Sales forecasting for the European drug store Rossman

Table of contents

1 Introduction

Predicting sales is a vital part for any business across all sectors, from manufacturing, retail, logistics, to wholesale. However, this is one of the most difficult tasks a business can undertake due to the complexities involved. Sales are driven by a great deal of different factors such as the store location, proximity to competition, macro scales of yearly seasonality, to the micro scales of the time of day and the day of the week, whether there was a promotion or what the weather is doing (Hasan, 2024). All of these things influence sales in different ways, so as you can see, this makes forecasting sales the ultimate challenge for a business.

1.1 Historic trend in sales

As a company Rossmann, a part of the A S Watson group, is the market leader for health and beauty retail in Germany with around 100 stores. It also has over 4,500 stores across Europe, from Poland, Turkey to Spain, employing over 60,000 people (Group, 2024). We have been asked to

2 Methodology

2.1 Data cleaning

The following will describe the processes involved in the preparation required to enable the data to be used in the various modelling techniques. There were three datasets provided: -

1. Store data
2. Train data
3. Test data

Each required various and different cleaning and preparation steps and each will be set out below.

2.1.1 Store data

It was decided that the changes to be made to the ‘store type’ and ‘assortment’ (category of the range of products held by a store) would be converted to numeric categories from alpha-characters (a, b, c, etc). This was so that the predictive power of the model(s) were as good as possible and the data was easier to manipulate. Missing values for the ‘competition distance’ variable were imputed using the mean of all other distances. The ‘competition open since month’ and ‘competition open since year’ variables contained too many missing values (30%), and there was no corresponding variable in the other data, so was removed entirely. It was considered that another option would have been to impute the missing data with an estimation calculated from the other data, however, this was in the end discarded due to the inevitable inaccuracy that would have been introduced considering the number of missing values. The ‘promo2sinceweek’, ‘promo2sinceyear’ and ‘PromoInterval’ variables were amended so that the binary was updated in accordance with whether there was a promotion running at the date of the observation.

2.1.2 Train data

The train dataset would be used to train the chosen models that would predict sales required cleaning as follows. The 'dayofweek' variable was missing a relatively small number of observations which were also randomly distributed throughout the entire dataset. This meant that the decision was made not to impute the missing values. The 'date' variable was split into 3 new variables (keeping the original) that consisted of one each for day, month and year. This was done because we envisaged that each would have a separate and differing level of impact on the sales. Within the 'open' variable, there were a number of stores that were stated as 'open == 0'. The rows for these observations were dropped from the data as the store remained closed throughout and sales would therefore skew towards 0. The 'stateholiday' variable required the creation of two new variables derived from it. The first was changing 'none' to d so that it could be treated as a categorical variable, and the second was to create one that contained a boolean for either holiday or not holiday. It was considered that the Christmas and Easter holidays could skew sales, but not sufficiently to force them to be treated differently from other state holidays.

2.1.3 Test data

Once the best sales prediction model was chosen, it would have to be tested on data that we 'did not know' the sales for. This required cleaning so that it was as useful as possible. The 'date' variable was split into the three components (day, month, year). again, because we believed that each would have a different impact on sales. he 'open' variable in this data contained some missing data (4). It was decided to infer 'open == True' for these, due to the related 'promo == T' variable.

After each dataset had been cleaned, and all data types had been converted etc., the 'store' and 'train' datasets needed to be joined. This was done using a simple join on the 'store_id' field.

2.2 Exploratory data anaysis

Once the data had been cleaned, various simple plots were produced so that distributions and outliers could be considered if observed.

Firstly, a distribution of the sales figures was plotted and can be seen in Figure 1. It shows that there are a large number of €0 sales values. These were often on a Sunday when a store was closed, so were not removed as they were related to the dayofweek variable and were not going to skew results, in fact were vital for a more accurate prediction. All other sales results were relatively normally distributed.

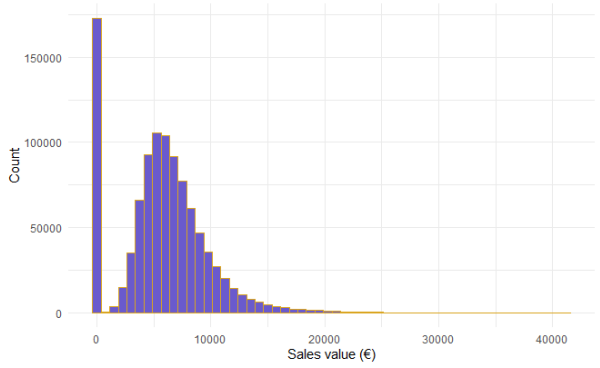

Figure 1: Distribution of historic sales

Figure 2 shows the distribution of customer counts. Again, there were a large number of 0 values, however these are due to a store not being open on a Sunday, and the number of stores that did not open on a Sunday were large. Most stores observed between 0 and around 2500 customers in total.

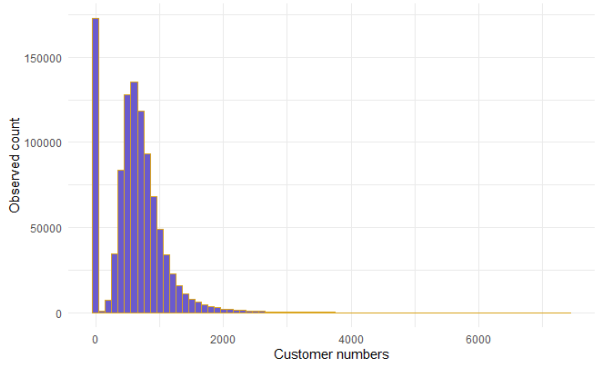

Figure 2: Distribution of historic customer counts

Figure 3 shows that the majority of stores were located relatively close to competition. Considering the nature of the business this shows both that the stores are located in areas with a large number of shops around them, so for example, high streets etc., but also that very few are in isolated locations by comparison to competitors.

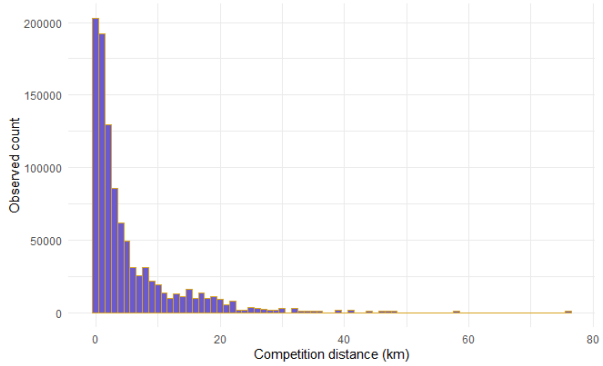

Figure 3: Competition distances counts

Once these initial observations had been carried out, and we were happy that the data looked reasonable and fit for modelling, so that we could visualise the correlation between the variables, a correlation plot was produced. This can be seen in Figure 4.

Here we can see that there was a relatively high correlation between the Customers/Sales variables, and the Promo2/Promo2SinceYear/ YearPromo/Promo2SinceWeek variables. Variables to be removed would be decided upon whether the correlation was above 0.75. This allowed us to remove some of these variables that displayed collinearity from the model. For example, Sales could easily be predicted by Customers, however, there are many factors at play in addition to this. In order to draw meaningful conclusions from our models, the 'Customers' and other variables displaying collinearity were removed (we couldn't remove 'Sales' as this was required to be the dependent variable in our model(s)). However, the one exception to this was to leave the WeekOfYear/Month relationship in the model, as these would be important. For example they would naturally be correlated due to the 1st week of the year always appearing in January, and this would affect sales as well.

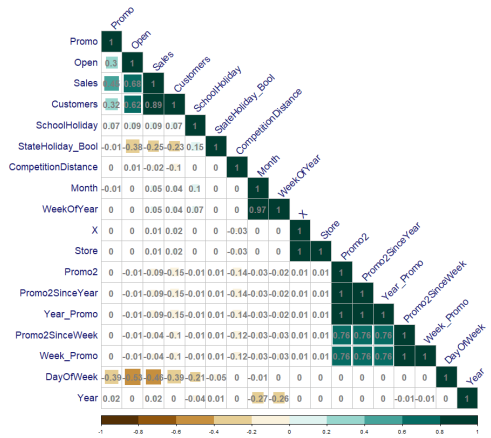

Figure 4: Correlation plot show collinearity

Once all of the variables had been removed, another correlation plot was produced just to check the validity of our decisions. This can be seen in Figure 5. This shows that all (except WeekOfYear/Month) correlations are now under 0.75 and do not have near perfect linear relationships.

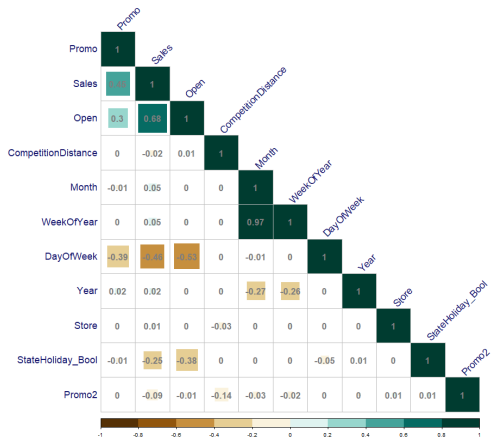

Figure 5: Correlation plot after multicollinearity was addressed

Table 1 shows the remaining variables and their degree of Variance Inflation Factor (VIF). This was showing that all the remaining variables GVIF values are under 5, and points to the measure of the relationship between each variable (Akinwande et al., 2015).

Table 1: Variance Inflation Factors for the linear model

	GVIF	Df	GVIF^(1/(2*Df))
DayOfWeek	1.677217	1	1.295074
Open	1.834113	1	1.354294
Promo	1.201859	1	1.096293
StateHoliday_Bool	1.304303	1	1.142061
Year	1.079366	1	1.038925
Month	15.586181	1	3.947934
StoreType	2.310017	3	1.149748
Assortment	2.256575	2	1.225639
CompetitionDistance	1.069460	1	1.034147
Promo2	1.039179	1	1.019402
WeekOfYear	15.472296	1	3.933484
Store	1.006353	1	1.003172

The final visualisation to be used in order that the correct modelling method was used, was to produce a heat map that shows the level of linearity in the relationships between the variables. As can be seen in Figure 6 map the majority of the relationships have a low linearity.

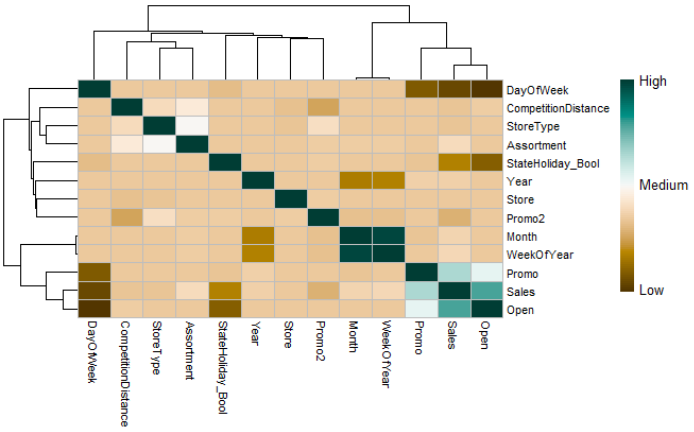

Figure 6: Heat map showing level of linear relationship

Once this table was analysed and we were happy that the remaining variables would not skew the results of any regression or other predictive models, we could then attempt to investigate which of the chosen models was able to best predict sales.

The first model to be used to predict sales was a simple linear regression model. Each variable was used to predict sales and Figure 7 shows the results of the model. As we can see, by looking at the p-values, it was clear that all variables included in the model are significant, or that the probability of obtaining the observed results by chance was very low. The R^2 value was around 0.56, or that the model can explain around 56% of the variability in the target variable (Sales in this case).

```
Call:
lm(formula = formula, data = mdata)

Residuals:
   Min     1Q Median     3Q    Max
 -9955  -1552   -249    919  34783

Coefficients:
                    Estimate Std. Error  t value Pr(>|t|)
(Intercept)        -3.285e+05  6.759e+03  -48.596  < 2e-16 ***
DayOfWeek          -1.529e+02  1.628e+00  -93.937  < 2e-16 ***
Open                5.465e+03  9.055e+00  603.532  < 2e-16 ***
Promo               2.084e+03  5.661e+00  368.217  < 2e-16 ***
StateHoliday_Bool  -1.173e+03  1.667e+01  -70.345  < 2e-16 ***
Year                1.635e+02  3.356e+00   48.718  < 2e-16 ***
Month               6.656e+01  7.844e-01   84.856  < 2e-16 ***
StoreTypeb          5.043e+03  2.953e+01  170.762  < 2e-16 ***
StoreTypec         -9.352e+01  7.705e+00  -12.139  < 2e-16 ***
StoreTyped         -1.648e+02  5.886e+00  -27.997  < 2e-16 ***
Assortmentb        -2.953e+03  4.036e+01  -73.175  < 2e-16 ***
Assortmentc         6.971e+02  5.254e+00  132.681  < 2e-16 ***
CompetitionDistance -1.929e-02  3.370e-04  -57.233  < 2e-16 ***
Promo2             -6.222e+02  5.120e+00 -121.516  < 2e-16 ***
Store               6.086e-02  7.826e-03    7.777 7.44e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2533 on 1017194 degrees of freedom
Multiple R-squared:  0.5672,    Adjusted R-squared:  0.5672
F-statistic: 9.522e+04 on 14 and 1017194 DF,  p-value: < 2.2e-16
```

Figure 7: Summary of linear regression model

Figure 8 shows the distribution of the residuals from the linear regression model. Residuals represent the difference between the actual values and those predicted by the model. This distribution of the residuals here show that there is a fairly large variation between the actual and predicted values.

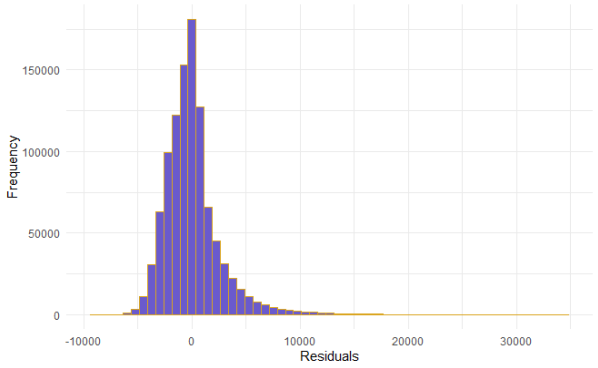

Figure 8: Plot of residuals for the linear model

Another useful way to visualise the effectiveness of the model to predict sales is to produce a QQ-Plot. The plot in Figure 9 effectively plots how well the residuals produced by the model match what we would expect from normally distributed data. As we can see it shows that up to a certain point (~1.25) match what we would expect from a normal distribution, and after that point the quantiles are greater than expected from normally distributed data. This suggests that there is a relatively large amount of data that is non-linear.

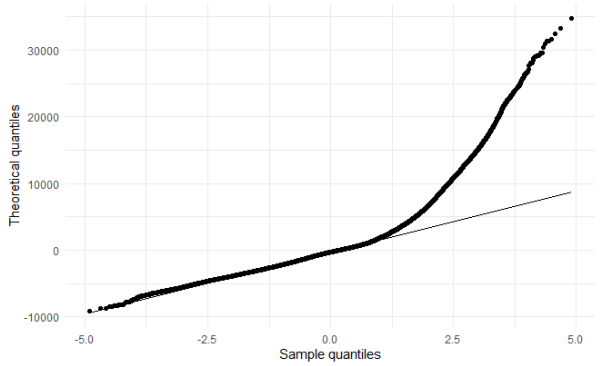

Figure 9: Q-Q Plot of Residuals

Before we could attempt to produce results from various non-linear modelling techniques, we had to split the data into separate train and validation sets. The train data consisted of around 70% of the data and the validation data consisted of the remaining 30%.

2.2.1 Decision Tree

The first non-linear model to be tested is know as a Decision Tree. These are effectively hierarchical models that use a series of decisions to produce various results, culminating in consequences of those decisions based upon likelihood of things like the chance of event outcomes for example.

Firstly a `tune_grid` was created to create a dataframe from all combinations of the variables in the data provided to the model. This is part of the hyperparameter optimisation which attempts to yield the optimal number of branches in the decision tree. The model was then trained using the `rpart` method which is a regression classification method, on the train dataset

Using the model, we could observe the most important variables according to the decision tree. As you can see in Figure 10, it is no surprise that the most important variable to predict (any sales at all) was whether a store was open. Following that, the day of the week was the next most important predictor of sales values, followed by 'Month' and 'State Holiday' (or whether there was a state holiday on a particular day.

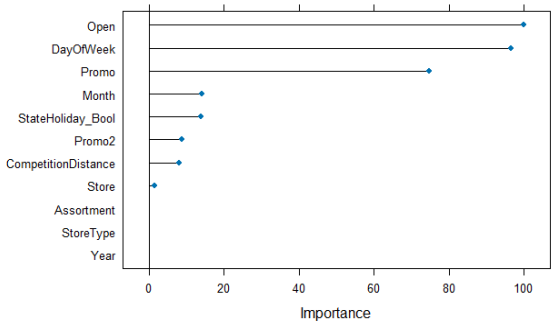

Figure 10: Importance of each variable to the model

Once the training had been completed, sales were predicted from the model using the validation dataset as previously mentioned, and the performance evaluated. As can be seen in Table 2, the R^2 was 0.56, which is another way of saying that the model can explain 56% of the variability observed in the model of the target variable ('Sales').

Table 2: Performance statistics from the Decision Tree model

RMSE	Rsquared	MAE
2525.3047307	0.5681591	1646.2677724

Once the calculations were done, the final RMSPE value of 55.31% was found. This value is higher than the 0.5, or 50% general rule-of-thumb for a good RMSPE which is between 0.2-0.4. The Route Means Square Percentage Error essentially measures the predictive power of a model by measuring the distance between the actual and predicted values in the data. Therefore, the lower the distance, the better the model is at predicting, in this instance; sales. As Figure 11 shows, the model is overall, predicting lower sales values than the actual observed sales.

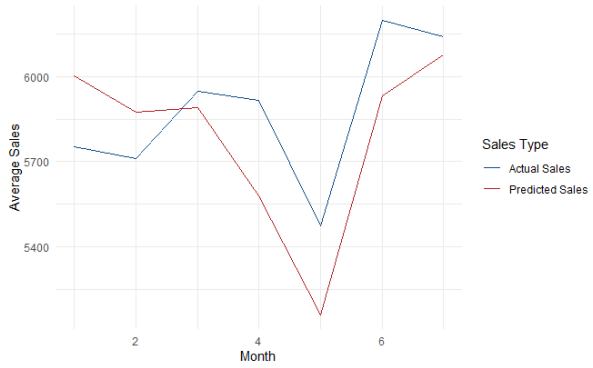

Figure 11: Average actual vs predicted sales per month using a Decision Tree model

2.2.2 Random Forest

We then tested the predictive power of another non-linear modelling technique called “Random Forest”. This is essentially a model that contains a set of decision trees. One of the main drawbacks over the decision tree model is that it requires considerably more processing power, but as Ali et al. (2012) point out, Decision Trees are very handy when using smaller datasets as the difference between the results would not be significant, whereas on larger datasets, the differences become greater, meaning that Random Forests have better predictive power in these instances.

Again, the model was trained on the same `train` data, and validated against the same `validation` datasets.

Once this had run, evaluation statistics were calculated and can be seen in table Table 3. This shows a marked improvement in the R^2 value, with 0.79, or 79% or the variation in predicted sales being able to be explained by the model. Effectiveness of the model was again evaluated with an RMSPE value of 48.41% being calculated. This proved to be a slight improvement overall from the Decision Tree model as it fell just below the 0.5, or 50% threshold.

Table 3: Performance statistics from the Random Forest model

RMSE	Rsquared	MAE
1782.0058417	0.7938676	1178.9595297

If we compare the Figure 11 above and Figure 12 below, we can visualise this improvement, with the overall gap between actual and predicted sales shrinking slightly, but with these figures significantly closer together between months 1 and 2.

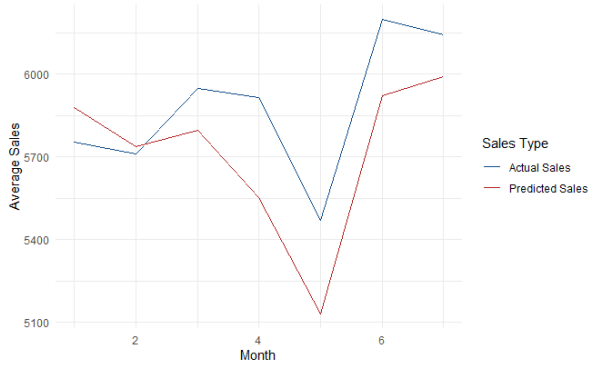

Figure 12: Average actual vs predicted sales per month using a Random Forest model

2.2.3 XGBoost

The final non-linear model to be chosen for testing on the ability to predict sales was the eXtreme Gradient Boosting (XGBoost) model. This is another non-linear tree based machine learning algorithm, similar to the Decision Tree and Random Forest previously discussed. The major difference(s) is that, for example, the Random Forest will calculate the results using a number of 'trees' that have been produced in parallel, whilst the XGBoost model uses a tree and sequentially tries to improve on that one.

Again, the model was trained and validated using the same datasets as before. The Root Mean Squared Error was used for the results calculation parameter, and the number of iterations was set to 1000.

Once the model had been trained and validated the R^2 value was calculated to 0.92 or, again, 92% of the variation in predicted sales can be explained by the model. In addition, again, the RMSPE was also calculated to 44.19% This is another improvement on the previous (Random Forest) model.

```
        RMSE     Rsquared          MAE
1101.292649     0.918781   747.022660
```

If we look at Figure 13 we can again see a visualisation of the improvements of the XGBoost model. The 1st 3 months were relatively similar, however there is a visual narrowing of the gap between the actual and predicted sales for the rest of the period being analysed.

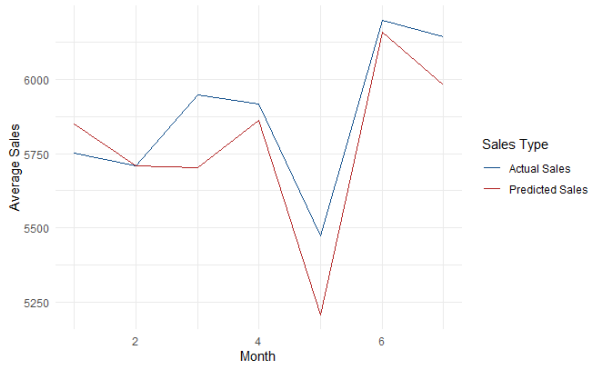

Figure 13: Actual v predicted sales from the xgboost trained model

2.2.4 USE XGBOOST MODEL TO PREDICT SALES

With the above in mind, we decided that the model that proved to be able to provide the best predictions of 'Sales' was the XGBoost model. Therefore we would use that in the final prediction of sales for a period where sales were unknown as opposed to the training and validation of predictions using known sales.

3 Plotting the final sales results

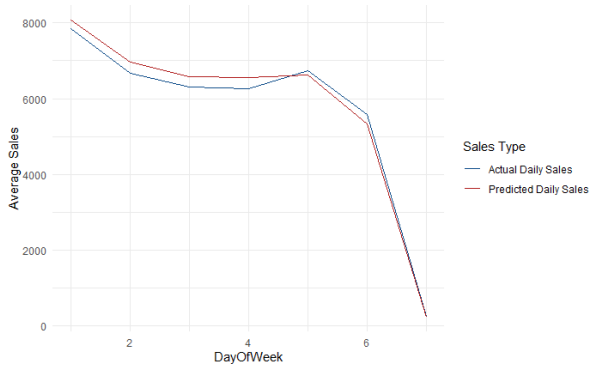

Figure 14: Average actual vs predicted sales per week of year using the XGBoost model

Review the available data and describe it in terms of its variables, quality, and relevance to the sales forecasting

Link data sets together as appropriate

Pre-process the data as appropriate for further analytics, for example, you may want to encode any categorical data, create new variables, identify how many missing values there are and deal with them appropriately, etc.

Identify the key factors affecting sales, for example, you may want to check whether competition and promotions have an impact on sales, and how public holidays cause sales fluctuations.

Build a forecasting model (which can be a linear regression model, a neural network model or something else) using the variables you identified. Please make sure to justify the choice of your modelling approach.

Use the Root Mean Square Percentage Error (RMSPE) to forecast accuracy

4 Results

Interpret key results, assumptions and limitations of your analysis.

5 Conclusion

5.1 Limitations

5.2 Implications

5.3 Recommendations

References

Akinwande, M.O., Dikko, H.G. and Samson, A. 2015. Variance inflation factor: As a condition for the inclusion of suppressor variable(s) in regression analysis. *Open Journal of Statistics*. **05**(0707), p.754.

Ali, J., Khan, R., Ahmad, N. and Maqsood, I. 2012. Random forests and decision trees. **9**(5).

Group, A.W. 2024. Rossmann. *AS Watson Group - A member of CK Hutchison Holdings*. [Online]. Available from: https://www.aswatson.com/our-brands/health-beauty/rossmann/.

Hasan, M.R. 2024. Addressing seasonality and trend detection in predictive sales forecasting: A machine learning perspective. *Journal of Business and Management Studies*. **6**(22), pp.100–109.