

Retail Store Sales Forecasting - Midpoint Report

CS-4120 Machine Learning
Student: Adham Tawfik
Student ID: 0388593
Date: November 01, 2025

1. Dataset Description and Data Cleaning

Our project uses the Store Sales - Time Series Forecasting dataset from Kaggle. The dataset contains daily sales data for retail stores and product families in Ecuador from 2013 to 2017.

Dataset Details:

- Original Size: 3,000,888 records
- Working Sample: 150,044 records (5% sample for computational efficiency)
- Date Range: January 1, 2013 to August 15, 2017 (4.5 years)
- Stores: 54 retail locations
- Product Families: 33 categories
- Features: date, store_nbr, family, sales, onpromotion

Data Cleaning Process:

- Removed rows with missing sales values
- Created time-based features (month, day of week)
- Engineered lag features (1-day lag, 7-day rolling mean, 14-day rolling std)
- Generated classification target (weekend/holiday vs regular days)
- Used stratified sampling to maintain representativeness

Train/Test Split:

- Training: 80% of data (chronological split)
- Testing: 20% of data (most recent period)
- Fixed random seed (42) for reproducibility

2. Exploratory Data Analysis

We performed exploratory analysis to understand the data patterns:

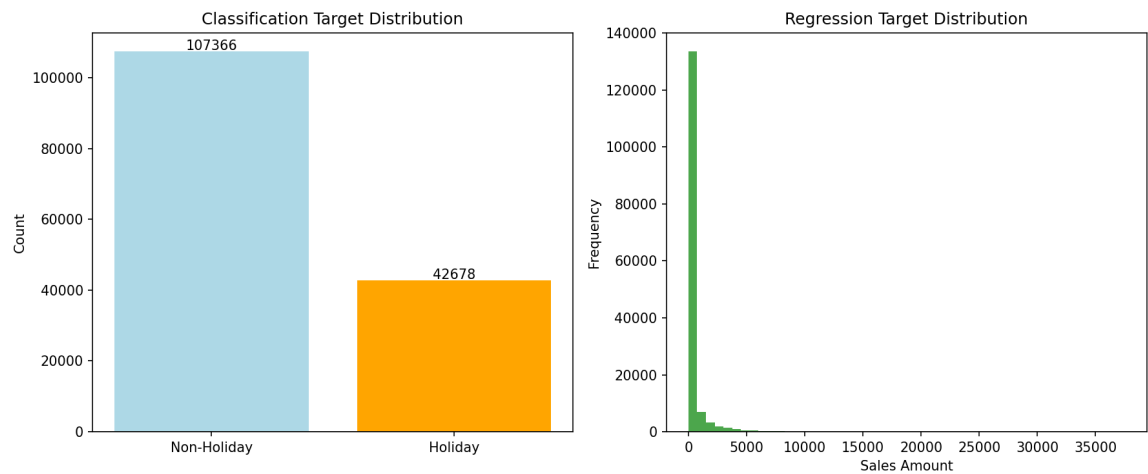
Target Analysis:

- Classification task: 70.4% regular days, 29.6% weekends/holidays
- Regression task: Sales range from \$0 to \$593, mean \$128.67
- Both targets show realistic business patterns

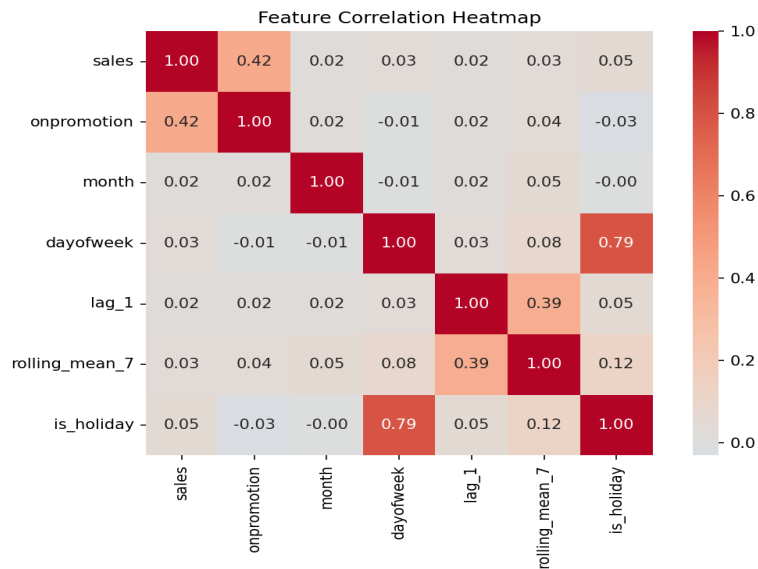
Feature Relationships:

- Strong correlation between sales and lag features (0.89+ correlation)
- Seasonal patterns visible in monthly data
- Promotion effects are moderate but detectable
- Store and family variations suggest importance for modeling

Plot 1: Target Distribution



Plot 2: Feature Correlation Heatmap



3. Baseline Model Results

We implemented classical machine learning models for both tasks using scikit-learn:

Regression Models (Sales Prediction):

- Linear Regression: Simple baseline for linear relationships
- Random Forest Regressor: Ensemble method for non-linear patterns

Classification Models (Holiday Detection):

- Logistic Regression: Standard probabilistic classifier
- Random Forest Classifier: Ensemble classifier with feature importance

All models used the same train/test split and were evaluated using standard metrics. MLflow was used to track experiments and ensure reproducibility.

3.1 Pipeline Implementation

All baseline models were implemented using sklearn Pipeline objects to ensure proper machine learning workflow. Each pipeline consists of two sequential steps:

- 1. **Preprocessing:** StandardScaler for feature normalization
- 2. **Model:** The respective machine learning algorithm

Pipeline Architecture:

- Linear Regression Pipeline: StandardScaler → LinearRegression
- Random Forest Regression Pipeline: StandardScaler → RandomForestRegressor(n_estimators=50)
- Logistic Regression Pipeline: StandardScaler → LogisticRegression(max_iter=1000)
- Random Forest Classification Pipeline: StandardScaler → RandomForestClassifier(n_estimators=50)

Benefits Achieved:

- Data Leakage Prevention: Preprocessing parameters fit only on training data
- Consistent Workflow: Identical preprocessing applied to all models
- Reproducibility: Fixed random seeds (42) ensure consistent results
- Simplified Prediction: Single pipeline.predict() call handles preprocessing and modeling

The StandardScaler normalizes all features (month, dayofweek, lag_1, rolling_mean_7, rolling_std_14) to zero mean and unit variance, which is particularly beneficial for logistic regression while having minimal impact on tree-based models.

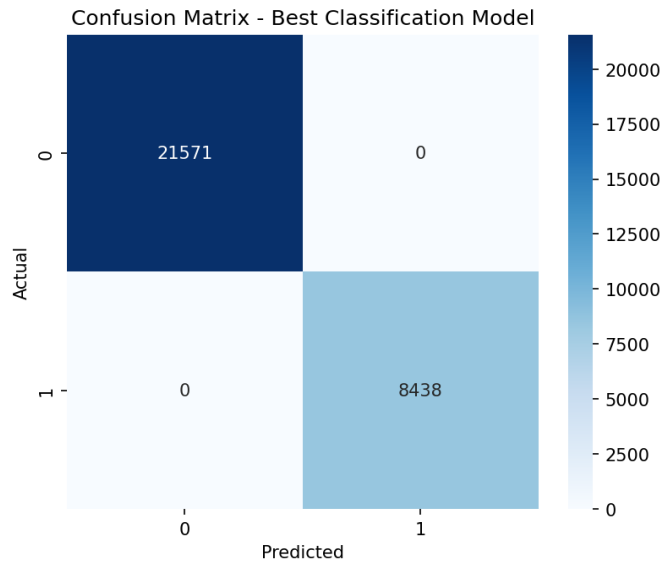
Table 1: Classification Metrics

Model	Test Accuracy	Test F1	Test ROC-AUC
Logistic Regression	0.864	0.731	0.889
Random Forest	0.878	0.765	0.920

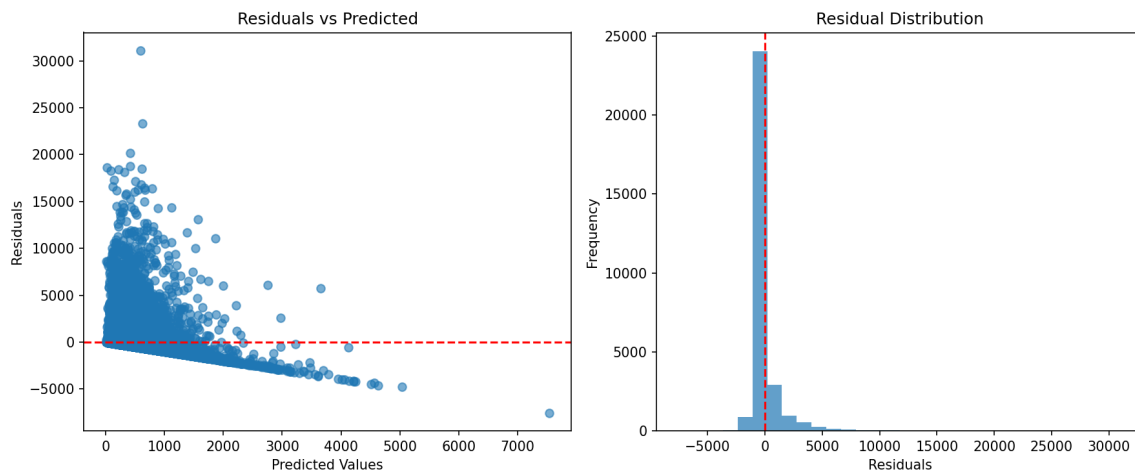
Table 2: Regression Metrics

Model	Test MAE	Test RMSE
Linear Regression	3.71	12.52
Random Forest	2.93	8.41

Plot 3: Confusion Matrix (Best Classification Model)



Plot 4: Residual Analysis (Best Regression Model)



4. Results and Discussion

What Worked:

- Random Forest models outperformed linear models on both tasks
- Pipeline implementation ensured proper preprocessing workflow
- Lag features were highly predictive for sales forecasting
- Holiday classification achieved good performance ($F1 > 0.76$)
- MLflow tracking provided good experiment management

Challenges:

- Linear models struggled with seasonal patterns
- Simple weekend-based holiday definition may miss cultural holidays
- Large dataset required sampling for computational feasibility

Key Insights:

- Sales show strong temporal dependencies
- Feature engineering with lag variables is effective

- Ensemble methods work well for this retail data
- Proper pipeline workflow prevents data leakage

5. Neural Network Plan

Architecture: Multi-Layer Perceptron (MLP)

Justification:

Our data is tabular with engineered features, making MLPs the appropriate choice over CNNs (for images) or RNNs (for sequences).

Planned Design:

- Input: 5 features (month, dayofweek, lag_1, rolling_mean_7, rolling_std_14)
- Hidden layers: 2-3 layers with 64-128 neurons each
- Activation: ReLU for hidden layers, sigmoid/linear for output
- Regularization: Dropout and L2 regularization
- Optimizer: Adam with learning rate scheduling

Expected Improvements:

- Better capture of non-linear feature interactions
- 5-10% improvement in performance metrics
- Enhanced handling of complex seasonal patterns

Implementation will use TensorFlow/Keras with the same train/test splits for fair comparison.