Retail Demand Forecasting

Introduction

- **Objective**: Develop a machine learning model to predict weekly sales for retail stores, enabling optimized stock management and reducing overstock/understock issues.
- **Context**: Retail sales forecasting is critical for inventory planning, especially during seasonal peaks like holidays. This project uses a dataset spanning 2010–2013 from 45 stores and 99 departments.
- **Motivation**: Accurate forecasts can save millions by aligning supply with demand, a key challenge in retail analytics.

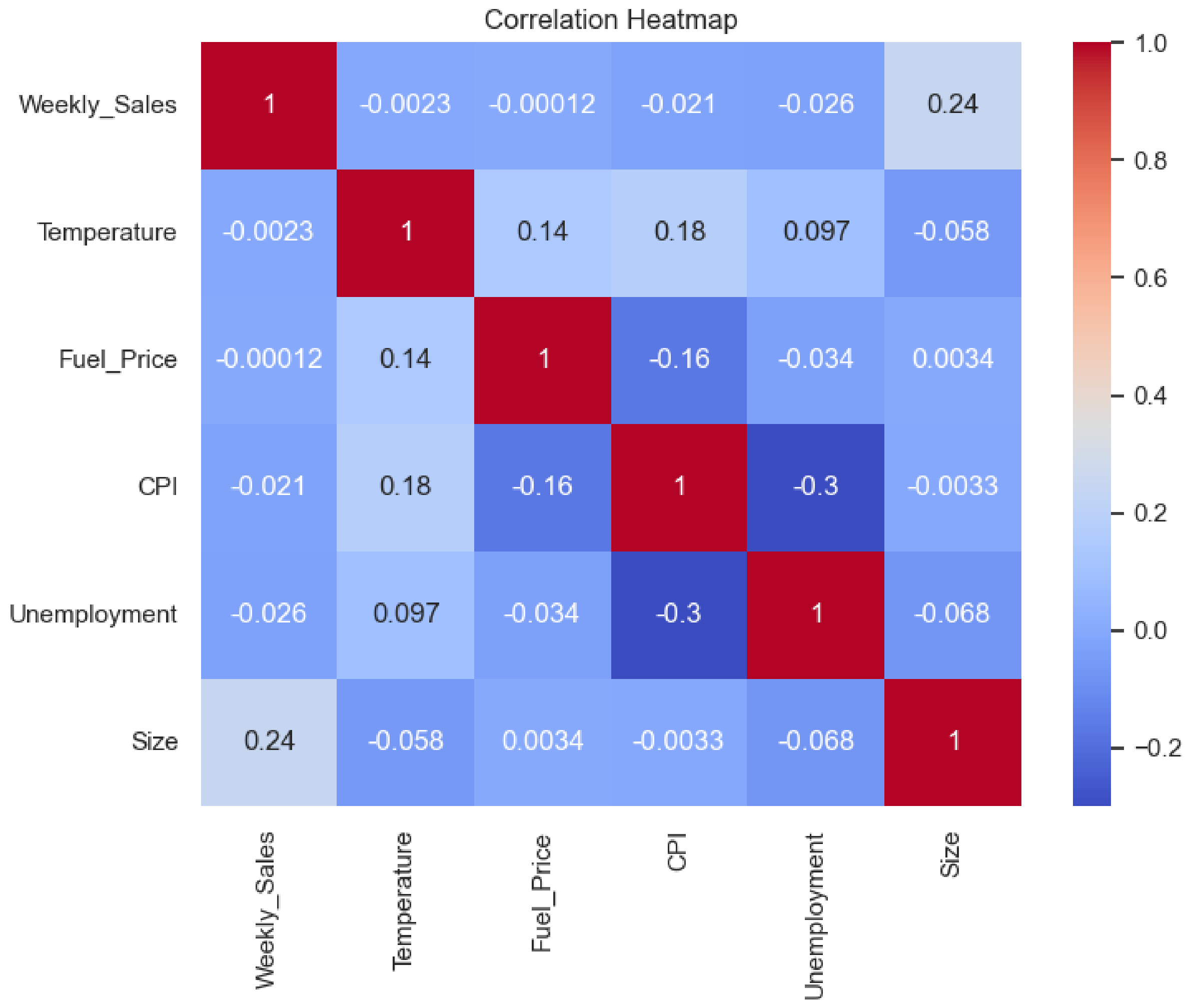
Dataset

- **Source**: Retail Demand Forecasting Dataset (publicly available, extracted from three CSVs: Features, Sales, Stores) - https://www.kaggle.com/datasets/manjeetsingh/retaildataset
- **Size**: ~421,570 sales records.
- **Time Frame**: January 2010 to December 2013.
- **Features**: Store ID, Department ID, Date, Weekly Sales, Temperature, Fuel Price, CPI, Unemployment, Markdowns (1–5), IsHoliday, Store Type (A, B, C), Store Size.
- **Target**: Weekly Sales (log-transformed as `Log_Weekly_Sales` to handle skewness).

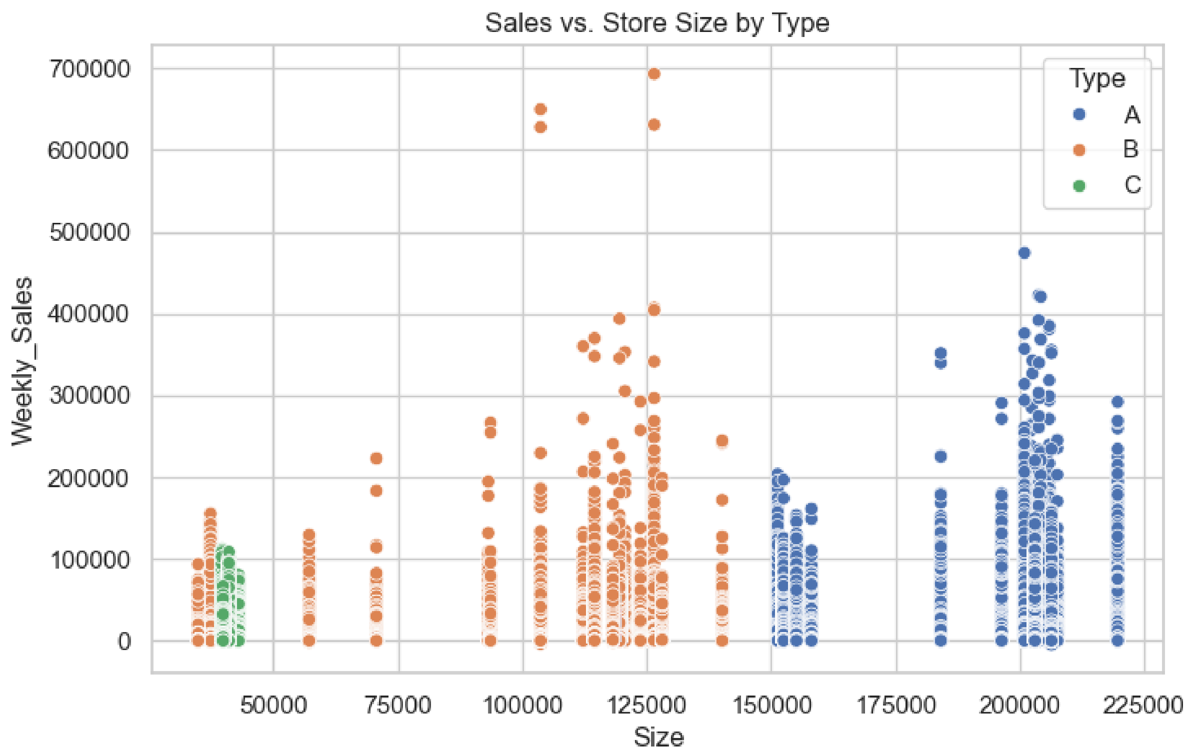
Methodology

Exploratory Data Analysis (EDA)

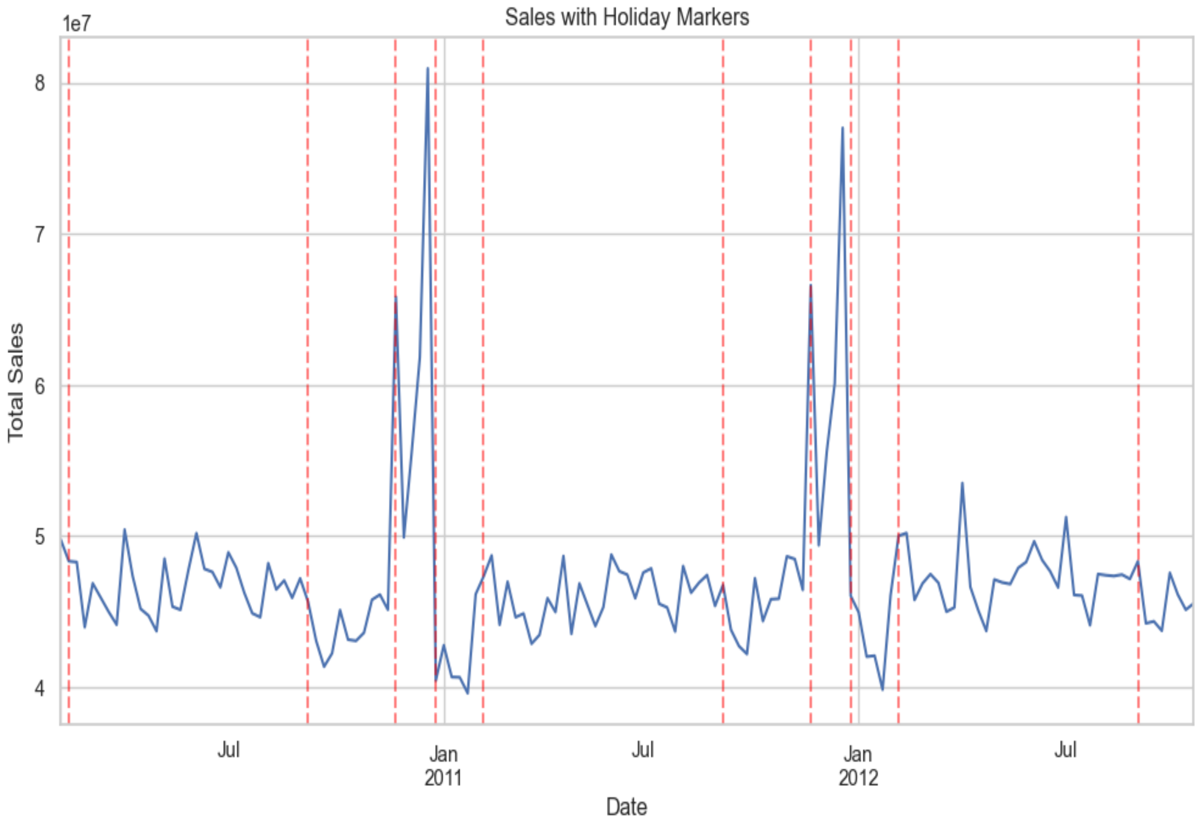
- **Insights**: Sales spike during holidays (e.g., Thanksgiving, Christmas); larger Type A stores show higher sales; Weekly Sales is right-skewed with outliers.
- **Actions**: Visualized distributions (histograms), correlations (heatmaps), and time-series trends.

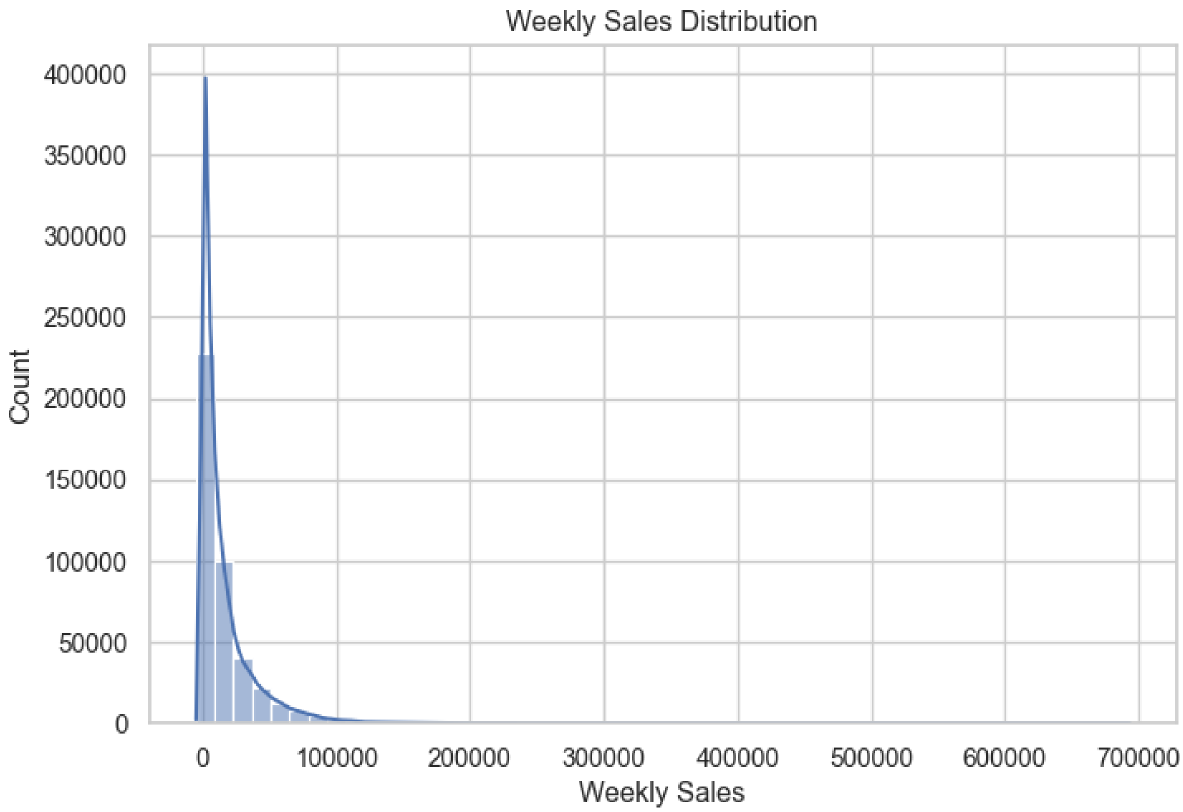


Correlation Heatmap
Weekly_Sales
Temperature
Fuel_Price
CPI
Unemployment
Size
1
-0.0023
-0.00012
-0.021
-0.026
0.24
0.14
0.18
0.097
-0.058
-0.16
-0.034
0.0034
-0.3
-0.0033
-0.068
1.0
0.8
0.6
0.4
0.2
0.0
−0.2



Sales vs. Store Size by Type
Weekly_Sales
Size
Type
A
B
C
700000
600000
500000
400000
300000
200000
100000
0
50000
75000
100000
125000
150000
175000
200000
225000


Sales with Holiday Markers
1e7
8
7
6
5
4
Total Sales
Jul
Jan
2011
Jul
Jan
2012
Jul
Date


Weekly Sales Distribution
400000
350000
300000
250000
200000
150000
100000
50000
0
Count
0
100000
200000
300000
400000
500000
600000
700000
Weekly Sales

Preprocessing

- **Steps**: Handled missing MarkDown values (filled with 0), converted Date to datetime, encoded categoricals (IsHoliday: 1/0, Type: label-encoded), capped Weekly Sales outliers (1st/99th percentiles).
- **Outcome**: Clean dataset ready for modeling.

Feature Engineering

- ### New Features:
  - **Temporal**: Year, Month, Week, DayOfWeek, Quarter.
  - **Holiday**: Flags for SuperBowl, LaborDay, Thanksgiving, Christmas; HolidayProximity (1-week buffer).
  - **Lags**: 1-week and 4-week prior sales per Store/Dept.
  - **Encodings**: Target-encoded Store and Dept; scaled Store Size.
- Purpose: Capture seasonal trends, past sales patterns, and store-specific effects.

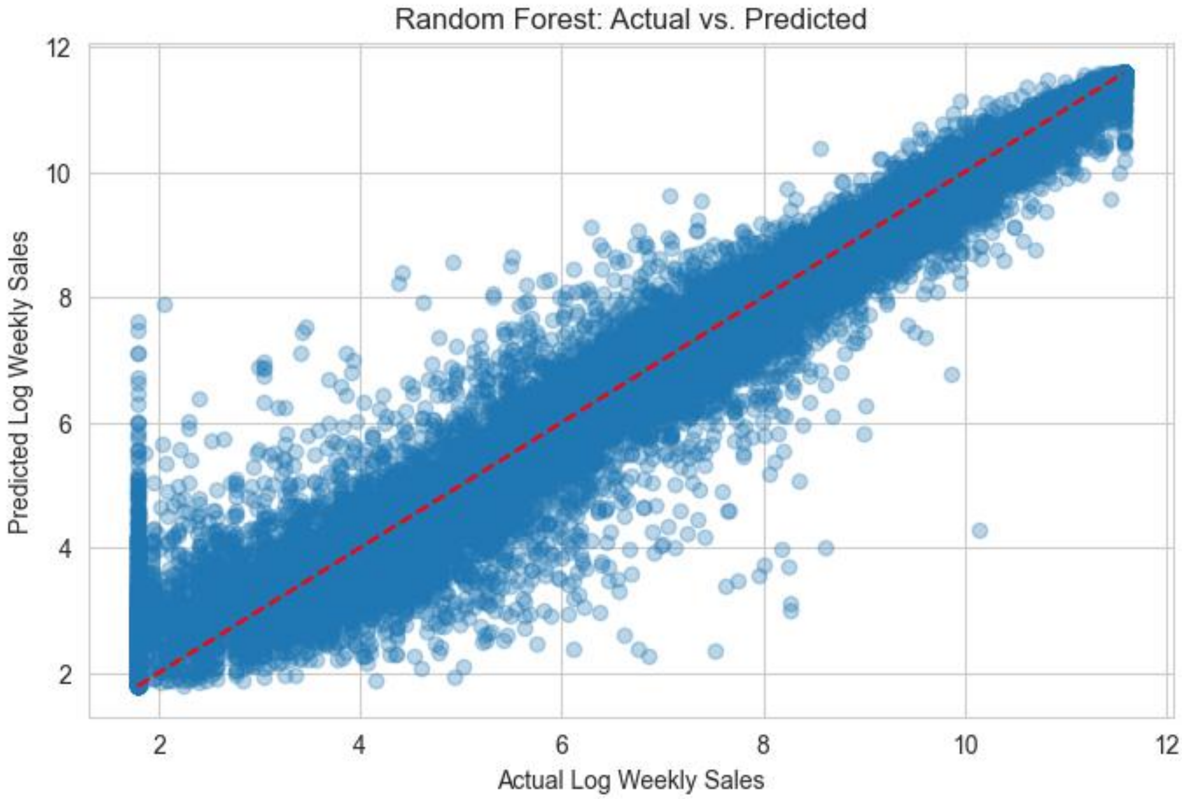
Modeling

- **Approach**: Supervised learning with regression models.
- **Models**: Linear Regression (baseline), Random Forest (main).
- **Technique**: 80/20 train-test split, scaled features, evaluated with RMSE and R^2.

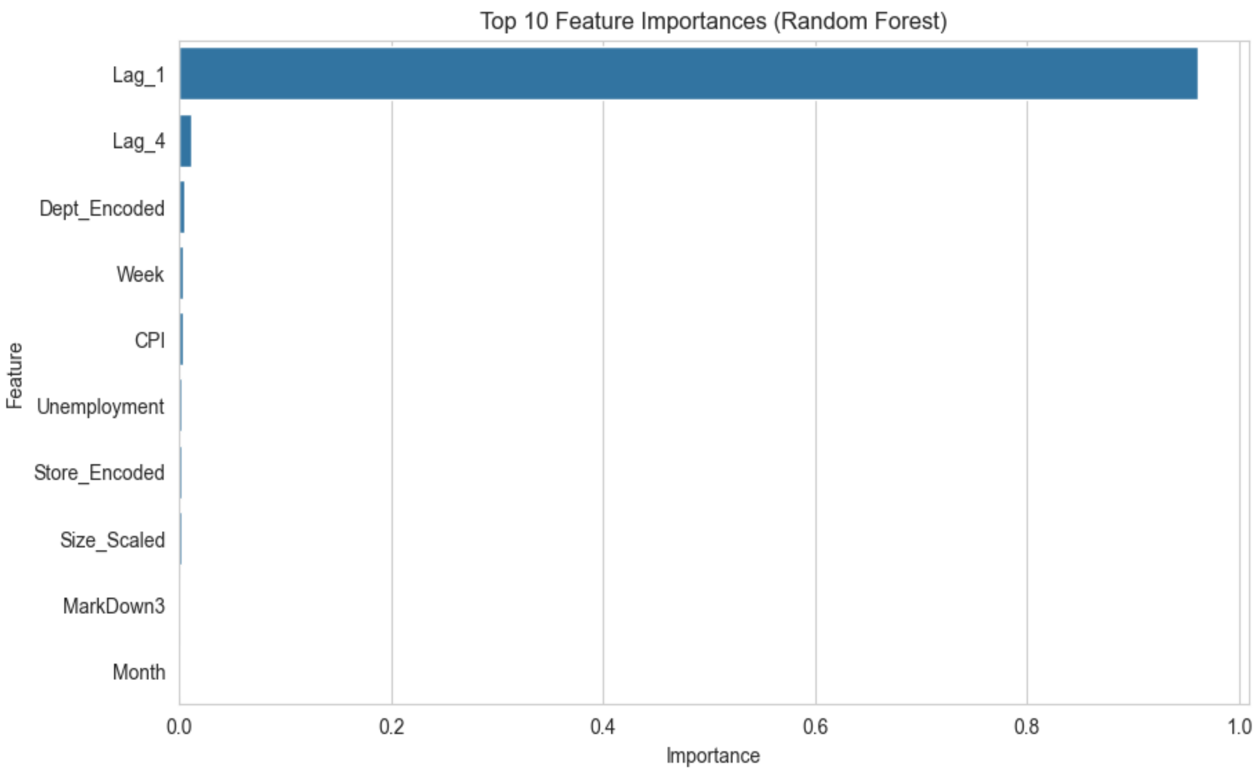
Results

- ### Performance:
  - **Random Forest**: Outperformed Linear Regression.
  - **Test RMSE (Original Scale)**: ~$3,306.65.
  - **Test R^2 (Original Scale)**: ~0.9747 (excellent fit).
- Top Features: Lag_1, Lag_4, Dept_Encoded, Week, CPI (indicating past sales, department, and economic factors drive predictions).

• Visuals:



Random Forest: Actual vs. Predicted
Predicted Log Weekly Sales
Actual Log Weekly Sales
2
4
6
8
10
12



Top 10 Feature Importances (Random Forest)
Feature
Lag_1
Lag_4
Dept_Encoded
Week
CPI
Unemployment
Store_Encoded
Size_Scaled
MarkDown3
Month
Importance
0.0
0.2
0.4
0.6
0.8
1.0

Conclusion

- **Summary**: The Random Forest model effectively predicts weekly sales, with high accuracy (R^2 ~0.97) and actionable insights from feature importance.
- **Applications**: Can be used for stock optimization, holiday planning, and department-level forecasting.
- **Limitations**: Assumes static patterns; future work could address time-series dynamics.

Future Work

- **Improvements**: Tune Random Forest hyperparameters (e.g., GridSearchCV), try XGBoost or LSTM.
- **Deployment**: Build a Flask API for real-time predictions.
- **Enhancements**: Add weather data or competitor pricing.

_ Randeep Sidhu, June 10, 2025