Rossmann Sales Prediction

Introduction:
In this supervised machine learning project, we are provided with sales and store data from Rossmann, a company that operates multiple stores across Europe. Our main objective is to analyze the sales data and develop a model capable of predicting sales for the next six weeks. Additionally, we aim to identify the key factors influencing sales and offer recommendations to enhance the company's growth and overall sales performance.

Data Merging and Overview:
We were provided with two datasets: store data and sales data. To create a comprehensive dataset, we merged these two datasets using the common column "store." The resulting dataset contains 18 features and over 1 million rows. Here, sales were the dependent feature that we aimed to predict for further analysis and modeling.

Data Wrangling and Missing Values:
During the data wrangling phase, we performed basic data cleaning and identified the presence of null values in the dataset. Fortunately, there were no duplicate rows. However, we observed that three columns related to promotions had approximately 50% missing values, and two columns related to competition had about 30% missing values. Additionally, a column called "competition distance" had only 2.6% missing values.

We saw that the "competition distance" column had a right-skewed distribution. We imputed missing values with the median to preserve the data distribution.

Mode is used to fill in missing values in the "CompetitionOpenSinceMonth" and "CompetitionOpenSinceYear" columns, where more than 30% of the data is missing. This approach ensures that the most frequent values are used to maintain the data distribution.

Similarly, "Promo2SinceWeek", "Promo2SinceYear", and "PromoInterval" columns, with around 50% missing values, are imputed with 0 because these missing values indicate that there was no promotion activity for those specific entries, making 0 a meaningful and appropriate value to use.

Exploratory Data Analysis (EDA) and Insights:
During Exploratory Data Analysis (EDA), we used the date feature to create new features related to year, month, and day of the week, providing valuable insights into temporal patterns. Our observations revealed that promotions positively influenced customer footfall and sales. Store type B, despite being few in number, had the highest sales average due to unique assortment offerings and being open on Sundays. Additionally, sales were higher on Mondays, likely because shops are generally closed on Sundays. A decline in sales was noticed, potentially due to competition in the market. These insights helped us understand the business better and provided valuable information for making necessary changes and improvements.

Recommendations:
Based on the insights, we recommend that promotions should be used to attract more customers and boost sales, as they have a positive impact on customer footfall and sales. Secondly, expanding the number of Store type B outlets could be beneficial, as they have higher profitability due to unique assortment offerings and Sunday openings. Leveraging seasonality by promoting products and offering discounts during holidays and special occasions can maximize sales during peak periods. Implementing these strategies can help Rossmann enhance overall sales and achieve sustainable growth.

Model Building and Handling Missing Values:
After conducting Exploratory Data Analysis (EDA), we went back to building our models. However, we noticed that some columns had around 50% missing values. To avoid potential bias and improve the accuracy of our machine-learning algorithms, we decided to remove those columns with significant missing data. This way, we ensured that our models were based on more reliable and complete information, leading to better predictions and results.

Feature Selection using Correlation Heatmap:
After addressing missing data, we employed a correlation heatmap for feature selection. Features with high correlation are closely related and have similar effects on the dependent variable. When two features are highly correlated, we can drop one to avoid redundancy. For example, the independent features "Month" and "DayOfYear" showed a high correlation, so we removed the "Month" column, keeping the "DayOfYear" column for our analysis. This ensured that we retain the most relevant and independent features for building accurate predictive models.

Outlier Treatment for Sales Data:
After feature selection, we proceeded to address outliers in the sales data. On plotting the sales data, we noticed a significant concentration of values around 0, causing the data to be skewed. However, before treating these outliers, it was crucial to determine whether they were justified or not. To identify outliers, we utilized the z-score test. Upon exploration, we discovered that many sales values were close to 0 because some stores were temporarily closed for refurbishment, resulting in zero sales during those periods. Since these values could potentially impact our model's accuracy, we decided to remove rows with sales values of 0 from the dataset for now. This step ensures that our model is not influenced by the temporary closures and helps us focus on relevant sales data for more accurate predictions.

Log Transformation for Sales Data:
After removing the outliers, we found that the sales data had a right-skewed distribution. To make the data more suitable for analysis and modeling, we applied a log transformation. This transformation scales down the values and reduces the impact of extreme values, making the data closer to a normal distribution.

Data Splitting for Training and Testing:
Then we divided the dataset into training and testing datasets. Since our goal is to predict sales for the next 6 weeks, we chose to use the last 6 weeks' data for testing. The remaining data were used for training the machine learning models. This division was made using a 94:6 splitting ratio, where 94% of the data was used for training, and 6% was set aside for testing. This approach helps us evaluate the model's performance on unseen data and ensures that our predictions are reliable and accurate for future sales.

Categorical Feature Encoding with One-Hot Encoding:
After splitting the data, we focused on encoding categorical features to prepare them for model training. We used one-hot encoding for three features: "store type," "assortment," and "state holidays." We used this encoding method as it is suitable for nominal data, where categories have no inherent ordering and cannot be directly compared or ranked. One-hot encoding ensures that each category is represented by its own binary column.

Data Scaling with StandardScaler:
In the dataset, features have different scales and units, which can introduce bias in the model. To address this issue, we used the StandardScaler method for data scaling. This method transforms the features to have zero mean and unit variance, bringing them to a similar scale. It ensures that no feature dominates others, thus preventing any bias in the model.

Model Implementation and Evaluation:
After preprocessing the data, we implemented machine learning models using three algorithms: linear regression as the baseline model, followed by ridge regression, and finally, a decision tree for model creation. To assess the model performance, we utilized various metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R2), and Adjusted R-squared (Adjusted R2).

Among these metrics, we considered Adjusted R2 as the primary evaluation criterion. It provides a more reliable indication of the model's goodness of fit, considering both the accuracy and the complexity of the model. By using Adjusted R2, we ensured that our chosen model would strike the right balance between predictive accuracy and model simplicity, leading to the best overall performance.

Linear Regression - Baseline Model
The baseline model, which is the linear regression model, gave an adjusted R2 score of 74%.

Ridge Regression with Hyperparameter Tuning:
Then we used Ridge Regression and performed hyperparameter tuning using Grid Search Cross-Validation (GridSearchCV) to tune the alpha parameter, which controls the complexity of the Ridge Regression model. We used 10-fold cross-validation with Grid Search for hyperparameter tuning and tested different alpha values. However, even with this optimized alpha value, the performance of the tuned Ridge Regressor did not improve significantly beyond the baseline model.

Improved Performance with Cross-Validated Hyperparameter Tuned Random Forest:
After extensive experimentation, we achieved a substantial improvement in our model's performance by using a cross-validated hyperparameter-tuned random forest model. The adjusted R-squared score reached an impressive 93%, indicating a significant enhancement compared to the baseline linear regression model. This improvement represents a remarkable 25.918% increase in accuracy.

To achieve this performance boost, we utilized RandomizedSearchCV with 5-fold cross-validation on a DecisionTreeRegressor model. We explored various hyperparameters, such as 'max_depth', 'max_features', 'min_samples_leaf', and 'min_samples_split', testing 10 different combinations to identify the best configuration. With the optimal set of hyperparameters, we created the final DecisionTreeRegressor model, which turned out to be a highly effective predictor for sales data.

Examining Feature Importance:
At last, we examined the feature importance of our model, as it's a Tree-based algorithm, it naturally provides feature importance scores during training. After analyzing the feature importance, we discovered that certain features had a more significant impact on the model's predictions. The most influential features turned out to be customers, competition distance, store type, and promotions.

These features play a crucial role in influencing the sales predictions made by our model, and understanding their importance helps us focus on the most relevant factors affecting sales in Rossmann stores.

Saving and Testing the Best-Performing Model:
Finally, we saved the best-performing machine learning model in a pickle file for future use. Later, we loaded the saved model file to test its predictions on unseen data as a sanity check. This process allowed us to confirm that the model's performance remained consistent when making predictions on new, unseen data, validating the reliability of our predictive model for practical use.