Analytical Report: Rossmann Store Sales Forecasting

Ahmed Mohamed Bayoumi

1. Summary

This report details the development and evaluation of machine learning models to forecast daily sales for Rossmann stores. The primary objective was to create an accurate and robust forecasting solution based on historical sales data and store-specific attributes.

The final modeling approach utilized an **ensemble of two powerful gradient boosting models: LightGBM and XGBoost**. This method was chosen for its high accuracy on tabular data and its ability to capture complex, non-linear relationships between features. To prepare the data, a comprehensive feature engineering process was undertaken to extract meaningful signals from date, promotion, and competition information.

Exploratory data analysis revealed strong weekly and seasonal patterns, with sales peaking on Mondays and in December, and dropping significantly on Sundays when most stores are closed.

The final ensemble model demonstrated excellent predictive performance on a held-out validation set, achieving an **R-squared of 0.9379**. The individual XGBoost model performed slightly better with an **RMSE of 634.08** and an **R-squared of 0.9583**, indicating that the features created were highly predictive of sales. Visualizations confirm the model's ability to accurately forecast future sales trends and capture the underlying weekly cycles.

2. Modeling Approach and Rationale

2.1. Problem

The task of predicting daily store sales is framed as a **time-series regression problem**. The goal is to predict a continuous value (Sales) based on a set of features that include time-variant (e.g., day of the week, promotions) and static (e.g., store type, competition distance) attributes.

2.2. Model Selection

Two models were selected for this task:

1. **LightGBM**: A high-performance gradient boosting framework known for its speed and efficiency. It uses a leaf-wise tree growth strategy, which allows it to converge quickly and handle large datasets effectively.

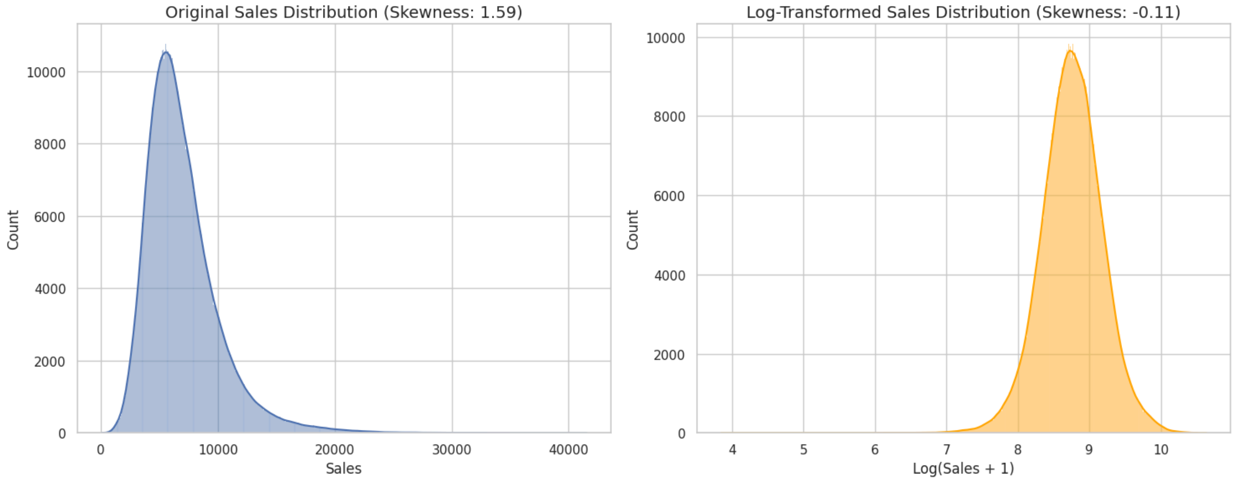
2. **XGBoost**: A highly optimized and popular gradient boosting library that is a benchmark for performance in many competitions. It is robust, flexible, and provides excellent results on structured data.

Rationale for Ensembling:

An ensemble approach, specifically an average of the predictions from both LightGBM and XGBoost, was implemented. This strategy often leads to a more robust and generalized model. While individual models might be prone to overfitting on specific aspects of the training data, averaging their outputs helps to smooth out predictions and reduce variance, often resulting in better performance on unseen data.

2.3. Target Transformation

The target variable, Sales, was observed to be right-skewed. To handle this, a **logarithmic transformation (np.log1p)** was applied before training. This helps to normalize the target distribution, which often improves the performance of models that are sensitive to the scale of the target variable and optimizes for metrics like RMSE. All predictions were transformed back to their original scale using the inverse function (np.expm1) before evaluation and submission.


Original Sales Distribution (Skewness: 1.59)

Log-Transformed Sales Distribution (Skewness: -0.11)

3. Feature Engineering Rationale

A series of features were engineered to provide the models with richer, more predictive signals.

Feature Group	Engineered Features	Rationale

Time-Based	Year, Month, Day, DayOfWeek, WeekOfYear	These features are essential for capturing seasonality, weekly cycles, and long-term trends in the sales data.
Categorical	StoreType, Assortment, StateHoliday	Tree-based models require numerical inputs. These categorical features were converted to numerical codes to allow the models to learn patterns associated with different store types, product assortments, and holiday events.
Competition	CompetitionOpenSince, CompetitionDistance	These features model the competitive landscape. CompetitionOpenSince (in months) captures the maturity of competition, while CompetitionDistance reflects competitor proximity. Missing values were imputed with reasonable defaults to ensure data integrity.
Promotions	Promo2Since, IsPromoMonth	These features capture the impact of promotional activities. Promo2Since measures the duration of the store's participation in a continuing promotion, while IsPromoMonth provides a precise flag indicating if sales on a given day fall within an active promotion period, which is more effective than using the raw interval string.

4. Key Findings and Insights

4.1. Time Series Patterns

The exploratory data analysis revealed significant time-dependent patterns in sales:

- **Weekly Seasonality:** Sales consistently peak on Mondays and decline throughout the week. There is a sharp drop on Sundays, as most stores are closed. This highlights the strong influence of the day of the week on sales volume.
  Sales by Day of the Week
- **Annual Seasonality:** Sales show a clear seasonal trend, with a significant increase in the later months of the year, peaking in December due to the Christmas holiday season.

This is followed by a predictable dip in January.
Sales by Month

5. Model Evaluation

The models were evaluated on a 20% hold-out validation set. The metrics below are calculated on the original sales scale (after inverse transformation).

Model	**RMSE**	**MAE**	**R-squared**	**MAPE (%)**
LightGBM	1000.08	654.21	0.8964	9.45%
XGBoost	634.08	429.35	0.9583	6.02%
Ensemble	**774.19**	**515.55**	**0.9379**	**7.35%**

Interpretation:

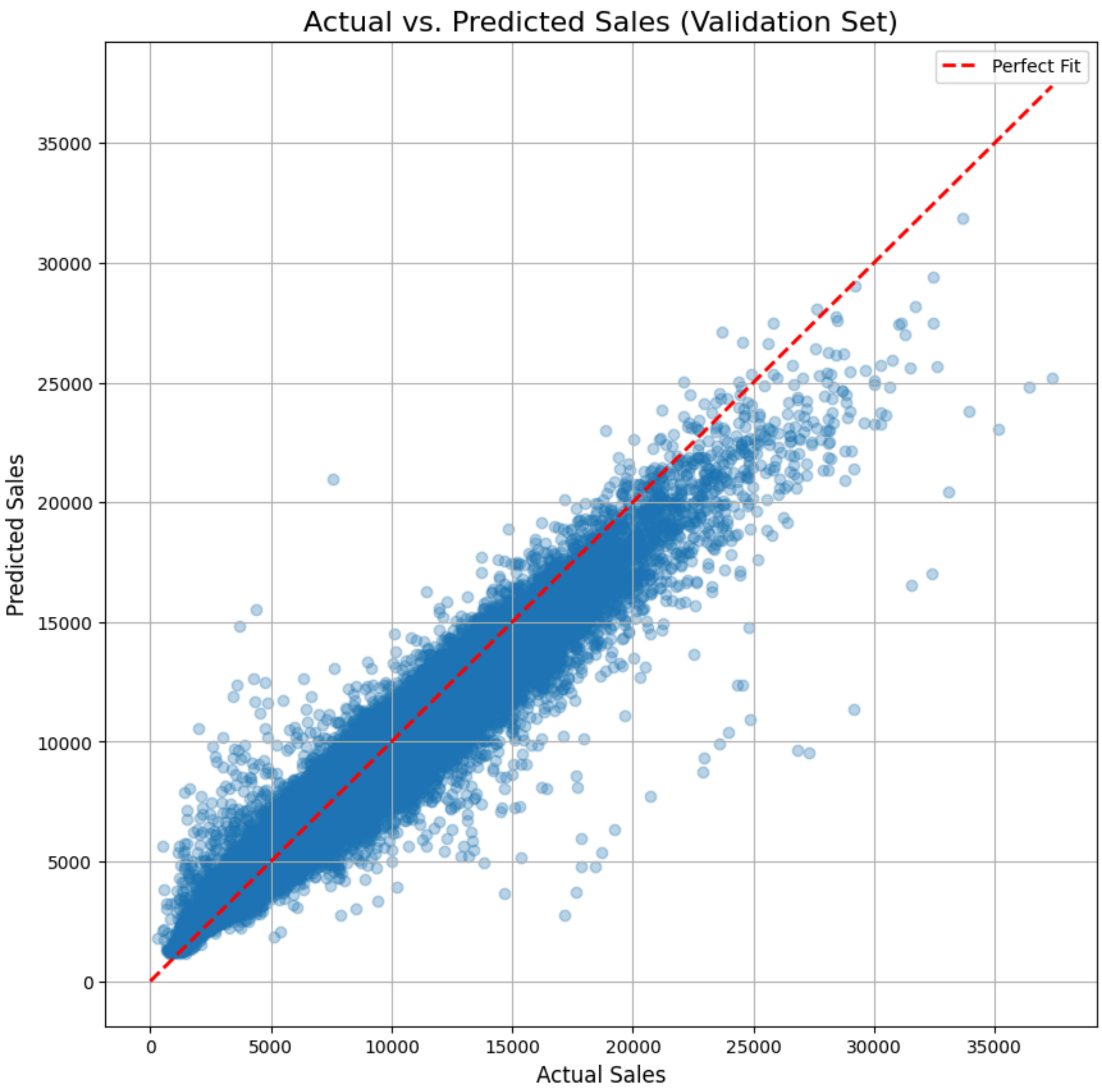
- Both models performed exceptionally well, with the **XGBoost model showing superior performance** individually.
- An **R-squared of 0.9583** for the XGBoost model indicates that it explains approximately 96% of the variance in the sales data, which is an excellent result.
- The **Mean Absolute Percentage Error (MAPE)** for the ensemble model is **7.35%**, suggesting that the average prediction is within about 7.4% of the actual sales value.

Additionally, i attempted to submit the competition to see how my approach is compared with other people’s approaches and got a score of 0.1185.

6. Forecast Visualization Examples

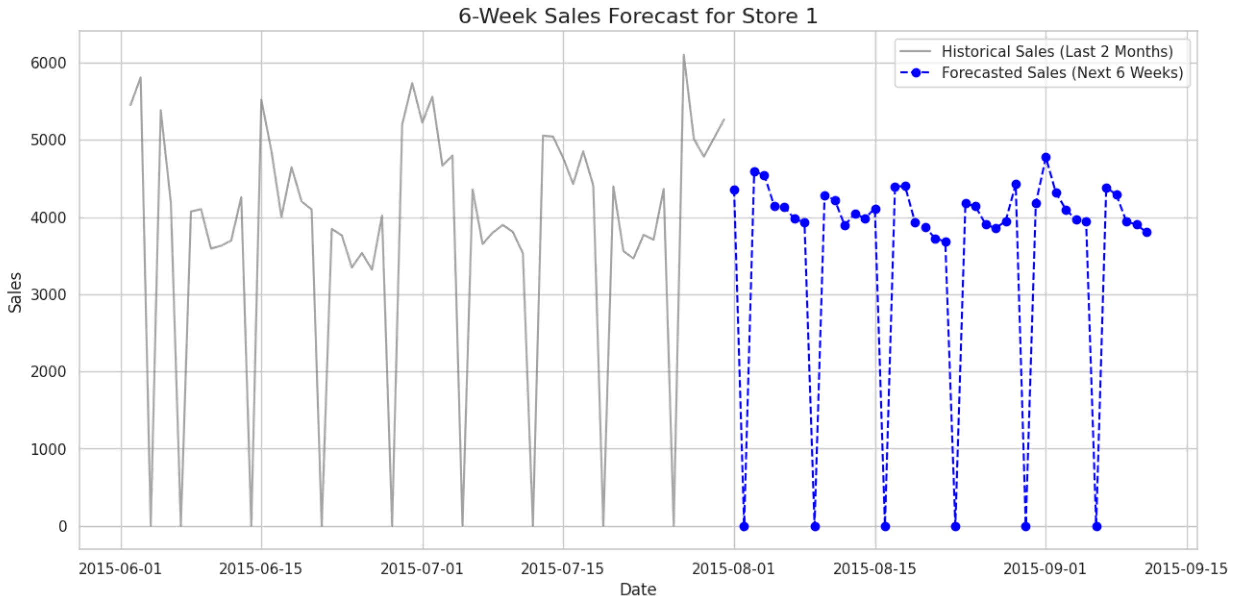
6.1. Actual vs. Predicted Sales

This plot compares the model's predictions to the actual sales values on the validation set. The tight clustering of points along the diagonal red line visually confirms the model's high accuracy and strong correlation with the true values.



6.2. 6-Week Forecast for Store 1

The model was used to generate a 6-week forecast. The plot below shows the forecast (in blue) against the most recent two months of historical data (in gray). The visualization clearly shows the model's ability to capture the weekly sales cycle, including the expected dips for Sundays.



7. Conclusion

The ensemble modeling approach, powered by robust feature engineering, proved highly effective in forecasting Rossmann store sales. The models successfully captured complex seasonal and promotional patterns, resulting in high accuracy on the validation set. The individual XGBoost model demonstrated the best performance, but the ensemble provides a stable and reliable solution. The full code can be found on this Google Colab link.

Backend API

To run a front end that shows the forcast for a certain store or the endpoint that should accept POST requests with JSON input like:

```
{
"store": 215,
"date": "2015-09-01",
"promo": 1,
"state_holiday": "0",
"school_holiday": 0,
"day_of_week": 2
}
```

● Return the forecasted sales in JSON format:

```
{
"predicted_sales": 4586.42
```

Either run the cells in colab under **Run Server**

Or clone this repo, download requirements and run app.py.