**Demand Forecasting for Rossmann’s Retail Chain**

George Stampolidis Data Scientist Intern Grant Thornton - Technology

**Modules**

**Abstract ………………………………………………………………. Page 2**

1. **Introduction …………………………………………………………... Page 3**
2. **Related Work ………………………………………………………… Page 5**
3. **Methodology ………………………………………………………….. Page 6**
4. **Results ………………………………………………………………… Page 9**

**4.1 Exploratory Data Analysis ……………………………………. Page**

**4.2 Time Series Models ……………………………………………. Page**

**4.3 Machine learning Models …………………………………… Page**

1. **Discussion …………………………………………………………… Page**
2. **Conclusion …………………………………………………………… Page**
3. **References …………………………………………………………… Page**

**Appendix …………………………………………………………….. Page**

**Abstract**

This analysis is based on the factors that have an impact on sales in Rossmann’s stores. The goal is to predict the incomings from sales for each store. I calculate the most significant descriptive measures and I create visualizations to understand data with the best possible way. Moreover, the correlation between variables is found out , in order to uncover patterns . To forecast sales for each store, time series models and machine learning algorithms are developed. More specifically, ARIMA, Prophet, XGBoost and LightGBM are built.

This project provides some important findings. Most stores are related to beauty and personal care while having basic variety of products. Sales and the number of customers are balanced with the passage of time and it looks like that promotion on specific days leads to better results compared to continuous promotion. Furthermore, sales are not affected by the distance of competitive stores which shows that Rossmann’s stores have built a stable client base.

In terms of predictive models, there are some cases where we have acceptable models and other times that models are inappropriate. Prophet seems to be more sufficient than ARIMA because most times it has lower metrics and takes into account more factors. There were experiments that ARIMA had better performance and some of these provided better results with logarithmic transformation on data. However, ARIMA does not meet the model assumptions therefore we cannot accept its forecasts. From all the experiments it is observed that XGBoost has lower metrics and provides more accurate predictions than LightGBM. This research shows that Prophet has better fit than ARIMA in terms of time series models, while XGBoost has better accuracy than LightGBM in machine learning algorithms.

1. **Introduction**

In this survey I have used data that contain information for Rossmann’s stores in Germany. I am provided with two datasets that include valuable details about Rossmann stores and as a result I have decided to merge those data sets. The main target is to forecast the turnover from sales for every single store using appropriate statistical models. The owners want to know the trend for the incomings from sales for each store for the next two months. Therefore, I was asked to make predictions for August of 2015 and the first three weeks of September of 2015 and more specifically until 17th of September. Data include historical information for 1115 stores from January of 2013 until July of 2015. There are eighteen variables for each store that will help for an accurate statistical analysis (See Table 1). These features include details about sales, the number of customers, holidays, variety of products, store types and data about competitive stores.

Something important that needs to be mentioned is that there were some stores under renovation without knowing the exact days that this situation took place. This is an important factor that should be taken into consideration for models’ interpretation. Also, there were many closed stores because of non-working days and I had to change some variable types in order to control our data with the best possible way. In numeric features I had to replace missing values with the median value of each variable and in categorical features I had to replace missing data with the most frequent value of each variable. During my project I observed that some stores were open without earnings from sales so I had to deal with this mistake in the collection of the data. An easy way would be to change the status of these stores as closed, but I decided to replace zero sales with the median value of sales in order to have more balanced data and not to add more outliers. Moreover, two new variables were created to help us with data analysis. The first is “PromotionFlags” that combines the variables “Promo” and “Promo2” which are respectively referred to promotion in a specific day or continuous promotion. The second feature is named “HolidayIndicators” which combines the variables of state and school holidays. In addition, some features like weekdays (WeekDay), months (Month), day of years (DayOfYear), weekends (IsWeekend) and weeks of year (WeekOfYear) were created to help models understand time patterns and improve their accuracy. In the next modules of this project, univariate and multivariate analysis will be applied, the methodology used will be explained, time series and machine learning models will be developed and the most important findings from this survey will be discussed.

**Table 1: Rossmann Stores Data**

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Type | Description | Values |
| Store | **Categorical** | **Store identification number** | **1-1115** |
| DayOfWeek | **Categorical** | **Days** | **1-7** |
| Date | **Temporal** | **Date** | **1/1/2013 - 31/7/2015** |
| Sales | **Quantitative** | **Incomings from sales** |  |
| Customers | **Quantitative** | **Number of clients** |  |
| Open | **Categorical** | **Shows whether a store is open or not** | **0=Closed**  **1= Open** |
| Promo | **Categorical** | **Indicates whether a store is running a promo on a specific day** | **0=Promo 1=No Promo** |
| StateHoliday | **Categorical** | **Describes a national holiday** | **0 = None a = Public b = Easter c = Christmas** |
| SchoolHoliday | **Categorical** | **Indicates the presence of school holiday** | **0 = No 1 = Yes** |
| StoreType | **Categorical** | **Type of the stores** | **a = Beauty & Personal Care b = Online Store c = Health & Wellness Store d = Home & Household Store** |
| Assortment | **Categorical** | **Variety of products** | **a = Basic b = Extra c = Extended** |
| CompetitionDistance | **Quantitative** | **Distance from competitive stores (in meters)** |  |
| CompetitionOpenSinceMonth | **Categorical** | **Indicates the month competitive stores opened** |  |
| CompetititonOpenSinceYear | **Categorical** | **Indicates the year that competitors opened** |  |
| Promo2 | **Categorical** | **Continuous Promotion** | **0 = No Promo 1 = Promo** |
| Promo2SinceWeek | **Categorical** | **Week that stores started participating in Promo2** |  |
| Promo2SinceYear | **Categorical** | **Year that stores started participating in Promo2** |  |
| PromoInterval | **Categorical** | **Months that continuous promotion (Promo2) starts** | **Jan/Apr/Jul/Oct Feb/May/Aug/Nov Mar/Jun/Sep/Dec** |

1. **Related Work**

Relevant literature on sales forecasting encompasses a wide range of methodologies, from traditional time series analysis methods to modern machine learning algorithms. In this section, we review key studies that have investigated the effectiveness of ARIMA, Prophet, XGBoost and LightGBM models in predicting earnings from store sales.

ARIMA model has been the most used method of time series analysis for many years, with plenty of researches evaluating its performance in sales forecasting. In recent years researchers have found various extensions of ARIMA to enhance its forecasting capabilities. Its performance can be limited by the assumption of linearity and stationarity in the data. Surveys have shown mixed results regarding the effectiveness of ARIMA but there are times that results are sufficient and some others not.

Prophet is a time series model released by Meta’s data science team in 2023. It is an additive model where non-linear trends fit with seasonality and holiday effects. It is a newly developed model and as a result it hasn’t been used in many surveys. However, on those used, it makes satisfactory predictions.

Moreover, machine learning algorithms like XGBoost and LightGBM have gained increased popularity for demand forecasting due to their ability to deal with big datasets. Their tree-based algorithm contribute to better predictions and higher performance.

There are many surveys that have used and compared the aforementioned methods. While some researchers support the flexibility and predictive power of XGBoost, others argue that the simplicity of ARIMA makes it more suitable for certain tasks ("Accuracy comparison of ARIMA and XGBoost forecasting models in predicting the incidence of COVID-19 in Bangladesh.", Md Siddikur et al.). The choice between ARIMA and XGBoost depends on the characteristics of the dataset and the forecasting horizon. In this survey, ARIMA performed better than XGboost for short period forecasts because it provided lower errors on test data and as a result more accurate predictions. The paper "Improved Sales Forecasting using Trend and Seasonality Decomposition with LightGBM" by Tong Zhou at the Department of Computer Science at Johns Hopkins University, used Prophet and LightGBM methods. Prophet was used for trend and seasonality analysis while LightGBM leveraged these features for the final forecast, achieving improved performance on complex sales data. According to the paper “Forecasting Sales Trends Using Time Series Analysis: A Comparative Study Of Traditional And Machine Learning Models” of the university of Portsmouth, both times series and machine learning models have unique strengths. ARIMA is better with structured and seasonal data patterns while XGBoost and LightGBM have more accuracy with big dimensional data. The conclusion of this research is that Machine learning algorithms provide improved forecasts compared to ARIMA.

1. **Methodology**

The methods used in this research contain correlation between features with appropriate statistical techniques, feature engineering for the best explanation of the data and predictive time series and machine learning models. The models developed are ARIMA, Prophet, LightGBM and XGBoost. It is important to mention that for the statistical analysis I used the programming language Python. In this section, the methodology used for data analysis will be described explicitly.

First of all, data were collected by Rossmann stores and I was asked to develop predictive models in an effort to predict future turnover from sales. Some mistakes in the collection of the data were detected, so I had to replace these mistakes with either median or most frequent value for each factor. In terms of feature engineering, I created two new variables for holidays and promotions. “HolidayIndicators” combines school holidays and national holidays and “PromotionFlags” combines continuous promotion for stores with promotion on a specific day. Moreover, some factors like weekends, months and weeks of years were created. Also, I had to create a new data frame that includes holidays with the exact dates respectively in order to use it for the Prophet model so it could understand better the holiday effects. This is crucial by the time Prophet’s function in python can get in input the dates with holidays.

In addition to, I applied exploratory data analysis to calculate descriptive statistics and table frequencies for each variable. Using plots such as correlation plots, boxplots, bar plots, histograms and pie charts for univariate and multivariate analysis will help us obtain important insights. It is crucial to understand the relationships between factors with suitable statistic tests. For numeric variables that do not meet the assumptions of normally distributed, homoscedastic, linear and independent residuals then non-parametric Kendall’s and Spearman’s correlation coefficient are used. Otherwise, Pearson’s correlation method is used. A value close to one indicates strong correlation, a value close to zero shows no relationship and a value close to minus one points out strong negative dependence. For categorical features Chi-Square test is used, with the null hypothesis being that there is no dependence and the alternative that correlation exists. With non-parametric Wilcoxon test we can examine whether an arithmetic variable depends on categorical variables. For chi-square and Wilcoxon tests we use p-values to determine if we are going to reject the null hypothesis for significance level a = 0.05.

The most vital part of this research is the development of predictive models to forecast the incomings from sales for every store.

* ARIMA (Autoregressive Integrated Moving Average) and SARIMA (Seasonal Autoregressive Integrated Average) are statistical models that predict future values based on historical observations. ARIMA is performed in stationary time series and SARIMA for non-stationary series. Stationarity is checked with Augmented Dickey Fuller Test (ADF test) that checks the null hypothesis of stationarity against the alternative hypothesis of non-stationarity. The function used in python is “adfuller”. For the tests we use p-value with significance level a=5%. The main reason stationarity is checked, is to see if the mean value, the variance and the autocorrelation of time series are stable with the passage of time. If so, we should have more accurate predictions. If we detect problem with stationarity then we should apply differencing method. Such method removes the trend from time series and coverts it in stationary series. In addition, a check for seasonal patterns should take place. Seasonality is checked with Autocorrelation plots (ACF plot) where we try to find out big spikes in certain lags. The command in python for ACF and Partial ACF plots are “plot\_acf” and “plot\_pacf”, respectively. It is important to understand where seasonality exists because I have to select the best possible parameter “m” for the command in Python. Auto regressive integrated moving average models assume that there are uncorrelated, homoscedastic and normally distributed residuals. If these assumptions are not met, then ARIMA forecasts could not be trusted. Metrics like RMSE (Root mean squared error) , MAE (Mean absolute error), MAPE (Mean absolute percentage error) and AIC (Akaike’s Information Criterion) are used to evaluate model’s performance. RMSE, MAE and MAPE calculate the distance between the predictions and the real values. MAPE is not appropriate for our dataset because of the many outliers and the closed stores without sales and therefore it returns enormous values sometimes. In python we use function “auto\_arima” with parameter “stepwise=True” to select the model with the lowest AIC with the best possible parameters p (autoregressive order) ,q (moving average order) and d (difference order). This command handles stationarity automatically and if non-stationarity is detected then difference method is performed (parameter d indicates differencing order). Also, this function can detect seasonality in time series when enabling the parameter “seasonal=True” ,in models’ input. The output of each model in Python, shows the selected model with the best possible parameters, AIC, autocorrelation test (Ljung Box – H0: Autocorrelated residuals vs H1: Uncorrelated residuals), heteroskedasticity test (ARCH Test – H0: No heteroskedasticity vs H1: Heteroskedasticity) and Normal distribution check (Jarque-Bera – H0: Normal distributed residuals vs H1: Not normally distributed residuals). Moreover, it returns plots with predicted and actual sales on test data, forecasts for August and September and models’ evaluation metrics. Data are divided in two parts. The first one contains data that will train the model (train\_data) and the second one includes data that will evaluate the performance of each model (test\_data). The error for each model is calculated by the division of RMSE with mean sales for each store in test data. In data science errors lower than 20% are accepted. Because of some negative predictions, logarithmic transformation was applied before model training. This transformation stabilizes the variance and reduces the impact of large outliers. After prediction, the inverse transformation was applied to return results to the original scale.
* Prophet is a time series forecasting model that considers seasonality and holiday effects. It performs better with series that have strong seasonality and many seasons of historical data. The same train and test data are used as above. By the time Prophet recognizes holidays, it is of vital importance to create a new data frame that will include holidays with respected dates and include this in model’s input. The function used in python is “Prophet”, with default parameters. We have the chance to add some important features in the model (command “add\_regressor”) but we must be very careful because if those features have many outliers then we might have negative predictions and wrong results. Unfortunately, there are cases where negative predictions are observed. This is a reasonable fact on this dataset because it includes non-working days with zero sales and as a result the model is affected. Statistics used to evaluate model’s performance are RMSE, MAE and MAPE. RMSE measures the average difference of the actual values from the predicted ones and the difference is squared to avoid the cancelation of positive and negative values, while they are summed up. MAE measures the average size of the errors in predictions and MAPE calculates the percentage of the errors. Prophet is an easier to apply algorithm because it does not have any model assumptions. Finally, log transformation was applied in the models because of the plenty outliers in sales to examine whether results have improved with such approach.
* LightGBM is a machine learning algorithm with high performance gradient boosting framework that uses tree-based algorithm. With tree-based algorithm we mean that many trees are created for every feature in order to achieve the best possible predictions. Firstly, it is important to separate data set on train and test data for model’s evaluation. Hyperparameter tuning is performed with grid search method in order to select optimal parameters for the models. This method checks all the combinations of parameters and selects those with the lowest RMSE. Although the best way would be to perform tuning for every time series, I perform hyperparameter tuning in the model with all the series so as not to spend much valuable time. A small number of combinations is selected therefore it could be great to compare models’ evaluation with both default and tuning parameters. Four models are developed with this algorithm but no one includes customers because they have strong correlation with sales and the model would not be trained with the appropriate way. The first one includes all the features with tuning parameters, the second one includes all the variables with default parameters, the third has features occurred from feature importance plot (Plot that shows the most important features for the model) and the last one includes some variables that from my point of view impact forecasts. SHAP is a method that explains machine learning predictions and it helps us understand how much each feature contributes to a specific prediction or overall model behavior. Metrics like MAE, MAPE, RMSE and cross validation are used for models’ evaluation. Cross validation with folds divided by the command “TimeSeriesSplit” , is applied to preserve the temporal ordering and avoid information leakage. This approach ensures that the model is always trained on past data and evaluated on future observations. If mean RMSE is lower, then the model provides accurate predictions and when lower standard deviation is observed, it is understood that algorithm is constant with different test data. Some negative predictions are observed for stores with many non-working days, so I set these values equal to zero. To predict the turnover from sales, a data set provided by the stores is used that has information about days in August and September where each store is going to run promotion and it also includes holidays.
* XGBoost is very similar to LightGBM algorithm. It is a machine learning model that uses gradient boosted decision trees and it is known for its high speed and accuracy. Algorithms with both tuning and default parameters are developed to compare their performance. Also, the same way mentioned in LightGBM is followed to create models, once again without including purchasers. SHAP plots are used for algorithms interpretation and cross validation as well as other statistics (RMSE, MAE etc.) are calculated to evaluate the accuracy of each model. Cross validation can be applied to compare the performances between different algorithms (for instance with LightGBM). If mean RMSE is lower for a specific model than the other, then we could conclude that the first model makes better predictions. Also, if a model has lower standard deviation than the other then we understand that the first one is more constant with different test data. If the performance differs for each fold, then we have indications that the model might overfit. Plots return predictions on test data for July and forecasts for August and September. Finally, the negative forecasts are replaced with zero. By setting “enable\_categorical = True” in command “XGBRegressor”, categorical features are handled with the appropriate way without needing to create dummy variables. For forecasts in August and September once again the future data set (“t”) is used which contains details about stores in these months.

1. **Results**

4.1 Exploratory Data Analysis

Firstly, it is important to point out that I have used Python for data analysis. After replacing the missing values, I calculated some descriptive statistics for each variable and I constructed several plots to illustrate the data. For numeric variables I calculate some statistics such as median value, mean value, standard deviation and range. For categorical variables I cannot calculate descriptive statistics so I create frequency tables.

**Table 2: Descriptive statistics for total sales across all stores, where many outliers are observed**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mean | Median | Standard Deviation | Minimum | Maximum | Skewness | Kurtosis |
| 5774 | 5744 | 3849 | 0 | 41551 | 0.64 | 1.77 |

**Figure 1: Histogram for sales with open stores where most values are concentrated on the left side of the plot**

A graph of sales

AI-generated content may be incorrect.

Table 2 shows that median daily sales across all stores in those two years are around 5744 euros. Large standard deviation is observed because of the fact that stores were closed on non-working days and as a result these stores did not have incomings from sales. Positive skewness means that most values are on low levels and positive kurtosis indicates that most values are concentrated on those levels.

**Table 3: Descriptive statistics for customers across all stores**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mean | Median | Standard Deviation | Minimum | Maximum | Skewness | Kurtosis |
| 633 | 609 | 464 | 0 | 7388 | 1.59 | 7.09 |

Table 3 explains the statistics of daily customers for all stores. Daily buyers are around 609 with some outliers being observed because of the non-working days. Positive skewness means that most values are on low levels and positive skewness shows that a huge number of those values are on those levels.

**Table 4: Statistics for the distance (in meters) between Rossmann’s stores and competitive stores**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mean | Median | Standard Deviation | Minimum | Maximum | Skewness | Kurtosis |
| 5422 | 2330 | 7706 | 20 | 75860 | 2.93 | 13.04 |

Large standard deviation points out that there are many competitive stores either close or far away from Rossmann’s stores. High kurtosis indicates that most competitive stores are close to the median value and as a result Rossmann’s stores deal with close competition.

**Figure 2: Pie chart for store types where it is observed that most stores are associated with beauty and personal care while the least ones are online stores**

A pie chart with numbers and text

AI-generated content may be incorrect.

**Figure 3: Pie Chart for the variety of products that stores provide where balance is observed between basic and extended assortment**

A pie chart with text and numbers

AI-generated content may be incorrect.

Before progressing in the multivariate analysis, it is important to understand the correlation and the dependencies between the most important features. From the statistical tests we understand that:

* There is strong correlation between the sales and the purchasers which is an expected outcome because an increase in the number of customers leads to increased sales (See Tables A1, A2 and Figure A9 in Appendix where Kendall=0,7 & Spearman=0,9)
* Sales are not affected by the distance of competitive stores which means that Rossmann’s stores have built their own standard clientele (See Tables A3, A4 in Appendix where Kendall=-0,01)
* Holidays have an impact on whether promotion will take place or not (See Tables A7, A8 in Appendix where Chi\_square Test – P-value<0,05). However, analysis shows that there is a balance between promotions and holidays. What I mean is that 72% of advertisement takes place during holidays and 68% of promotion happens without holidays which suggests a medium correlation.

**Figure 4: Total sales from 2013 to July 2015 showing balance with some outliers observed**

A blue graph with numbers

AI-generated content may be incorrect.

**Table 5: Characteristics for the highest sales in Rossmann’s retail chain**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Store | Sales | Date | Day | Holiday | Promotion | Continuous Promotion | Store Type | Assortment |
| 909 | 41551 | 22-6-2015 | Monday | No | No | Yes | Beauty | Extended |
| 262 | 38722 | 3-4-2015 | Friday | Easter | Yes | No | Online | Basic |
| 262 | 38482 | 1-5-2015 | Friday | Public | Yes | No | Online | Basic |
| 262 | 38367 | 14-5-2015 | Thursday | Public | No | No | Online | Basic |
| 57 | 38037 | 16-4-2014 | Monday | No | Yes | No | Home | Extended |
| 817 | 38025 | 16-12-2013 | Monday | No | Yes | No | Beauty | Basic |
| 261 | 37646 | 16-12-2013 | Monday | No | Yes | Yes | Home | Extended |
| 262 | 37403 | 29-5-2014 | Thursday | Public | No | No | Online | Basic |
| 262 | 37376 | 22-12-2013 | Sunday | No | No | No | Online | Basic |
| 262 | 37122 | 21-12-2014 | Sunday | No | No | No | Online | Basic |

Figure 4 shows that total sales are balanced with the passage of time. However, there are some dates with unexpected results so I decided to investigate what happened on those dates (See Table 5). We understand that in those days there was balanced advertising and most of the days did not coincide with holidays. Furthermore, online stores had the highest sales with basic assortment being selected mostly. Monday was the main day that these situations happened. From my point of view, these circumstances are not due to random factors since most sales were made for store “262” and generally we observe some specific patterns.

**Figure 5: Sales with promotion on a specific day (Promo) and continuous promotion (Promo2)**

A screenshot of a graph

AI-generated content may be incorrect.

In figure 5 we can see that when promotion on a specific day exists then we have higher sales. On the other hand, when continuous promotion happens then we have unexpected results because sales are lower than not having continuous advertisement. Circles indicate the outliers, boxes show the concentration of most observations and the line explains the median value. It looks like continuous promotion did not have as great performance as promotion in days had.

**Figure 6: Customers per dates**

A blue line graph with white text

AI-generated content may be incorrect.

**Table 6: Information for the highest number of customers in Rossmann’s stores**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Store | Customers | Date | Day | Holiday | Promotion | Continuous Promotion | Store Type | Assortment |
| 817 | 7388 | 22-1-2013 | Tuesday | No | Yes | No | Beauty | Basic |
| 262 | 5494 | 3-10-2014 | Friday | Public | Yes | No | Online | Basic |
| 262 | 5458 | 1-5-2015 | Friday | Public | Yes | No | Online | Basic |
| 262 | 5387 | 9-6-2014 | Monday | Public | No | No | Online | Basic |
| 262 | 5297 | 29-5-2014 | Thursday | Public | No | No | Online | Basic |

Figure 6 reveals some dates with an unexpected number of customers. As a result, I decided to investigate why this happened. In table 6 we can see that store “817” had the biggest number of customers on 22-1-2013 when running promotion on that day. Also, there was not any holiday so we understand that promotion was good enough in order to attract a huge number of customers. The next days with many customers refer to online store “262” when there are public holidays.

**Figure 7: Boxplot for sales per store type and variety of products**

A graph of different colored squares

AI-generated content may be incorrect.

In Figure 6 it is clearly understood that online stores (type b stores) have more sales than the others because their boxes are higher than the others. Extended assortment (C) has most sales for this type of store while there are balanced sales for each product variety across the store types with health products, household items and beauty stores.

**Figure 8: Bar plot for sales for each assortment depending on holidays where we observe that purchasers select extra assortment (b) in holidays**

A graph of different colored bars

AI-generated content may be incorrect.

**Figure 9: Customers for each store type depending on holidays where we observe that online purchases are preferred on holidays**

A screenshot of a graph

AI-generated content may be incorrect.

**Figure 10: Boxplot for purchasers per store type with promotion or not**

A screenshot of a graph

AI-generated content may be incorrect.

Figure 10 presents the number of customers when promotion takes place or not. I observe that boxes when promotion comes about then boxes are higher comparing to not running promotion. Therefore, when running promotion the number of consumers is increased, as expected.

* 1. Time Series models

In this section ARIMA and Prophet methods will be presented. Firstly, I load the needed libraries in Python that will help us with the analysis. I divide the main dataset in two parts. The first contains data before July 2015 (train) and the second includes data for July 2015 (test). I use this method in order to evaluate our model predictions with real sales and then forecast incomings for August and September of 2015. Stationarity is checked with augmented Dickey Fuller test (ADF) where if time series is not stationary then “difference” method is applied. However, as I have already mentioned before, auto\_arima function deals automatically with seasonality so I perform ADF test to see if most of our time series are stationary. In Figure 11 we can observe that there is seasonality in our data because in autocorrelation plot (ACF) for total sales we can see sharp peaks appearing every seven lags (days). However, each store has its own seasonality so I it is better to check seasonality individually.

**Figure 11: ACF plot for total sales across stores, where seasonality is observed every seven lags**

**A graph with blue dots and numbers

AI-generated content may be incorrect.**

In the programming language, I have developed code for forecasting based on total sales by store type, individual shop-level predictions and a function that allows selecting a store from a specific store type and returns a random store. Additionally, I implemented code using log transformation on sales to examine whether the results change, given the presence of many extreme values in this feature.

Figures A10,A11,A12 and A13 in the appendix show that online stores will have increased sales in the next months while stores with household items and health products will have balanced patterns. Beauty stores have a model with unsatisfactory results and as a result we cannot take into consideration its forecasts. All models have huge RMSE values because of the large number of incomings from sales. Plots for online, health and home shops indicate that real values are close to the predictive ones, so we understand that forecasts might be sufficient. However, all the models for store types have a large error, which arises from the ratio of RMSE and mean sales in test data for each shop type, thus it is better to create models for each store separately and not to observe the general trend from these inappropriate models.

Data analysis proves that store 262 (Online Store) had the most sales since 2013 and Store 543 is one of the stores that had lower earnings from sales. Because it is practically impossible to include the forecasts for more than a thousand stores, I decided to compare ARIMA and Prophet models with lower and higher turnovers and a store that has medium earnings.

**Figure 12: ARIMA for sales forecasting in store 262 where poor fit is observed from the plot, however metrics suggest a sufficient model**

A graph of a graph

AI-generated content may be incorrect.

**Figure 13: Prophet for sales predictions in store 262 where satisfactory fit is indicated**

A graph of a graph

AI-generated content may be incorrect.

It is known that some of the assumptions in time series models are uncorrelated and normally distributed residuals. Figure A20 in appendix shows that there is no significant autocorrelation for store “262”. Figure 12 and Figure 13 show the model’s evaluation metrics, the predicted values compared to real values in test data and future predictions for both August and September. Plots A16 and A18 in appendix show graphs with logarithmic transformation in sales because of the large number of outliers. According to Figure A24 in the appendix, seasonality is seen every seven days for this store. Those graphics will help us understand whether it is better to keep the starting data or if it is better to remove the noise. It is crucial to mention that mean sales for store 262 in test data are around 21730 euros. This fact will help us decide if models are appropriate by calculating the ratio between RMSE and mean sales to see the error. In Figure 12 it is observed that ARIMA has inadequate fit and as a result we are not sure that we can accept its forecasts. However, it provides an error around 16% which is acceptable in forecasts. When applying log transformation, RMSE gets a lower value (3263) but there is no big difference with the ARIMA model. On the other hand, Prophet has much lower metrics (RMSE=2529) and it is observed from the plot that it has a great fit on data. Logarithmic transformation for Prophet does not help because there are again many deviations and RMSE is increased. With an RMSE of 2529 and an error around 11% we conclude that prophet has better performance. Therefore, predictions should follow a specific pattern with maximum sales around 30000 euros and minimum incomings around 18000 euros for the next two months. Finally, both models have some deviations on test data probably because they cannot understand the effect from school holidays (12 days in July).

**Figure 14: Sales predictions with ARIMA for store 543, where acceptable fit is observed**

**A graph of sales

AI-generated content may be incorrect.**

**Figure 15: Prophet for store 543 where appropriate model is observed**

A graph of a graph

AI-generated content may be incorrect.

Figure A21 in appendix shows that the assumption of uncorrelated residuals is not rejected for store 543. It is vital to refer that store 543 has seasonality every fourteen days and as a result it is important to change model parameter “m” in python for this case (See Appendix Figure A25). To evaluate the model’s performance, it is important to know that mean sales for store 543 in July were 2267 euros. Figures 14 and 15 present ARIMA and Prophet models, with their forecasts and statistics. In order to reduce the noise from the outliers, I apply logarithmic transformation in data to understand if improved results are provided (See Appendix Figures A17 & A19). According to Figure 14, ARIMA results in a RMSE of 563 euros which leads to an error of 24%. Usually, we cannot accept predictions with that large error because in most cases the most appropriate error is lower than 20%. For this reason, Prophet has better fit than ARIMA, because with a root mean squared error equal to 372 we have an acceptable error of 16%. ARIMA has some deviations in test data probably because of school holidays (5 days on July) and balanced promotion. By applying logarithmic transformation on data, we observe that both Arima and Prophet have higher statistics compared to the starting models. Therefore, it is understood that transformation on data with lower sales and lower outliers does not improve the future predictions and models accuracy. Consequently, Prophet yet again has better performance and its predictions suggest that a pattern will be followed in the next two months with maximum sales around 4000 euros and minimum sales around 2000 euros. Something that I have noticed is that on test data there were days that store was closed but in predictions we do not have days with zero sales. So either the store was closed due to renovation or the predictions cannot recognize the future days that store 543 will be closed. Data analysis indicates that store 543 which sells health products, was closed in 17.1% of days since 2013 and thus it was opened 82.9% of days. We understand that Prophet has recognized that this store is a pharmacy and rarely is it closed.

Table A9 in the appendix shows some experiments for ARIMA and Prophet models for different stores in order to compare their performances. I have compared random stores with high revenues, medium turnover and stores with lower earnings. Some stores had problems with outliers on sales and log-transformation was needed. However, there are some stores that have a great fit without logarithmic transformations. For instance, store 165 has sufficient forecasts for both models (See Appenidx Figures A14, A15). With mean sales equal to 3446, there is an error of 18% for both models, which is acceptable in data science. From other experiments, I have observed that most times Prophet has better accuracy with small errors observed(See Appendix Table A9). Although there are some deviations in forecasts, we observe that models understand the patterns for some shops and their unique characteristics. It looks like there are some cases where either ARIMA or Prophet have great performance and some others that transformation in data is needed. It depends on the special properties of each store and the outliers in turnover.

* 1. Machine Learning models

In this module machine learning models like XGBoost and LightGBM are going to be applied. For both algorithms I have created three different models. The first one includes all the variables, the second one includes the most important factors that occur from feature importance plots and the third model has some variables that I believe have an impact on our data. The target for these models is to check which one has the best fit and provides the most accurate predictions. Hyperparameter tuning helps us to find the optimal parameters for the models, feature importance graphs suggest the most crucial factors that should be included in the model and SHAP explains the output of machine learning models by quantifying the contribution of each feature to individual predictions in a consistent and interpretable way. To calculate models performance we use statistics such as RMSE and MAE, and cross validation helps us identify which algorithm makes better predictions and which one is more constant with different test data. Before applying such methods, it is important to load the needed packages into python and divide our data on test and train data. The dataset was randomly split into training and test sets using shuffling to ensure that data distribution remains representative and to avoid potential biases from sequential patterns.

To develop the best possible model, we start by setting the parameters. Firstly, we create XGBoost and LightGBM models that include all the features. Then Hyperparameter tuning is used which is the process of systematically selecting the best combination of hyperparameters for machine learning models to optimize their performance. After setting some intervals for parameters, then with Grid Search approach we find the most appropriate combinations and we can use them to train the next models. The best combination is the one that has the lowest RMSE. Some information about the parameters used for LightGBM model are:

* *Learning\_rate = 0.1:* Determines how much the model adjusts at each iteration. Lower values lead to more stable but slower learning
* *n\_estimators = 100:* Number of trees
* *max\_depth = 5:* Controls model complexity and helps prevent overfitting
* *min\_data\_in\_leaf = 10:* Increasing this value makes the model more conservative and helps avoid overfitting on noise.
* *num\_leaves = 5:* Controls the complexity of each tree. A higher value can capture more patterns but increases the risk of overfitting

For XGBoost models, the same approach is used as above, and the parameters that occur are:

* *max\_depth = 5:* Higher values can obtain more complex patterns while lower values can reduce overfitting
* *learning\_rate = 0.3:* Smaller values make learning slower but more stable while larger values may cause overfit
* *n\_estimators = 500:* Number of boosting rounds

Furthermore, it is crucial to choose the best possible model by comparing statistics and by using cross validation method. For each algorithm, I create models with all the features (Despite customers that have strong correlation with sales and can cause bias), another one with the most important variables as feature importance plots suggest and the last one with some variables that in my opinion have a huge impact on sales. Also, I convert the target variable (sales) with logarithmic transformation to obtain whether we have improved accuracy. Unfortunately, this approach leads to disappointing results because there is such a poor fit on test data. Table A10 on appendix shows cross validation information for each model. A lower RMSE indicates better accuracy while a smaller standard deviation shows how constant an algorithm is across different test datasets. As a result, both LightGBM and XGBoost models, including all the variables, have better accuracy because of the lower root mean squared errors compared to the others. The first one has RMSE equal to 2201 while the latter has RMSE equal to 885. Also, all the experiments have low standard deviation so there is no problem in terms of this. After choosing the best algorithms, we continue this survey by applying SHAP on models. SHAP is a powerful method for explaining machine learning model predictions and it helps us understand how much each feature contributes to a specific prediction or overall model behavior. In Appendix SHAP plots are presented for each machine learning algorithm (See Appendix Figures A26 & A27). Before writing my observations, it is of vital importance to mention how to interpret such plots. Positive SHAP values push predictions higher while lower SHAP values decrease the predictions. Red indicates high feature values while blue shows low variables values. As a result, it is understood for LightGBM algorithm (See Appendix Figure A26) that closed stores push predictions lower while open shops increase the forecasts, which is reasonable because when a store is closed it has zero sales. Competitive stores that are close to Rossmann’s retail chain push sales down and it is understood that both promotion on a specific day (Promo) and assortment have a strong influence on earnings. For XGBoost (See Appendix Plot A27) closed or open stores, promotion and the distance from competitive shops are again dominant features that have the same explanation as they have for LightGBM. Moreover, there are certain times of the year (DayOfYear) that impact sales significantly, suggesting seasonal effects.

Now it is time to compare each model’s accuracy by predicting incomes from sales and calculating statistics for each store. Metrics like RMSE, MAE and MAPE will be measured by test data that includes the actual sales for July. MAPE gets enormous values because it is sensitive with zero sales when stores are closed. This situation cannot be changed because closed stores is a realistic factor in this data set so we should evaluate models accuracy exclusively from RMSE. In my opinion, it is better to compare stores “262”, “165” and “543” as I did with ARIMA and Prophet models. With this comparison we could understand which model has better accuracy and what problems occurred for each approach. Also, these stores have high, medium and low incomes from sales respectively and we could understand for instance if models have some deviations when many sales or lower sales are registered. Tables A11 and A12 in the Appendix show the summary for root mean squared errors per store with LightGBM and XGBoost algorithms respectively. These tables are important because an overall trend is presented to understand the sufficiency of each model. It is clearly understood that XGBoost has better fit than LightGBM for many reasons. The first one, has almost half median and mean value compared to the latter which indicates lower errors in predictions. Standard deviation is smaller and as a result there are not many outliers in root mean squared errors which shows that most values are closer to median value on comparison to LightGBM where large deviation is observed. There are differences in minimum and maximum values, but most importantly XGBoost outperforms LightGBM in 25th and 75th percentile. In the best performing quarter of predictions (lowest RMSEs), XGBoost performs better than LightGBM, with smaller errors. Furthermore, even in the higher percentiles, XGBoost still maintains lower prediction errors which makes it more sufficient model.

Table A12 in the Appendix shows some experiments for LightGBM and XGBoost algorithms across different shops. We can observe the store identification number, RMSE for both models, mean sales on test data and the best error with the respected model. Error is calculated by RMSE/mean sales for store and usually values lower than 20% are accepted. According to table A12, XGBoost has lower RMSEs than LightGBM which shows that forecasts are closer to actual values on test data providing more accuracy. As a result, it leads to lower errors, some of which are acceptable while some others are not. Most experiments have acceptable errors (<20%) suggesting that XGBoost has great fit on data.

**Figure 16: LightGBM algorithm for store 262 where an inappropriate fit is observed**

A graph of a graph

AI-generated content may be incorrect.

**Figure 17: August and September predictions for store 262 with LightGBM, where a specific pattern is followed**

A graph showing sales

AI-generated content may be incorrect.

**Figure 18: Actual turnover and forecast for Store 543 with LightGBM model where significant deviations can be seen**

A graph of a graph

AI-generated content may be incorrect.

**Figure 19: LightGBM predictions for store 543 where a specific trend takes place**

A graph with green lines

AI-generated content may be incorrect.

Figures 16 and 18 show LightGBM’s evaluation by comparing the actual incomings from sales to forecasts. Figures 17 and 19 indicate the future predictions for stores 262 and 543 respectively. Firstly, it is crucial to refer that mean sales for test data (July) for store 262 are 21730 euros and for store 543 mean incomings are 2267 euros. Mean sales will help us calculate the error for each model and decide whether it brings acceptable forecasts or not. Plots 16 and 18 show high deviations between real sales and predictions and as a result these models are insufficient. This fact is validated by the errors equal to 32% and 165% respectively which suggest unacceptable models. With these metrics we cannot take into consideration the predictions for each model (See Figure 17 and Figure 19).

Plots 20 and 22 show XBoost’s accuracy on test data (July) by comparing the real turnover with the predictions. Graphs 21 and 23 show August and September predictions for stores “262” and “543”. Figure 22 shows an acceptable model with small deviations however, its error which is equal to 32%, suggests insufficient results in data science. On the other hand, store “262” has an acceptable algorithm with a small error equal to 10% despite some deviations in the plot (See Figure 20). Therefore, we cannot trust the predictions for store 543 but we can for store 262. The latter store deals with some bad news. With given future factors (promotion days, holidays etc.), its incomings from sales are about to decrease dramatically in the following two months according to XGBoost model. Comparing these predictions with Prophet’s we can see that both models understand that sales are going to be reduced. However, Prophet shows future sales between 18000 euros and 30000 euros while XGBoost predicts a turnover between 4500 euros and 8000 euros. Analysis has proved that there are some stores with some negligible negative predictions just a little lower to zero. I set these values equal to zero because negative predictions are not reasonable values. What I have read from sources is that machine learning models are sensitive with zero sales and as a result they might provide inaccurate forecasts because predictions are prevented from increasing. Days with zero turnover are absolutely reasonable values in this data set and I cannot handle them as missing values. Also, I applied a logarithmic transformation in the model but there were still negative predictions. From my point of view, in this case we should accept Prophet’s predictions because of the aforementioned reasons.

**Figure 20: XGBoost algorithm for store 262 where there are some small deviations between real sales and predictions**

A graph with blue and orange lines

AI-generated content may be incorrect.

**Figure 21: August and September forecasts for store 262 where XGBoost indicates that the store will deal with reduced earnings**

A green line graph with numbers

AI-generated content may be incorrect.

**Figure 22: XGBoost model for store 543 where acceptable performance is observed**

A graph with blue and orange lines

AI-generated content may be incorrect.

**Figure 23: Predictions for store 543 based on XGBoost where different trends are observed across every day**

A graph showing sales

AI-generated content may be incorrect.

Last but not least, I have decided to investigate another experiment in order to compare LightGBM and XGBoost. Figures A28 and A30 in the Appendix, show LightGBM and XGBoost algorithms for store 165. On July this store had mean sales around 3442 euros. A root mean squared error equal to 2122 for LightGBM and a RMSE of 591 for XGBoost, lead to errors of 61% and 17% respectively. Once again, we accept the accuracy of XGBoost and thus we can understand that August and September sales are about to increase (See Figure A31). Comparing predictions with Prophet’s, it is clearly observed that XGBoost brings more increased sales. This situation contradicts with what happened in store 262. Machine Learning model has an error rate of 17% while both ARIMA and Prophet have errors equal to 18%. However, the difference between predictions for each model are not that far as in the previous situation, so we can understand that the best model selection depends on the unique factors of each store and the methodology each model uses to identify store trends and patterns.

1. **Discussion**

Taking everything into account, there are many significant findings from this survey. From univariate analysis, it is understood that most shops in dataset are beauty and personal care stores with most shops containing basic variety of products. As expected, customers and sales have a strong correlation which means that if stores could find a way to increase their purchasers then incomings from sales should increase. In the past two years, Rossmann’s retail chain seems to have balanced number of customers and sales so a way should be found to keep these numbers high. Promotion on a specific day increases the sales while continuous promotion does not have the expected results because there is a balance between sales with this kind of promotion or not. Therefore, the owners should start finding ways to improve continuous promotion to have improved results. Also, online stores were having the most sales while containing only 1.6% of the dataset in terms of store types. As a result, the chairman of Rossmann’s stores should think about expanding online stores in order to increase their turnover.

The main task of this project was to develop time series and machine learning models to forecast incomings from sales for August and September of 2015. The models that I used are ARIMA, Prophet, LightGBM and XGBoost. Each one provides important findings while having some limitations in methodology. It is important to mention that all models have both advantages and disadvantages. ARIMA requires uncorrelated and normal distributed residuals, something that is not achieved in most time series in this research. Rarely it provides better accuracy than Prophet, however we cannot take into account its predictions by the time it does not meet the above assumptions. Prophet has better evaluation than ARIMA because it provides lower metrics (RMSE) and as a result more reliable forecasts. It understands better holiday effects and models these cases with an appropriate way. In most series, logarithmic transformation on Prophet does not improve the metrics compared to the starting Prophet. Also, some negligible negative predictions are observed (predictions close to zero) but I have decided to set these values equal to zero. This fact is reasonable by the time there are so many non-working days on the dataset. Therefore, it is understood that Prophet is sensitive with outliers and correct regressors should be selected in order to avoid increased negative predictions. In terms of machine learning algorithm, it is understood that XGBoost has the best fit in Rossmann’s stores data. After developing four models for each algorithm, I concluded that the most accurate ones are those including all the features. Hyperparameter tuning gets enough time to find the best combinations between the parameters, so I decided to create a model that includes parameters from tuning and another model that had the default parameters. This experiment provided better results for LightGBM but not for XGBoost. This is an important observation because when having so many stores it is not easy to wait and spend time on tuning. So it is of vital importance to know that even the default parameters help the accuracy to be improved. SHAP (SHapley Additive exPlanations) plots help us understand which features impact both the predictions and the models behavior. In this case it is important to understand that close stores decrease the predictions let alone when we have negative forecasts. In all the experiments, XGBoost has better performance because of the lower metrics that occurred from many methods like cross validation and RMSE.

1. **Conclusion**

In this analysis, we explored key factors influencing sales performance across different stores. What we have learnt is that descriptive statistics and diagrams for features contribute us to understand patterns between variables and provide us with some important findings. We cannot handle outliers in sales since they represent logical values when stores are closed. As a result, we need to manage extreme values effectively in the models so that they can understand these values properly. Feature engineering helps to combine important variables and model the data with them to achieve optimal results. Prophet has significantly better performance (lower evaluation metrics) than ARIMA and handles better seasonality and holiday effects. “auto\_arima” command in python helps to find model with the lowest AIC and with the best parameters which prevents us from applying multiple experiments and spend valuable time. Prophet is sensitive with outliers because when I add a regressor with many outliers (for instance the number of customers) then negative predictions arise which are not acceptable. After performing many experiments in XGBoost and LightGBM models I observe that XGBoost brings improved results compared to LightGBM. XGBoost’s metrics are balanced and there are small deviations between real values and predictions. The problems that occurs with these models, are in the forecasts where some negative values are obtained. However, these values are just a little lower than zero and are caused due to non-working days. As a result, models are sensible with these values and I have to make sure that I set these forecasts equal to zero. Hyperparameter tuning with grid search method takes some time, however it is important in order to find the most appropriate parameters for each model. Feature importance plots help us identify the most crucial factors that affect sales, which is important when having large data sets. With cross validation method we can compare machine learning models and understand which one makes better forecasts and which one adapts better with different test data. Taking everything into consideration, it is understood that machine learning models have better fit than time series models because of the fact that we have many tools to control the accuracy of the models and that everything has gone correctly with features and parameters selection.

In terms of future research, there are a couple of things that could be done to improve the methodology. First and foremost, it would be helpful if I knew the exact dates that stores were under refurbishment so as to control the impact of these days without sales. More crucial features could result in more appropriate results. Such variables could be periods with discounts to understand if discounts increase the incomings from sales, local events like concerts or festivals, touristic periods to see if there is an impact on sales, weather conditions to check if bad conditions could affect sales, product’s quantity levels, competitor’s activities for instance if a competitive store was running promotion and socio-political events. Moreover, we could try more models in order to compare their performance with the aforementioned models. “Catboost” is a machine learning model that handles categorical variables with a more appropriate way and “NeuralProphet” is a statistical model that combines time series and machine learning algorithms. “NeuralProphet” can analyze complicated patterns and it can deal with large datasets.

1. **References**

* Tong Zhou, *Improved sales forecasting using trend and seasonality decomposition with LightGBM*, Department of computer science – Johns Hopkins University
* A. Chowdhury et al, Accuracy comparison of ARIMA and XGBoost forecasting models in predicting the incidence of COVID-19 in Bangladesh, Department of statistics – Begum Rokeya University
* E. Hossin, *Forecasting Sales Trends Using Time Series Analysis: A Comparative Study Of Traditional And Machine Learning Models* , University of Portsmouth

Further References

* Y. Ensafi, S. Amin, G. Zhang, B.Shah, *Time-series forecasting of seasonal items sales using machine learning – A comparative analysis*, Ryerson University & University of Winsdor
* H. Chang, Z. Yang, Y. Zhang, *Forecasting daily freight flow in cold regions of China using the hybrid Prophet model considering the importance of festivals and epidemic prevention policy*, Harbin University
* M. Angelo et al, *Comparative Analysis of ARIMA and Prophet Algorithms in Bitcoin Price Forecasting,* Bina Nusantara University
* W. Darmawan et al, *Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET,* Bina Nusantara University
* Candice Bentejac et al, *A comparative analysis of XGBoost*, University of Bordeaux
* T. Finley, W. Ma et al, *LightGBM: A Highly Efficient Gradient Boosting Decision Tree,* Peking University
* B. Kumar, T. Kumar, *Analysis of ML based Gradient Boosting Algorithms: XGBoost, CatBoost, and LightGBM*, Columbia University

**Appendix**

Figure A1: Bar plot for years that continuous promotion (Promo2) started where we observe that most stores started in 2011

**A graph with blue rectangles

AI-generated content may be incorrect.**

Figure A2: Pie chart that describes the interval that continuous promotion started where we observe that most stores started promotion in January, April, July and October

**A pie chart with numbers and a triangle

AI-generated content may be incorrect.**

Figure A3: Bar plot showing years that competitive stores were opened where it is understood that most ones were opened in 2013

**A bar graph with blue squares

AI-generated content may be incorrect.**

Figure A4: Bar plot showing months that competitive stores were opened where it is observed that most ones were opened in September

**A graph of a bar chart

AI-generated content may be incorrect.**

Figure A5: Boxplot for sales per store type with or without continuous promotion which indicates that there are more sales without continuous promotion (blue box higher that orange box and the same for median values)

**A graph of a sales report

AI-generated content may be incorrect.**

Figure A6: Boxplot showing customers per store type and assortment where most purchasers select online stores (b) and at the same time they prefer extended variety (C)

**A graph of different colored squares

AI-generated content may be incorrect.**

Figure A7: Boxplot for customers per store types with or without promotion where plenty of customers are observed when promotion takes place

**A screenshot of a graph

AI-generated content may be incorrect.**

Figure A8: Boxplot for customers per store type with or without continuous promotion where there is a balance between each category, with most buyers observed in online stores without continuous promotion

**A graph of different colored squares

AI-generated content may be incorrect.**

Correlation between variables

In order to check the correlation between features, I build linear models where I check the following assumptions (a=5%) :

* Normally distributed residuals (Kolmogorov Smirnov Test because n>50)
* Homoskedasticity check with Levene Test
* Linearity
* Autocorrelation between residuals with Durbin Watson Test

If the above assumptions are not met, then I use Kendall’s and Spearman’s methods to calculate the correlation between factors.

*Sales ~ Customers*

Table A1: Shapiro Wilk Test where residuals are not normally distributed because p-value is lower than a=5% and as a result I reject the null hypothesis of normality

|  |
| --- |
| Shapiro Wilk Normality Check (a=5%)  P-value=0  Statistic= 0.12 |

Table A2: Correlation methods where strong correlation is indicated between sales and customers

|  |
| --- |
| Correlation Methods  Kendall = 0.74  Spearman = 0.9 |

Figure A9 : Correlation plot for sales and customers where a line is shaped indicating strong correlation

A graph showing sales and customers

AI-generated content may be incorrect.

*Sales ~ CompetitionDistance*

Table A3: Normality check for the residuals of the model with sales and competition distance where residuals are not normally distributed because p-value<a

|  |
| --- |
| Shapiro Wilk Normality Check (a=5%)  P-value=0  Statistic= 0.09 |

Table A4: Correlation methods where no correlation is indicated between features

|  |
| --- |
| Correlation Methods  Kendall = -0.01  Spearman = -0.02 |

*Customers ~ CompetitionDistance*

Table A5: Normality check for the residuals of the model with customers and competition distance where we reject null hypothesis because p-value<a

|  |
| --- |
| Shapiro Wilk Normality Check (a=5%)  P-value=0  Statistic= 0.07 |

Table A6: Correlation methods where no correlation is observed between variables

|  |
| --- |
| Correlation Methods  Kendall = -0.11  Spearman = -0.17 |

*Promo ~ SchoolHoliday*

Table A7: Chi square test to check the dependence between promotion and school holidays where it is shown that holidays have an impact on whether promo will take place or not

|  |
| --- |
| Chi Square Test (a=5%)  P-value=0  Chi Statistic= 4631 |

*Promo ~ StateHoliday*

Table A8: Chi square test to check the dependence between promotion and holidays where it is shown that state holidays have an impact on whether promo will take place or not

|  |
| --- |
| Chi Square Test (a=5%)  P-value=0  Chi Statistic= 2948 |

Time Series

ARIMA - Prophet

Figure A10: Total sales predictions for beauty stores where we observe that performance is unacceptable because RMSE value is enormous

A graph of a graph

AI-generated content may be incorrect.

Figure A11: Forecasts for online stores where increasing trend is observed for the next two months, with medium model performance

A graph of a graph

AI-generated content may be incorrect.

Figure A12: Plot describing that real sales and predictions are close and there is a balanced trend in future sales with some ups and downs

A screenshot of a graph

AI-generated content may be incorrect.

Figure A13: In stores with household items we observe that the model has great performance and future forecasts have a balanced pattern

A graph of different colored lines

AI-generated content may be incorrect.

Figure A14: ARIMA model for store 165 which has great fit and a balanced trend is observed in forecasts

A graph of sales

AI-generated content may be incorrect.

Figure A15: Prophet model for store 165 which is adequate and a specific pattern is indicated in predictions

A graph of different colored lines

AI-generated content may be incorrect.

Figure A16: Arima model with log-transformation for store with most revenues where some deviations can be seen

A graph of a graph

AI-generated content may be incorrect.

Figure A17: ARIMA model with log-transformation for store 543 where small deviations are observed on test data and decreasing forecasts A graph of sales

AI-generated content may be incorrect.

Figure A18: Prophet with logarithmic transformation for store 262

A graph of a graph

AI-generated content may be incorrect.

Figure A19: Prophet with logarithmic transformation for store 543

A graph of sales

AI-generated content may be incorrect.

Figure A20: ACF plot for store 262 where most of the spikes lie outside the confidence bounds, indicating significant autocorrelation at those lags (Ljung Box Test- P\_Value =0.1>0.05)

A graph with blue dots and numbers

AI-generated content may be incorrect.

Figure A21: ACF plot for store 543 where many spikes are outside the confidence bounds, suggesting significant autocorrelation at those lags (Ljung Box Test – P-value=0.1 >0.05)

A graph with blue dots and numbers

AI-generated content may be incorrect.

Figure A22: Autocorrelation plot for store 165 where most spikes are outside the confidence bounds, showing significant autocorrelation at those lags (Ljung Box Test – P\_value =0.7 >0.05)

A graph with blue dots and numbers

AI-generated content may be incorrect.

Figure A23: Seasonality check for store 165 which is observed every seven days because of the spikes

A graph with blue dots and numbers

AI-generated content may be incorrect.

Figure A24: Seasonality test for store 262 which is observed every seven days because of the spikes in ACF plot

A graph with blue dots and numbers

AI-generated content may be incorrect.

Figure A25: Seasonality check for store 543 which is indicated every fourteen days because of the large spikes

A graph with blue dots and numbers

AI-generated content may be incorrect.

Table A9: Table with Arima and Prophet experiments for different stores

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STORE | MEAN | ARIMA-RMSE | PROPHET-RMSE | ARIMA-LOG RMSE | PROPHET-LOG RMSE | BEST MODEL -ERROR |
| 817 | 18179 | 18644 | 2231 | 8645 | 5013 | **Prophet:12%** |
| 562 | 17526 | 1450 | 1603 | 1494 | 2155 | **ARIMA& LOG-ARIMA :8%** |
| 1114 | 20212 | 11992 | 3027 | 9951 | 6207 | **Prophet:15%** |
| 251 | 15816 | 5324 | 1808 | 6342 | 3030 | **Prophet:11%** |
| 543 | 2267 | 687 | 372 | 1002 | 792 | **Prophet: 16%** |
| 198 | 2457 | 618 | 742 | 1283 | 864 | **ARIMA:25%** |
| 200 | 6461 | 2281 | 1112 | 2496 | 1226 | **Prophet: 17%** |
| 163 | 5701 | 7235 | 690 | 1616 | 1348 | **Prophet: 12%** |
| 11 | 6512 | 1409 | 1156 | 2961 | 1529 | **Prophet: 17%** |
| 170 | 3562 | 2414 | 840 | 1491 | 803 | **Log-Prophet: 22%** |
| 500 | 5565 | 1933 | 883 | 2766 | 1022 | **Prophet: 15%** |

Machine Learning Algorithms

LightGBM & XGBoost

Table A10: Table for cross-validation across different models

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Cross Validation -RMSE | Cross Validation- Standard Deviation | Features |
| LightGBM | 2272 | 123 | All (Without customers)– Hyperparameter tuning |
| LightGBM | 1689 | 141 | All (without customers) – Default model parameters |
| LightGBM | 3664 | 150 | DayOfYear, CompetitionOpenSinceMonth, CompetitionDistance, CompetitionOpenSinceYear |
| LightGBM | 3160 | 119 | Promo, Promo2, StoreType, Assortment, StateHoliday, SchoolHoliday |
| XGBoost | 1270 | 154 | All (without customers)– Hyperparameter tuning |
| XGBoost | 1387 | 160 | All (without customers)- Default model parameters |
| XGboost | 2492 | 117 | Promo2, Promo, Open, Promo2SinceYear, Assortment, StoreType |
| XGBoost | 3155 | 118 | Promo2, Promo, SchoolHoliday, StateHoliday, Assortment, StoreType |

Figure A26: SHAP plot for selected LightGBM model

A graph of different colored lines

AI-generated content may be incorrect.

Figure A27: SHAP plot for chosen XGBoost algorithm

A graph of a graph with text

AI-generated content may be incorrect.

Table A11: Descriptive statistics for RMSE per store with both LightGBM and XGBoost models including all the features

|  |  |  |
| --- | --- | --- |
| Statistics | RMSE- LightGBM | RMSE- XGBoost |
| Mean | 1325 | 897 |
| Median | 1114 | 776 |
| Standard Deviation | 755 | 458 |
| Minimum | 376 | 308 |
| Maximum | 9090 | 5772 |
| 25% | 839 | 629 |
| 75% | 1555 | 1014 |

Table A12: Table with LightGBM and XGBoost experiments for different stores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STORE | MEAN | LightGBM-RMSE | XGBoost-RMSE | BEST ERROR  (RMSE/MEAN) |
| 817 | **18179** | **2158** | **1473** | **XGBoost: 8%** |
| 562 | **17526** | **1958** | **1147** | **XGBoost:6%** |
| 1114 | **20212** | **9090** | **3519** | **XGboost: 17%** |
| 251 | **15816** | **3532** | **1331** | **XGBoost: 8%** |
| 543 | **2267** | **2981** | **1695** | **XGBoost: 74%** |
| 198 | **2457** | **1591** | **468** | **XGBoost: 19%** |
| 200 | **6461** | **1107** | **839** | **XGBoost: 13%** |
| 163 | **5701** | **959** | **785** | **XGboost: 14%** |
| 2 | **4315** | **879** | **456** | **XGboost: 11%** |
| 800 | **4164** | **2281** | **852** | **XGBoost: 20%** |
| 3 | **5920** | **732** | **584** | **XGBoost: 9%** |
| 7 | **8966** | **2198** | **1588** | **XGBoost: 17%** |

Figure A28: Forecasts vs Actual sales for store 165 with LightGBM algorithm where big deviations are observed

A graph of blue and orange lines

AI-generated content may be incorrect.

Figure A29: August and September predictions for store 165 with LightGBM model where fluctuations are indicated

A graph with green lines

AI-generated content may be incorrect.

Figure A30: Real sales and predictions for store 165 with XGBoost, where an adequate performance is observed

A graph with blue and orange lines

AI-generated content may be incorrect.

Figure A31: Forecasts for store 165 provided by the XGBoost model where fluctuations are shown

A graph showing the growth of sales

AI-generated content may be incorrect.