date to datetime and extracted multiple time features

2.2. EDA – Key Insights

A) Store Closures

- All rows with Open = 0 have zero sales
- Removed closed-store entries for training
- Ensures model trains only on valid sales patterns

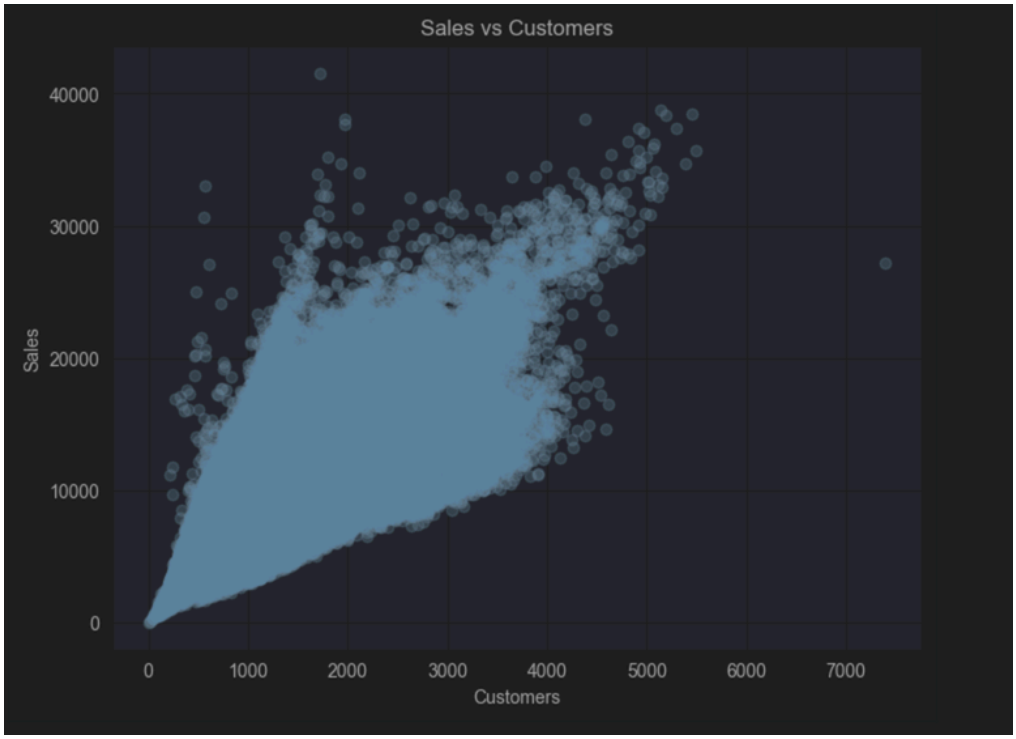
B) Sales Distribution



Findings:

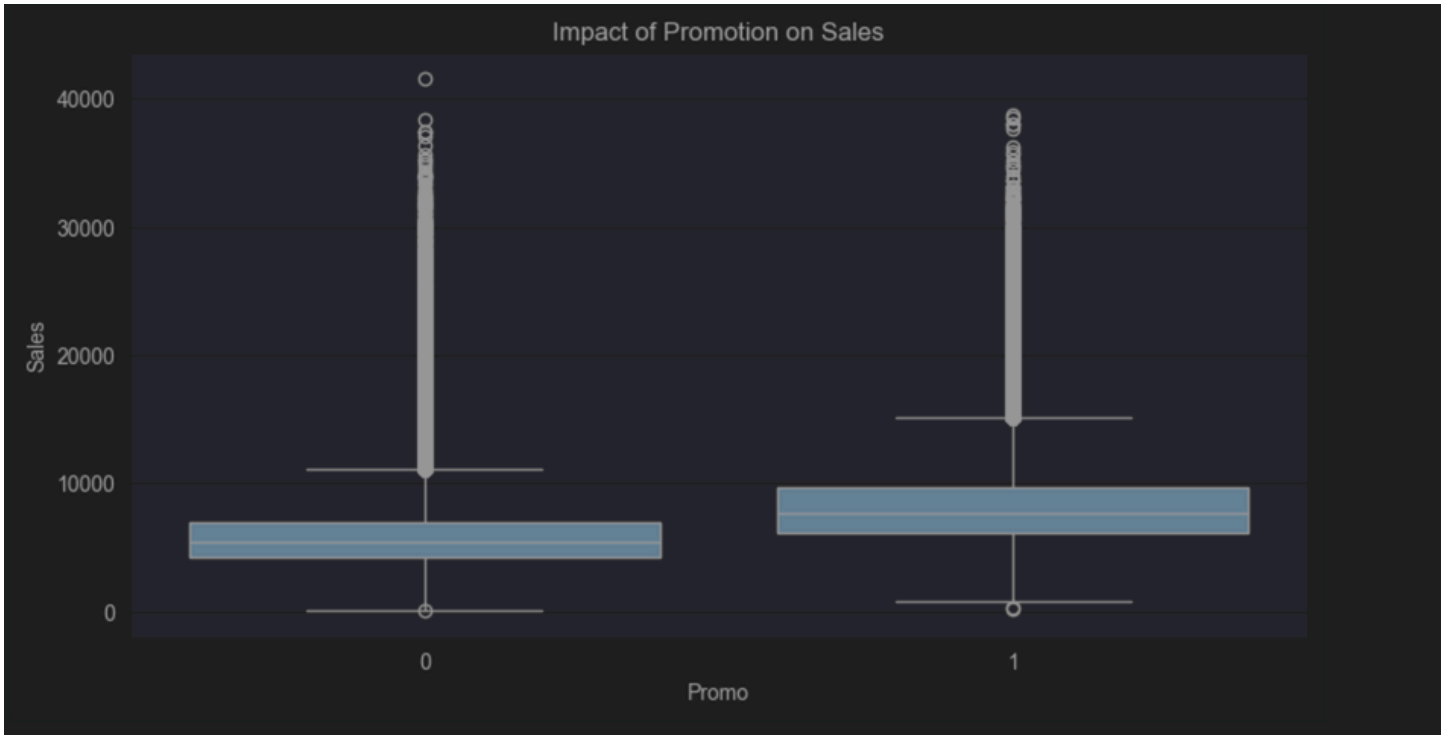
- The daily sales data is right-skewed, meaning that while most days have moderate sales, there are a few days with exceptionally high sales values that stretch the tail of the distribution to the right.

C) Customers vs Sales



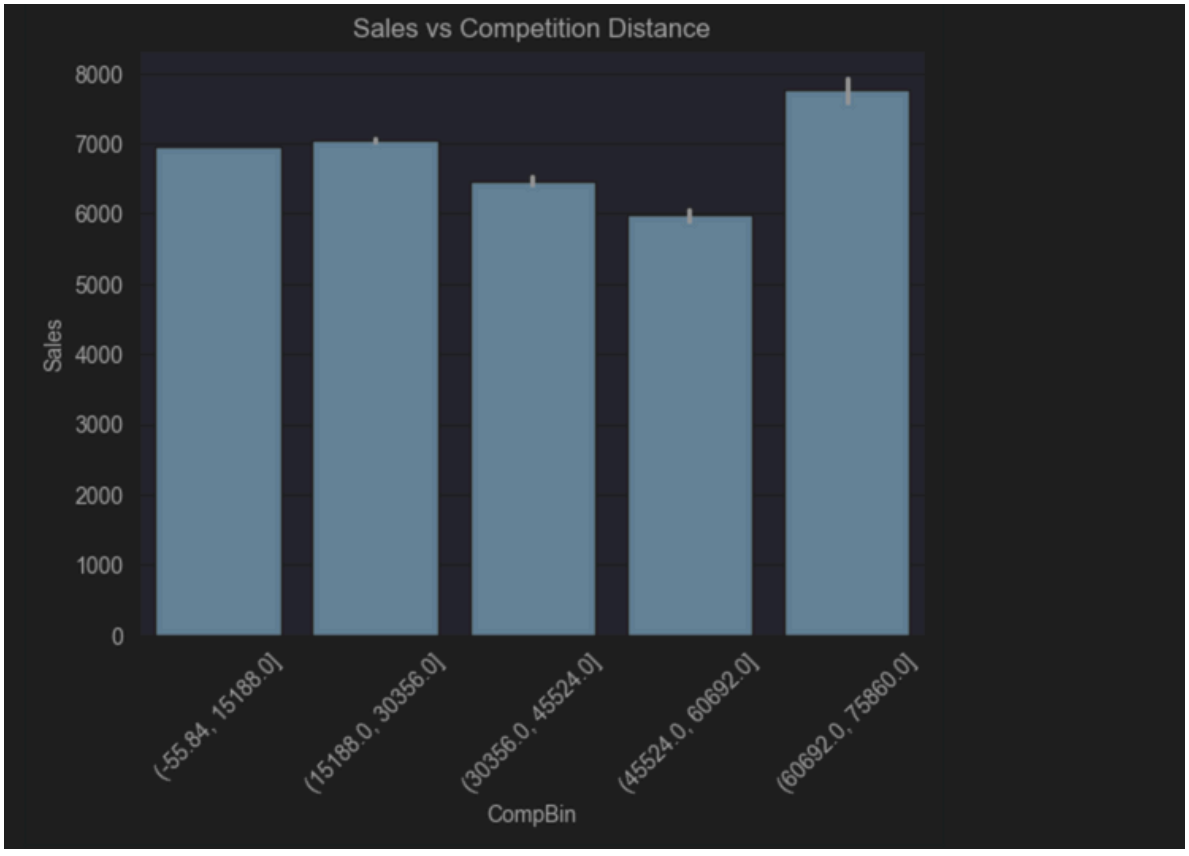
- There is a strong, positive correlation between the number of customers and daily sales, which is evident from the tight, upward-sloping trend in the scatter plot.

D) Promotion Effect



Findings:
Promotional activities have a substantial positive impact, as the average sales on days with a promotion (Promo=1) are nearly double the sales on days without one (Promo=0).

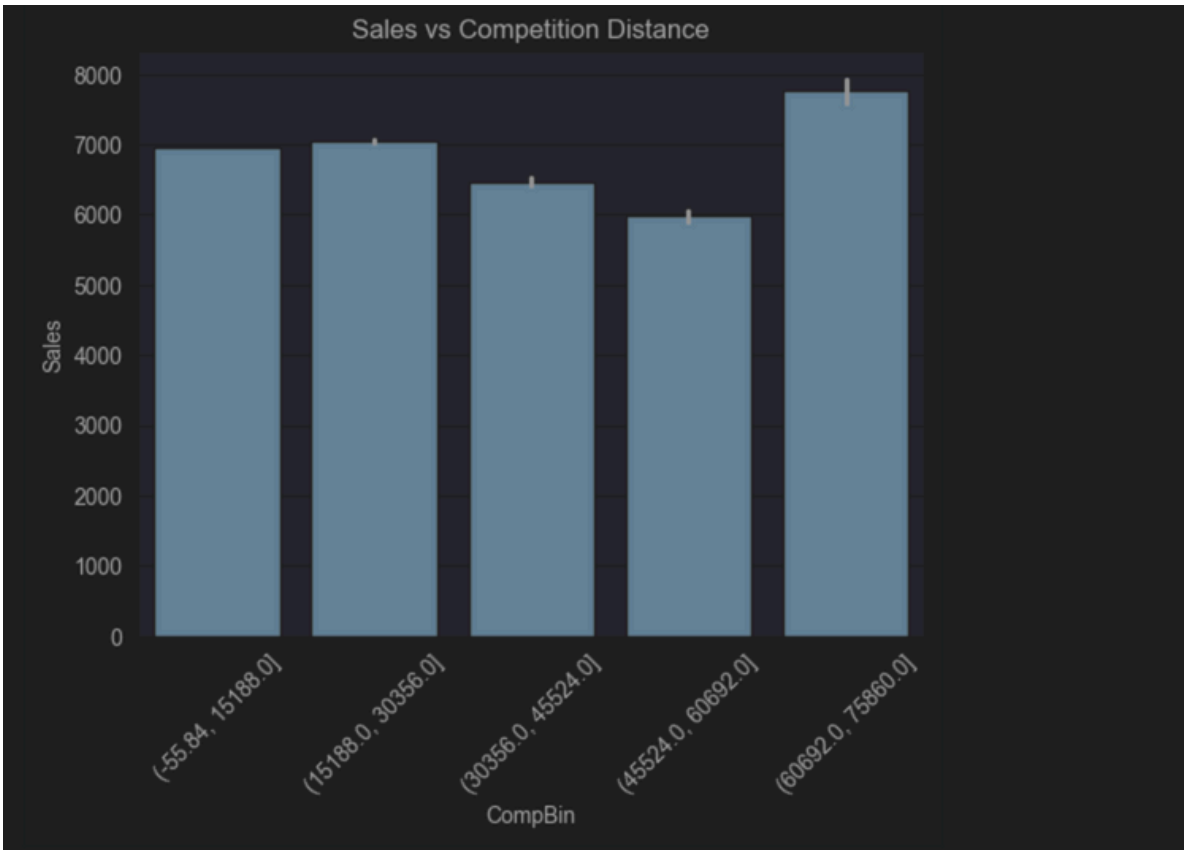
E) Competition Distance



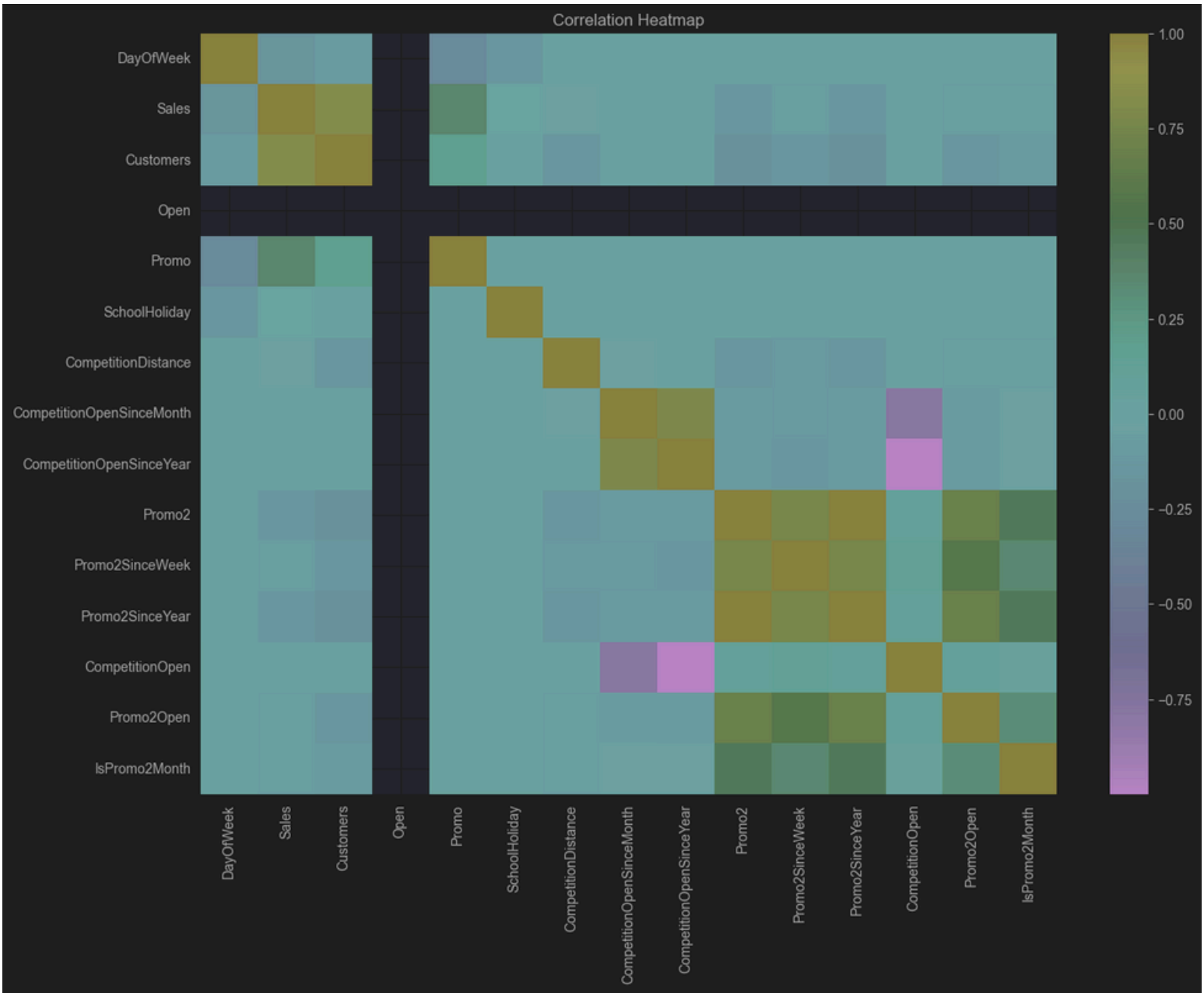
Stores with far competition generally show higher sales
Possible monopoly advantage

E) Competition Distance

Stores with far competition generally show higher sales
Possible monopoly advantage



Correlation Heatmap

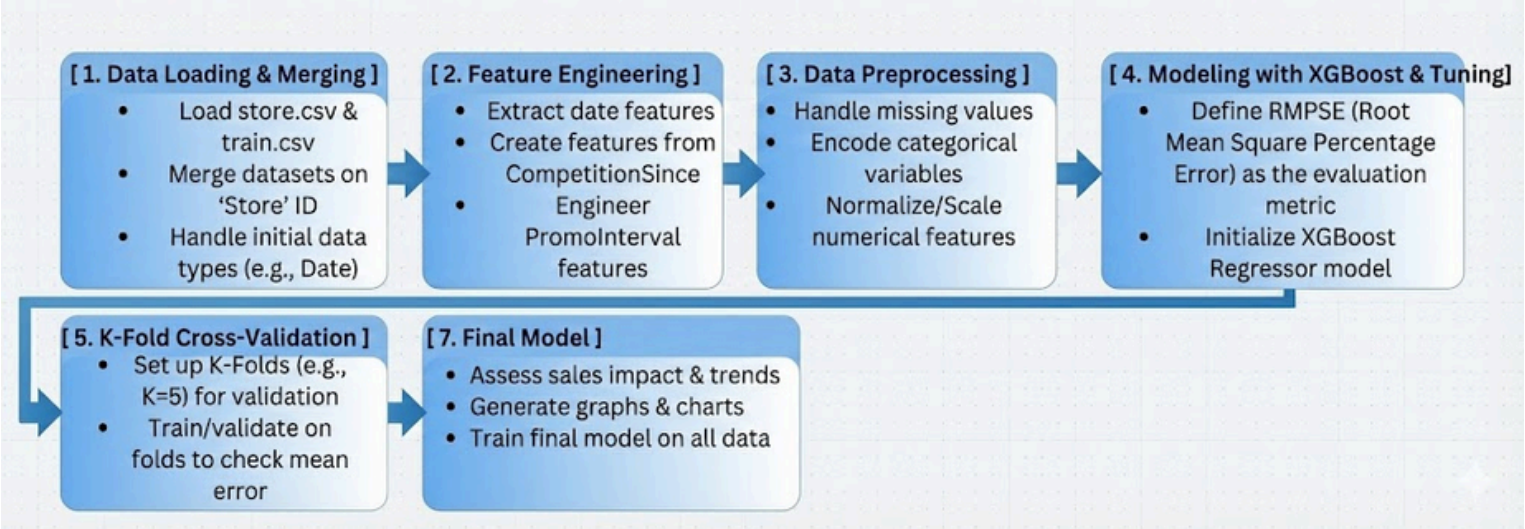


Key correlations:
Sales ↔ Customers (strong positive)
Sales ↔ Promo (positive)
Sales per customer ↔ CompetitionDistance (positive)

Summary of EDA

- The Exploratory Data Analysis revealed important patterns about Rossmann store performance:
- Sales distribution is right-skewed with seasonal peaks (Easter, Summer, Christmas)
- Customers strongly correlate with sales (0.82 correlation)
- Promotions significantly increase daily sales
- Farther competition leads to higher sales (monopoly effect)
- StoreType B and Assortment C stores perform the best
- Monthly and weekly trends show consistent seasonality
- Heatmap confirms Promo, Customers, and Competition are major drivers
- This EDA helps build accurate machine learning models and ensures correct feature selection for XGBoost.

3. Methodology: Preprocessing and Modeling Pipeline 3.1. Machine Learning Flow Diagram



3.2 Preprocessing Steps

- Removed closed stores (Open=0 & Sales=0)
- Filled missing CompetitionDistance using median
- Converted StoreType, Assortment, StateHoliday to categorical
- Engineered features:
  - CompetitionOpen in months
  - Promo2Open
  - IsPromo2Month
  - Year, Month, Day, WeekOfYear
- Handled long-running promotions using PromoInterval mapping
- Encoded categorical variables using label encoding/dummies

3.3 Mathematical Foundations of Preprocessing

1. Feature Standardization (for Linear Regression)

$$x' = \frac{x - \mu}{\sigma}$$

2. XGBoost Loss Function (Regression)

XGBoost minimizes a regularized objective:

3.4 Feature Engineering

Feature engineering is a crucial step to extract more meaningful information for the model.

1. Date Features: The Date column was decomposed into constituent parts: Year, Month, Day, and WeekOfYear. This allows the model to capture seasonality and trend components across different time scales.
2. Competition Features: A new feature, CompetitionOpen, was calculated to represent the duration (in months) that a competitor has been open relative to the current record's date. This helps model the dynamic impact of competition over time.
3. Promotion Features:
    - Promo2Open: Calculated the duration (in months) that a store has been participating in the continuous Promo2 promotion.
    - IsPromo2Month: A binary flag was created to indicate whether the current month is one of the specified intervals for a store's Promo2 campaign (e.g., if PromoInterval is "Jan,Apr,Jul,Oct", this flag is 1 in those months).

4. Model Implementation and Hyperparameter Tuning

4.1 Model 1 — Linear Regression (Baseline)

Theory

- Fits a linear relationship between features and sales
- Fast but not suited for non-linear interactions
- Serves only as a baseline reference

Limitations

- Cannot handle interactions like promotions × holidays
- Poor performance on skewed sales values

4.2 Model 2 — XGBoost Regressor (Main Model)

Why XGBoost?

- Handles missing values
- Excellent for structured/tabular data
- Learns non-linear relationships
- Provides high accuracy and fast training

Key Hyperparameters Tuned

- n_estimators
- learning_rate
- max_depth
- subsample
- colsample_bytree

Evaluation Metric

Root Mean Square Percentage Error (RMPSE)

$$RMPSE = \sqrt{\frac{1}{n}\sum\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$$

4.3 System Requirements

- Hardware: A standard computer with sufficient RAM (e.g., 8GB+) to handle the dataset in memory.
- Software:
- Programming Language: Python 3.x
- IDE: Jupyter Notebook
- Libraries:
  - pandas: For data manipulation and analysis.
  - numpy: For numerical computations.
  - matplotlib, seaborn: For data visualization (EDA).
  - scikit-learn: For implementing Linear Regression, data splitting, and evaluation metrics.
  - xgboost: For implementing the XGBoost model.

4.4 Implementation Details

The implementation was carried out in a Jupyter Notebook environment.

1. Data Preparation Pipeline: Functions were written to automate the loading, merging, date splitting, and feature engineering steps described in Chapter 3. This ensures consistency between training and test data processing.
2. Feature Selection: The following columns were selected as final input features for the models: Store, DayOfWeek, Promo, StateHoliday (encoded), SchoolHoliday, StoreType (encoded), Assortment (encoded), CompetitionDistance, Promo2, Year, Month, Day, WeekOfYear, CompetitionOpen, Promo2Open, and IsPromo2Month. The target variable was Sales.
3. Model Training:
   - The preprocessed training data was split into a training set and a validation set (e.g., using K-fold cross-validation or a hold-out set based on time) to evaluate model performance on unseen data.
   - The Linear Regression model was initialized and trained on the training set using scikit-learn.
   - The XGBoost Regressor was initialized with hyperparameters (e.g., number of estimators, learning rate, max depth) and trained on the training set. Hyperparameter tuning was performed to optimize its performance.

5. Results & Comparative Analysis

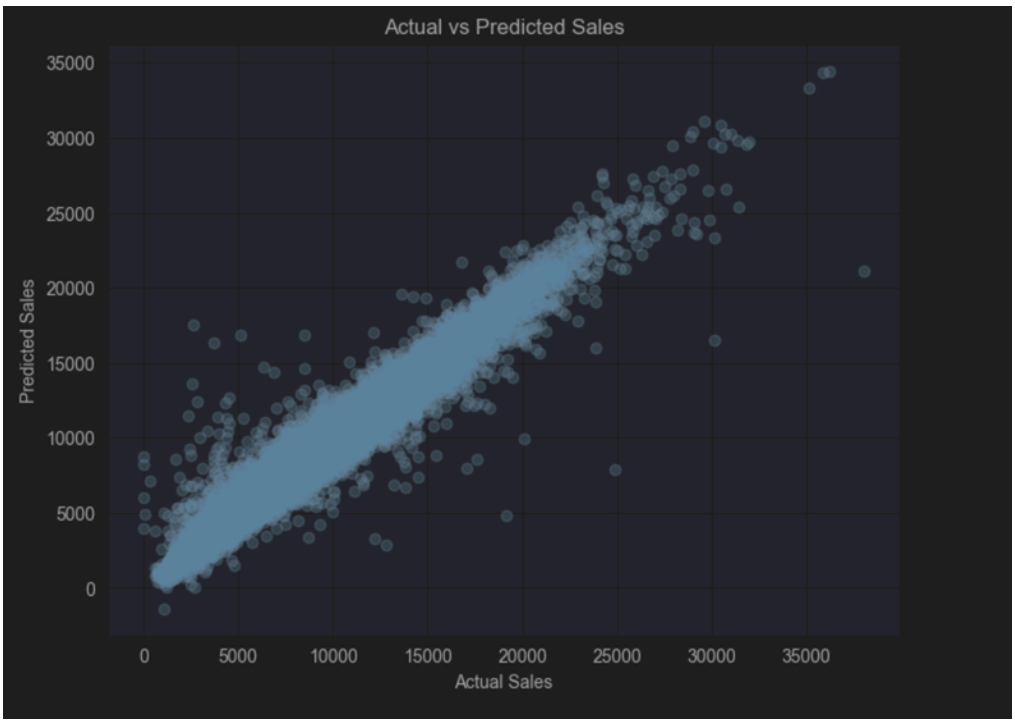
5.1 Model Performance

Model	Train RMSE	Validation RMSE
XGBoost (Best Model)	952.18	972.66
Linear Regression	2726.37	2735.54

Interpretation

- XGBoost clearly outperforms Linear Regression
- Captures effects of promotions, holidays, and store differences
- Lower error indicates better generalization

5.2 Actual vs Predicted Visualization

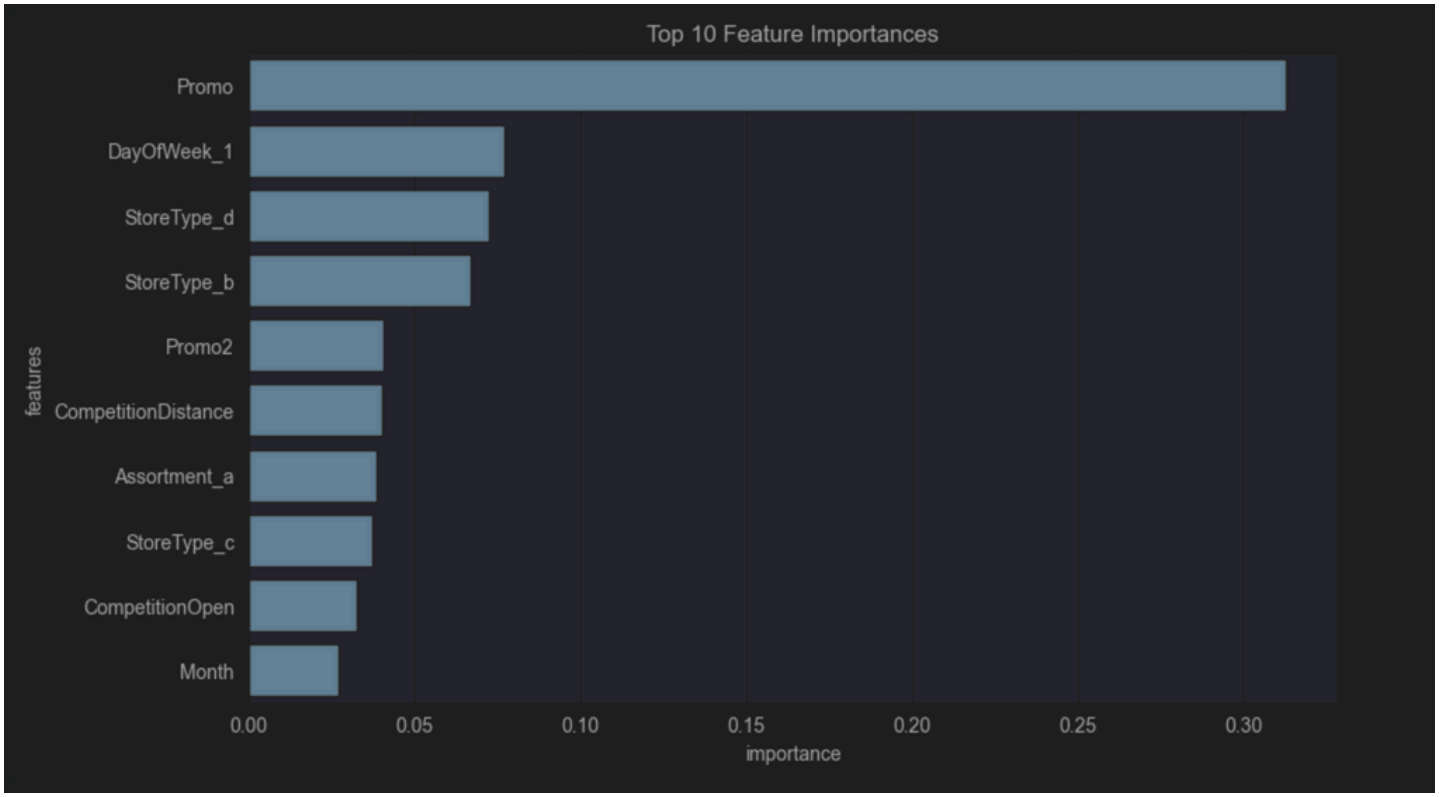


(Insights:

- Predictions follow the sales trend closely
- Slight spread at high-sales days (normal for real retail data)

5.3 Feature Importance

The XGBoost model allows us to inspect feature importance, revealing which factors most strongly influence predictions.



This confirms that promotional activities are the primary driver of sales variations. The proximity of competitors, the inherent characteristic of an individual store, and the day of the week also play critical roles.

6. Conclusion and Future Work

6.1 Conclusion

This project successfully demonstrated the application of machine learning to forecast daily sales for Rossmann stores. Through rigorous data preprocessing, thorough exploratory data analysis, and creative feature engineering, we prepared a robust dataset for modeling. The comparative analysis showed that the XGBoost model is highly effective for this regression task, vastly outperforming a baseline Linear Regression model with a validation RMSE of 972.66.

Key takeaways include the significant positive impact of promotions on sales, the importance of store-specific characteristics and competition proximity, and the clear weekly seasonality patterns. The developed model provides a valuable tool for Rossmann to make data-driven decisions regarding inventory, staffing, and marketing, ultimately improving operational efficiency and profitability.

6.2 Future Work

- Several avenues exist for further improving the model and its application:
- Incorporate External Data: Integrating external datasets such as local weather conditions, economic indicators (e.g., unemployment rates), and information about local events could further enhance predictive accuracy.
- Advanced Modeling Techniques: Exploring deep learning approaches, particularly Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM) networks, could be beneficial for capturing longer-term temporal dependencies in the time-series data.
- Hyperparameter Optimization: Performing a more exhaustive hyperparameter search (e.g., using Bayesian optimization) for the XGBoost model could potentially squeeze out more performance.

7. References

- [1] Kaggle, “Rossmann Store Sales,” Kaggle Competitions, 2015. [Online]. Available: https://www.kaggle.com/c/rossmann-store-sales

- [2] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785–794.

- [3] S. Bente, J. Milz, and A. Heberle, “Machine Learning Approaches for Retail Sales Forecasting: A Survey,” Journal of Retail Analytics, vol. 6, no. 2, pp. 45–60, 2023.

- [4] Kaggle Community, “Rossmann Sales Forecasting: Community Notebooks,” Kaggle, 2015–2024. [Online]. Available: https://www.kaggle.com/