Machine Learning Model for Forecasting of Rossman Sales

Ashton Dickinson, Huzaifa Wahid, Matthew Haussrer

13 July 2024

1 Introduction

1.1 Business Background

Sales forecasting presents a significant challenge for retailers due to the diverse factors that influence sales, including promotions, competition, holidays, seasonality, and geographic location. Forecast analysis via modelling can support businesses and managers in many ways, including in boosting sales turnover and attracting customers from competitors via optimally timed promotions.

1.2 Task explanation and goal

The overarching goal of this project was to predict six weeks of daily sales for 1,115 retail outlets of the Rossman drug store chain in Germany. This was carried out by developing a time series machine learning model that integrates historical sales data to understand the variables that affect sales, which can be optimized to improve analytical value and the accuracy of forecasting.

1.3 Research questions

In this context, the key research questions to address from analysis of model results included:

1. One Which regression model provides the most accurate sales predictions?
2. Two Which feature exerted the greatest impact on sales?
3. Three Which days are optimal for implementing additional promotions to enhance sales?
4. Four How does the Sale of specific store varies from the Average sales for the other Store?

1.4 Project scope and objectives

The historic data set was first reviewed and pre-processed to ensure reliability and accuracy in forecasting. Subsequent processing involved data linking and integration, encoding of categorical data and cleaning of missing values for clarity of analysis. The impact of specific variables on sales was determined by examining relevant sales related variables such as closure days, daily turnover, and customer numbers. Additional store-related variables such as store type, distance from competitor, and promotional status and periodicity were also assessed for their effect on sales. Regression modelling was selected as the most suitable approach for producing the time series forecasting model based on the key variables identified above. This suitability derives from the applicability of regression modelling for distinguishing and quantifying both linear and non-linear relationships between sales and the variables that influence sales (Brockwell and Davis, 2016). Through comparison of three regression modelling techniques, the accuracy of the forecasting model could be evaluated using the Root Mean Square Percentage Error (RMSPE), in addition to other appropriate measures of error. These metrics helped assess data precision for training and validation of our model, and subsequent interpretation of the model's performance in terms of forecasting accuracy and the implications of this for optimizing sales at Rossman stores.

2 Methodology

To answer the questions, research on different machine learning models was conducted. After thorough analysis Regression models were selected to enhance our analysis and forecasting because of their ability to handle complex data and relationships, which makes them more suited for in-depth examination of various parameters on sales, thereby more precisely measuring our hypothesis and outcomes. Three models, the "Gradient Boosting Model" "Random Forest Model" and the "Multilayer Perceptron," were developed and trained simultaneously to come up with the best and most accurately predicted sales. The model with the greatest accuracy was then chosen for the results. Metrics like Mean Squared Error (MSE), R-squared, and Mean Absolute Error (MAE) are used to quantitatively assess the model's performance. These metrics provide valuable insights into the accuracy and reliability of the model. Before proceeding with model training, it was crucial to have an in-depth understanding of the data. Therefore, the "Store" and "Train" datasets were merged to create a new data frame named "Store open data," which included only the days when the store was open. The methodology was divided into two major parts: a. Data overview and analysis, and b. Model training and prediction.

2.1 Data Exploration

Before working on the model directly the data was explored and visualized to better understand the relationships and importance of the variables. The goal was to predict sales, so the main goal of the data exploration was to understand how each variable effected sale. Since the data set was a time series it was important to establish if it contained any periodicity. As figures 01 and 02 show the data has both a weekly as well as yearly periodicity of sales. This makes sense intuitively since the highest sales correspond to the holiday season, and daily low sales correspond to when many of the stores are closed.

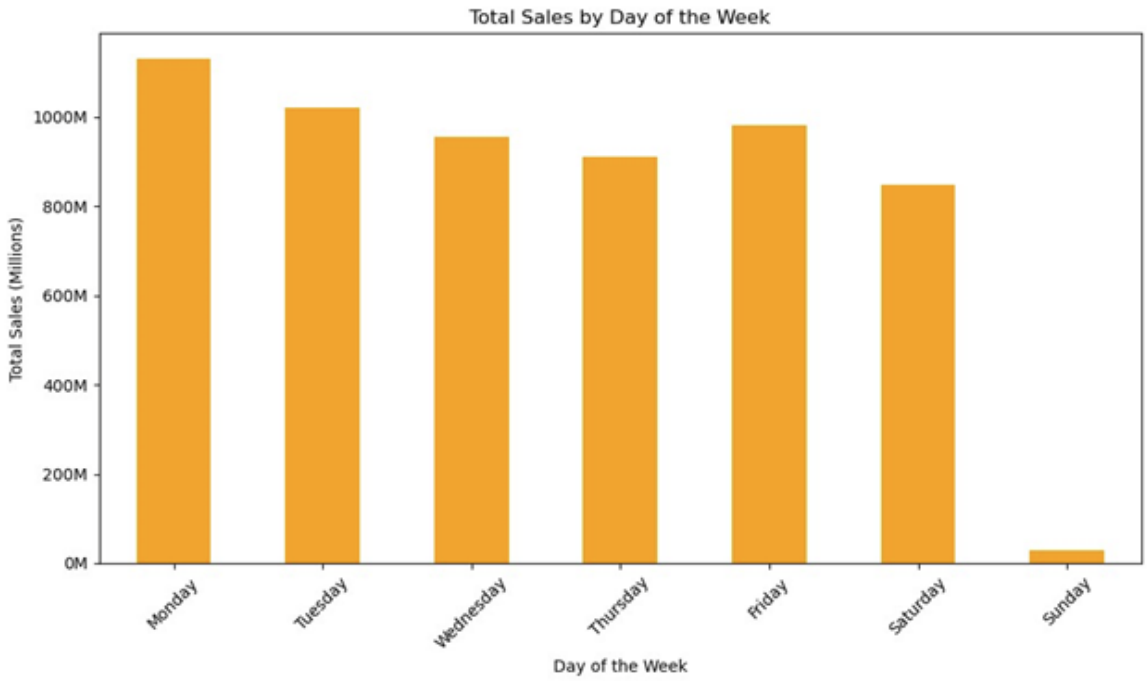

Figure 1: Total sales by day of the week

Next the importance of promotional status with regard sales was explored. Though this will be discussed in further model sections the overall finding as shown in Figure 03 showed a strong increase in sales under promotions as would be expected.

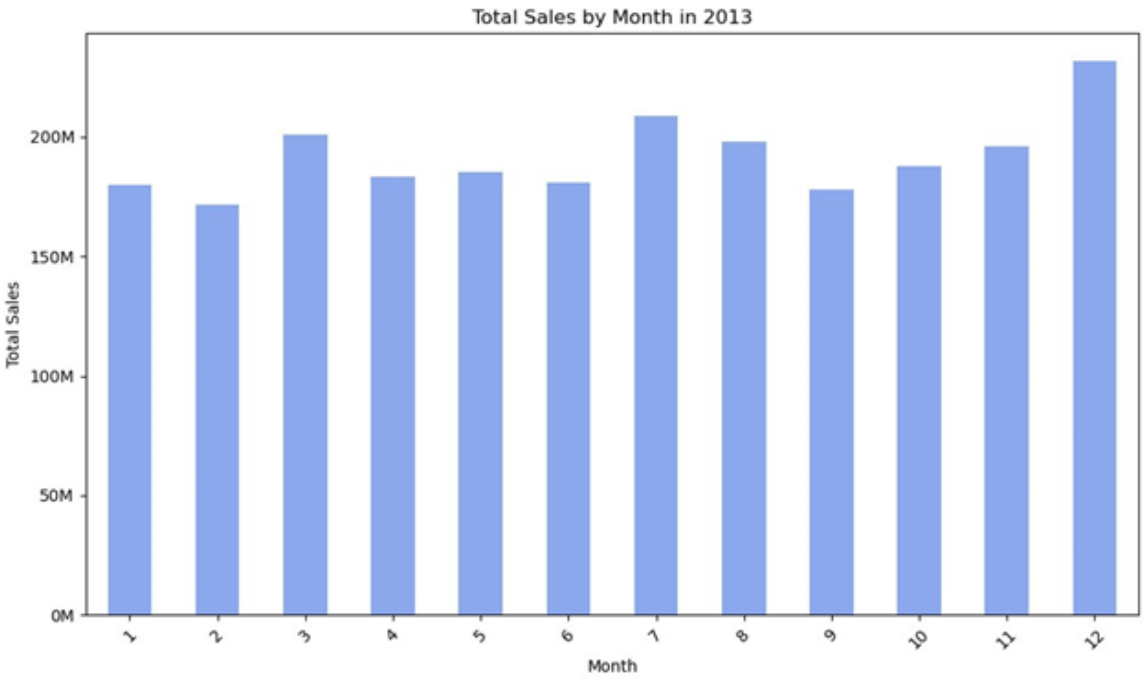

Figure 2: Total sale by month for the year of 2013

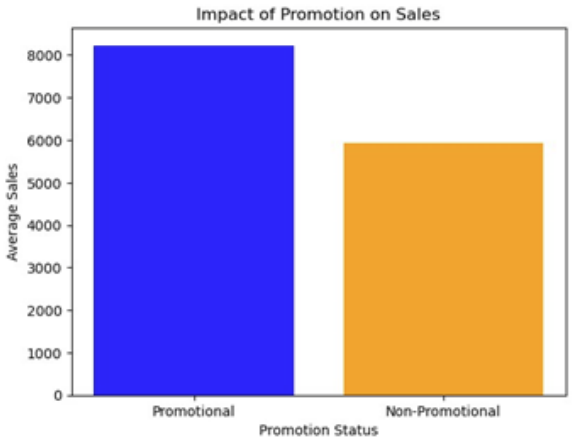

Figure 3: Average sales by promotion and non-promotion

The importance of competition distance and store type and assortment type was also explored. As Figure 04 shows, competition distance was related to the performance of different store types and promotion types. But since competition distance is also related to number of customers, being closer to competition appeared to improve sales.

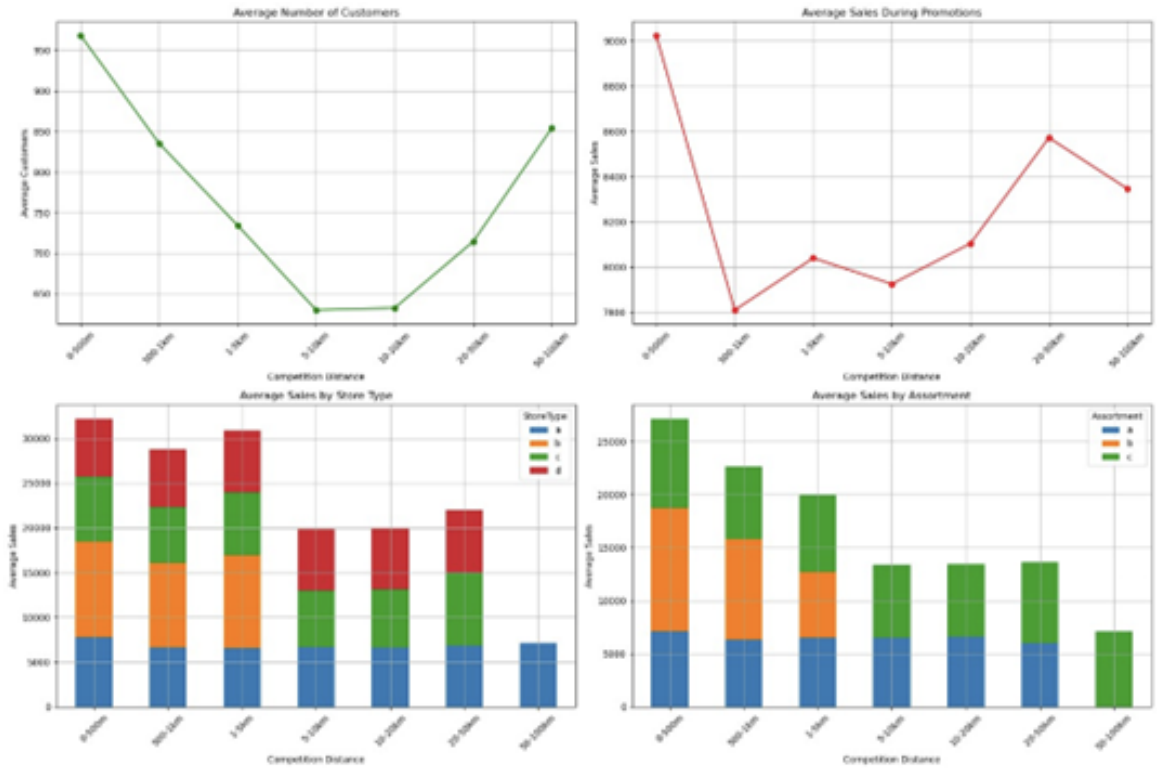

Figure 4: Comparisons of competition distance and sales

2.2 Multi-Layer Perceptron (MLP) and Random Forest

The first and second model chosen were the sci-kit learn multi-layer perceptron and random forest. MLP is a supervised neural network model which can be used for both classification and regression predictions. The model does warn on its hosting page that it does not have any GPU support, which was a limitation for its application in this project. Random forest is an algorithm which produces random sets of decision trees and can also be used for classification and regression problem. Though it also does not have GPU support in experience it appeared to train on the data set for this project much quicker.

2.2.1 Data Preparation

For both models the same data preparation and training was performed. In general data preparation followed many of the same steps as XGBoost so for the sake of brevity only the difference will be mentioned. The main difference in approach was the number of variables used to train the model. For example, competition distance was not used, rather than imputing its missing values. Beyond this difference the variables were processed in the same manner, first they were onehot encoded to improve the model interoperability of the store type variables, then the numeric variables were normalized. Due to the length of time it took to train models using the at first a single split, single run model was developed to test parameters. For this model the preprocessing steps were done separately. Though they were evaluated for performance using RMSE this score from the single split was only used for reference, not to evaluate performance compared to other models. After a range of parameters were found which could train in reasonable time, then the parameters were used as part of a cross-validation grid search or random search which could produce more rigorous results. In these cases, the sci-kit pipeline tool was used to scale the data after each split. This ensured that the training and validation data were completely isolated.

2.2.2 Grid Search, Random Search and Cross Validation

To optimize each model and get a comparable RMSE the sci-kit tools for random search, grid search and pipeline were used. These tools performed a search which tests each combination of provided hyper-parameters, or randomly selected parameters, and scored them based on cross validation, which was mentioned in the previous section. For MLP the grid search returned a better result. For random forest, the random search produced a better result. Since random trees could train faster but had more hyper-parameters the random search made it possible to test many combinations. Random forest did run into memory issues during training though which limited the size of parameters which could be used. The MLP training time was very slow, so a grid search was used across a narrow band of parameters. This is a notable limitation, and it is possible the model could have performed better if more processing power was available.

2.2.3 Model Selection and Forecasting

After the results from the different models and hyper-parameters were compared the best performing model was selected. To train a model to provide the forecast the entire training data was processed as above and used to train the model. The model was then run to predict the 6 weeks of sales for the stores. The model and the outputs were then saved. The model was compressed and saved using joblib, and the outputs were combined with test.csv and again exported as a .csv.

2.3 Gradient Boosting

The gradient boosting model was selected for its robustness in accurately predicting data and alignment with project goals. Its capability to handle complex data patterns effectively makes it a top choice. Features like rolling averages and delayed variables improve forecasting precision, while regularization techniques mitigate overfitting and promote generalization to new data. The model's resilience to outliers is crucial for time series data prone to anomalies, as observed in our dataset. Modern implementations such as XGBoost and LightGBM excel in handling large datasets and managing missing data, further enhancing accuracy and reliability in our analyses.

2.3.1 Competition and Promo Value Handling

To facilitate model training, we formatted competition and promotion values. We calculated the months since each competition opened by finding the difference in years and months between the current date and each competition's opening date. Next, we assessed the impact and timing of Promo2 campaigns. The 'promo cols()' function calculated the duration since the start of each Promo2 campaign by finding the difference in years and weeks. It also determined if the current month was part of the campaign, helping the model identify the impact on sales.

2.3.2 Data Preparing for Training and Testing

After formatting values, datasets were divided into training and testing sets. The testing set evaluated the model's hypothesis, while the training set provided initial knowledge. We assigned all columns except sales as input columns, with sales as the target column. Using "Store open data," we created input and test data frames. Finally, columns were categorized into "numeric cols" for numeric data and "categorical cols" for objective data.

2.3.3 Handling Missing Values

We used pandas' isna() function to identify missing values, finding 2,186 in the CompetitionDistance column of the inputs data frame and 96 in the test data frame. Stores "291," "622," and "879" had missing values. For imputing these values, we analyzed potential outliers and the value distribution using box plots and histograms. It was observed that 70 percent stores out of 1115 total stores have a competition distance between range 0-20,000 m and 58 percent of total stores have competition between 0-5000 m range. We chose median imputation for missing values, as it handles outliers and

skewed distributions well. The median distance of of stores with competition between 0-20,000m was used for imputing which came out to be 2170m.

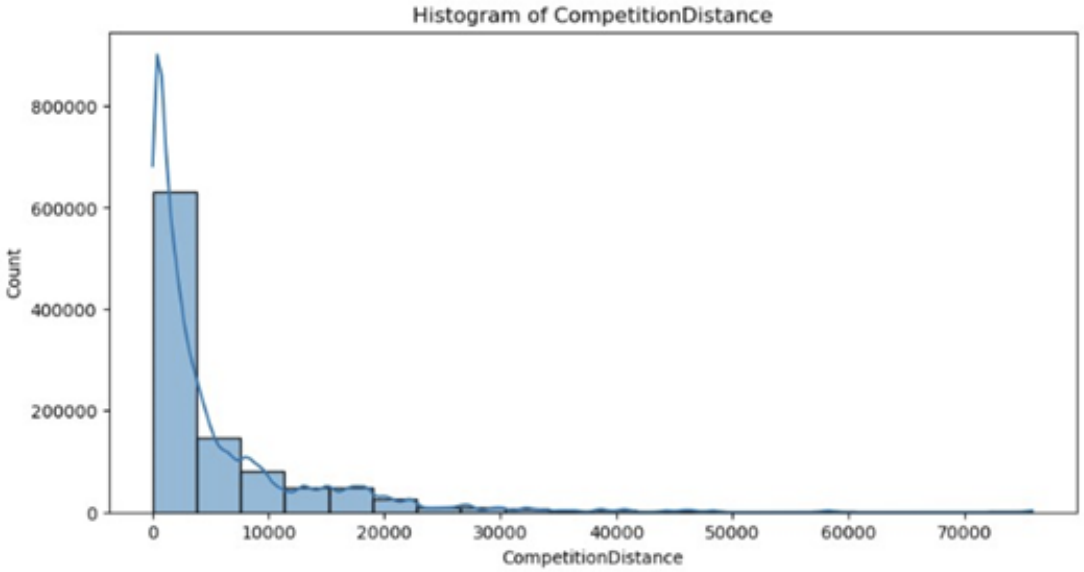

Figure 5: Histogram of competition distance respective to their occurrence

2.3.4 Scaling Numeric Values and Encoding Categorical Columns

To enhance model performance and convergence, we scaled all numeric values using the MinMaxScaler function, normalizing numerical columns in both data frames to a 0-1 range preventing features with larger scales from dominating those with smaller scales. Additionally, we converted categorical columns with one-hot encoding, representing each categorical column by separate binary columns, ensuring effective processing by machine learning algorithms.

2.3.5 Extracting Data for Training

To ensure that both datasets have a consistent set of features for model training and evaluation, we combined the numerical and one-hot encoded columns algorithms.re sets "X" for the training data and "X test" for the test data.

2.3.6 Training a Gradient boosting Model

Our predictive model was trained using XGBoost's XGBRegressor, which was configured with important parameters including number of boosting stages (n estimators), learning rate (learning rate), maximum depth of trees (max depth), and random state (random state) to ensure reproducibility. Using the prepared feature set X and matching target values (targets), we trained the model. The aim was to take advantage of gradient boosting to maximize predictive accuracy and capture complex patterns in the data.

2.3.7 Predicting and Evaluation

For the feature set X, we produced predictions (predicts) using the trained model. Once predictions were made a new function rmse was created to assess the accuracy of the model. Between the predicted values (predicts) generated by our trained model and the actual target values (targets).

2.3.8 Feature Importance Analysis

Using our trained XGBoost model (model), we calculated the feature importance. The most significant features are ranked in the resulting dataframe, importance df, according to how well they predict the target variable. Through the process of prioritizing and interpreting these influential factors for future insights and decision-making, this analysis assisted us in determining which features have the greatest influence on the predictions made by our model. Results displayed in Figure 6.

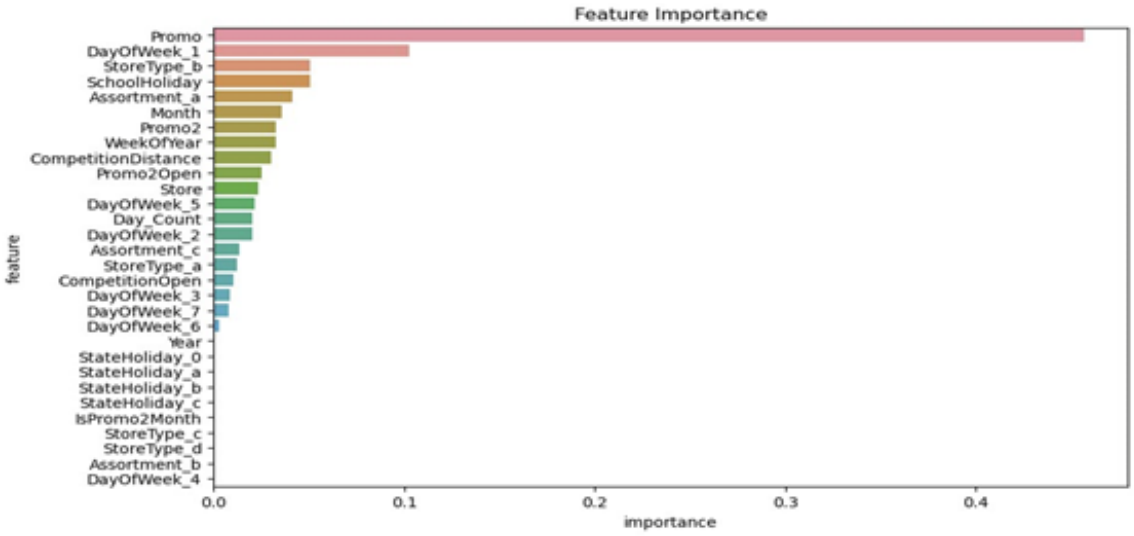

Figure 6: Features and their importance respective to sales

2.3.9 K-Fold Cross Validation

A helper function train and evaluate was created that trains a model with specified parameters and returns the trained model, along with the training and validation errors. It initializes the model with specified parameters (params), trains it using X train and train targets, and calculates Root Mean Squared Error (RMSE) for both training and validation sets (X val, val targets). This function streamlines the process of model training and evaluation, providing insights into how well the model performs on both training and unseen validation data. K-Fold utility was used to create the different training/validations splits and train a separate model for each fold.

2.3.10 Hyperparameter Tuning and Regularization

For hyperparameter tuning and regularization two functions were created. The test params kfold function divides the data into training and validation sets at multiple folds in a methodical manner, then uses K-Fold cross-validation to evaluate the model's performance. It computes the RMSE for training and validation for every fold, averaging these metrics to assess the model's overall efficacy with the given parameters. Test params, on the other hand, employs a single holdout set technique to assess model performance. It computes RMSE for both training and validation data, trains an XGBoost regressor with predetermined parameters, and offers vital information about the model's capacity to generalize to previously unseen data prior to deployment.

2.3.11 Final Prediction

Once the XGBoost Regressor model was configured with optimized parameters for predicting values, the fit method was employed to train it on the feature set X with corresponding target values (targets). This process optimized the model's parameters and internal state, allowing it to effectively learn from the provided data and make accurate predictions based on the learned patterns and

relationships within the dataset. Subsequently, the predict method was utilized to generate predictions (test predicts) using the trained model on a new dataset X test. These predictions enable us to forecast the target variable by leveraging the insights gleaned from the training data's patterns and relationships. The final results were stored in a data frame called "prediction file". The final prediction was saved using pickle.dump function, and the model was named "Predicting model.py"

3 Results

Out of the two sci-kit algorithms the random forest performed much better so no further validation was performed on MLP. But to improve comparison with the XGBoost the random forest was run again with two additional metrics. The run was cross validated, but only had two spits because of long run times.

3.1 Evaluation of Models

Metrics	Random Forest	XGBoost
RMSE	1290.45529217	381.70
MAE	848.51215071	278.44
R2	0.82637031	0.98
MSE	1665274.861	145701.5

Table 1: Comparative Assessment of Models

The Gradient Boosting Model, which performed well across all four quantitative assessments, was chosen as the final model to forecast the 6-week sales.

3.2 Predicted Results

The sales prediction is for six weeks, from August 1 to September 17th. The predicted sales data reflect an extensive analysis of the forecast period, including significant parameters such as a mean of €6295.069 and a standard deviation of €2184.297. Sales ranged from €0 to €24458.336, corresponding to variation across different percentiles, with values at the 25th, 50th, and 75th percentiles reported as €4882.467, €5988.609, and €7359.036, respectively. These findings give a thorough summary of the predicted Euro sales performance throughout the forecasted timeframe.

Statistic	Value
count	41088.000
mean	6295.069
std	2184.297
min	0
25%	4882.467
50%	5988.609
75%	7359.036
max	24458.336

Table 2: Predicted Sales statistics from 01-Aug-2015 until 17-Sep-2024

3.3 Dashboard for Visualization

A dashboard was created to help users interpret data and make informed decisions. It takes the store ID as input and presents several insightful charts: daily sales for the specific store in both line and bar formats, monthly sales trends, weekday-wise sales analysis to guide promotional strategies during low-sales days, and a comparison of daily sales for the selected store versus the average sales of all other stores. This dashboard serves to analyze the store's performance comprehensively and facilitates informed decisions for enhancing the store's progress and benchmarking against others.

4 Conclusion

In our analysis to predict six weeks of daily sales for 1,115 retail outlets of the Rossman drug store chain in Germany, we evaluated several regression models and features to identify the most effective approach for accurate sales forecasting. Our primary objective was to determine how various variables could be optimised to enhance the analytical value and accuracy of these predictions. Our findings indicate that the XGBoost model outperformed other models in terms of accuracy. This model's robustness and ability to handle complexity in both the amount of data and the relationships between many variables made it particularly effective for our time series forecasting. The features with the greatest influence on sales were found to be promotions and the day of the week, highlighting the importance of these variables in boosting sales. Additionally, our analysis showed significant variability in day-wise sales across different stores, emphasising the need for tailored promotional strategies. Our dashboard provides detailed insights into the sales performance of each store compared to the average sales of other stores, enabling more informed decision-making for store managers.

4.1 Implications

The results of our study have several implications for the Rossman drug store chain, including in:

- Model selection. Of the regression models implemented, XGBoost exhibited superior performance over the Random Forest run. This model comparison therefore showed the value of using advanced machine learning techniques for sales forecasting. Retailers might then consider integrating such models into their forecasting processes to improve accuracy and reliability in sales predictions and inventory management.

- Setting of promotional strategies. The strong influence of promotions on sales suggests that targeted promotional campaigns can significantly boost sales. Although intuitively obvious, this is important to validate so managers can optimise days for these promotions to help maximize their impact.
- Providing store-specific insights. The variability in sales performance across stores indicates that a one-size-fits-all approach may not be effective. By customising strategies based on store-specific data, better sales volumes can be achieved.
- Providing useful analytical tools for Rossman staff. The use of dashboards provides clear visual insights based on the historic sales data. This can help managers to identify trends, make more informed decisions around promotion strategy or adjustments to promotion periodicity, and ultimately raise store profitability.

4.2 Recommendations

Based on our findings, we derive a few recommendations for Rossman (and indeed similar retail chains) to continue enhancing forecast accuracy and overall sales performance. Primarily, adoption of advanced models such as XGBoost, trained on sales data and account for numerous interacting variables, can deliver robust and accurate future forecasts for strategic decision-making by stores. Our analysis can be used as a tool to plan promotions on days identified as having the highest potential for increased sales. This more targeted approach in turn helps achieve better promotional outcomes. We also recommend stores develop tailored strategies based on their specific sales patterns. This approach ensures that each store's unique characteristics, e.g. proximity to competitors, are considered, leading to more effective optimisation of sales. Finally, as the retail environment changes over time, e.g. through economic conditions that impact stock turnover, the performance of the forecasting models should be regularly reviewed, and parameters adjusted as necessary to maintain accuracy and relevance in the predictions.

5 References

Brockwell, P. J., Davis, R. A. (2016). Introduction to time series and forecasting. In Springer texts in statistics (3rd ed.). Springer Cham. https://doi.org/10.1007/978-3-319-29854-2

6 Annex

Figure 7: Dashboard showing predicted Sales for the inquired store from 01-Aug-2024 until 17th Sept-2024