Sales forecasting for the European drug store Rossmann

Word count: 1963

Table of contents

Context

Predicting sales is a vital part for any business across all sectors, from manufacturing to retail. However, this is one of the most difficult tasks a business can undertake due to the complexities involved and the number of factors that can influence them, such as the store location, yearly seasonality, and the day of the week (Hasan, 2024).

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 investigate predictive modelling techniques to help the company anticipate sales across the business. This report will describe the importance of good quality data and the processes involved in using four predictive modelling techniques, from which one will be chosen to demonstrate the benefits of machine learning techniques in predicting sales.

Methodology

Data cleaning

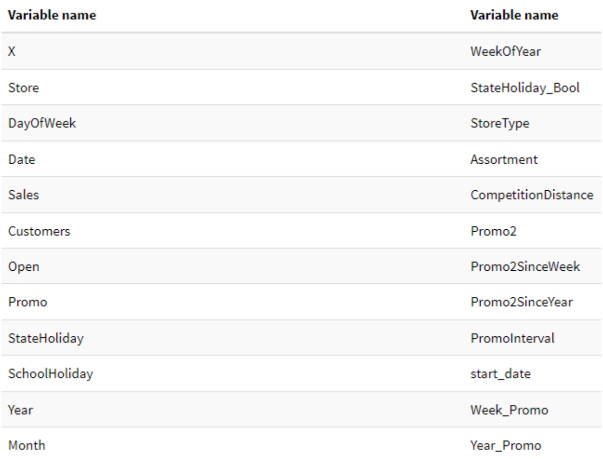
There were three datasets to be cleaned and used in the models: -

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

Data quality is vital because problems can and have caused problems with failed projects and customer turnover. Addressing poor data quality is vital in machine learning to minimise the severe problems that can arise. For example, missing data can be treated in a number of ways, which will have different effects on how well models perform. If a large enough proportion of the data is missing and is deleted, the statistical power of the model will be dramatically reduced in many cases, which is why choosing the correct option for each is important. (Gudivada et al., 2017).

Store data

See Figure 1 for variable names. Categorical variables were converted to numeric categories . A decision was made to imupte many of the missing values using the mean of all other distances. Variables that contained too many missing values were however, removed completely as they would not be used in the model, while others were used to update a binary variable to show if promotion was running at the date of the observation.

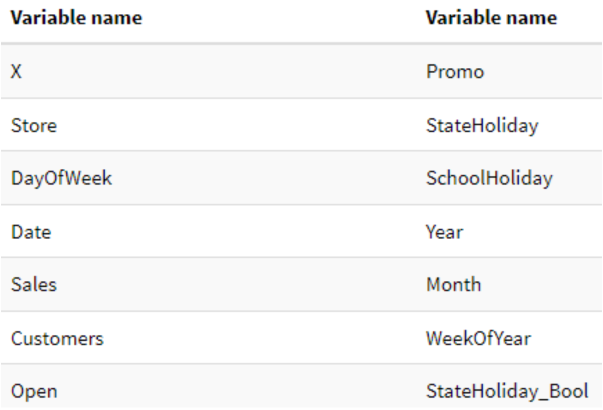


Variable name	Variable name
X	WeekOfYear
Store	StateHoliday_Bool
DayOfWeek	StoreType
Date	Assortment
Sales	CompetitionDistance
Customers	Promo2
Open	Promo2SinceWeek
Promo	Promo2SinceYear
StateHoliday	PromoInterval
SchoolHoliday	start_date
Year	Week_Promo
Month	Year_Promo

Figure 1: Variables included in the 'Store' dataset

Train data

See Figure 2 for the training dataset. This would ultimately be used to train the model with a 70/30 split (train/validate), however, all of the data involved would need to be cleaned in the same manner. For example, the 'date' was split into three that consisted of day, month and year because each was anticipated to have a level of impact on the sales individually. If a store was not open, they were dropped. A boolean was created to show either (state) holiday or not-(state)holiday.

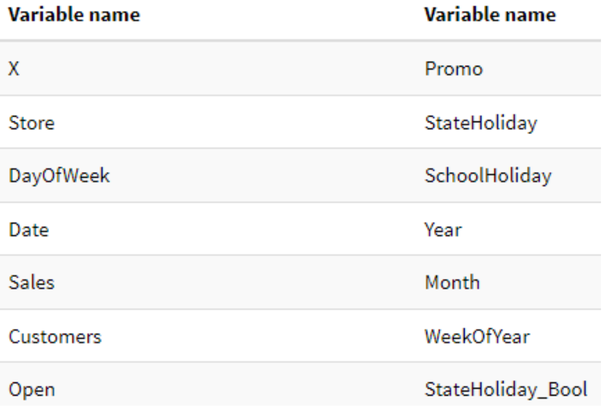


Variable name	Variable name
X	Promo
Store	StateHoliday
DayOfWeek	SchoolHoliday
Date	Year
Sales	Month
Customers	WeekOfYear
Open	StateHoliday_Bool

Figure 2: Variables included in the 'train' dataset

Test data

See Figure 3 for the test data. This data contained no sales information, and would be used to test the final model once the best one had been chosen, tested, and validated. The 'date' variable was again split into the three components. The 'open' variable contained some missing data which it was decided to infer 'open == True' for these, due to the related 'promo == T' variable.



Variable name	Variable name
X	Promo
Store	StateHoliday
DayOfWeek	SchoolHoliday
Date	Year
Sales	Month
Customers	WeekOfYear
Open	StateHoliday_Bool

Figure 3: Variables included in the ‘test’ dataset

The ‘store’ and ‘train’ datasets were finally joined using the ‘store_id’ field.

Exploratory data analysis

Distribution plots were then produced. Firstly, sales figures were plotted (Figure 4). €0 sales values were often on a Sunday when a store was closed, so were not removed as they were related to the ‘dayofweek’ variable.

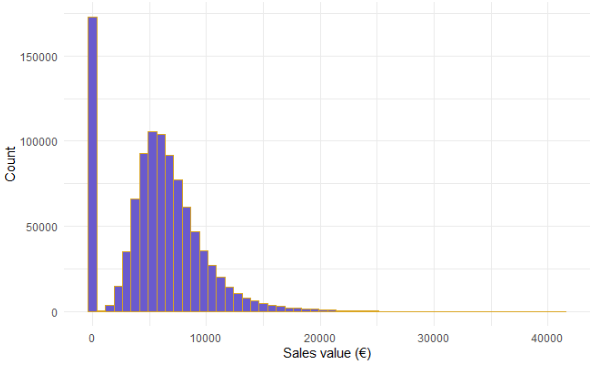


Figure 4: Distribution of historic sales

Figure 5 shows the distribution of customers. There were a large number of 0 values due to a store not being open on a Sunday, and the number of stores that did not open on a Sunday were large.

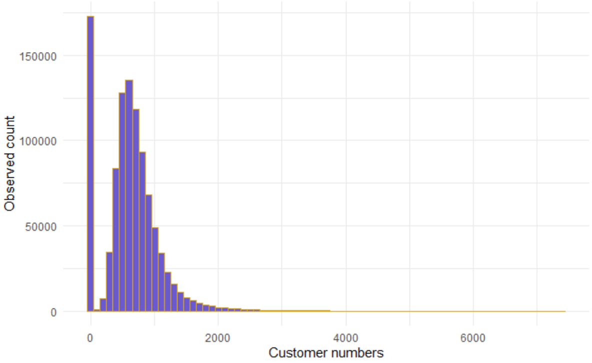


Figure 5: Distribution of historic customer counts

Figure 6 shows that the majority of stores were located relatively close to competition. Considering the nature of the business this shows that the stores are located in areas with a large number of shops around them but very few are in isolated locations by comparison to competitors.

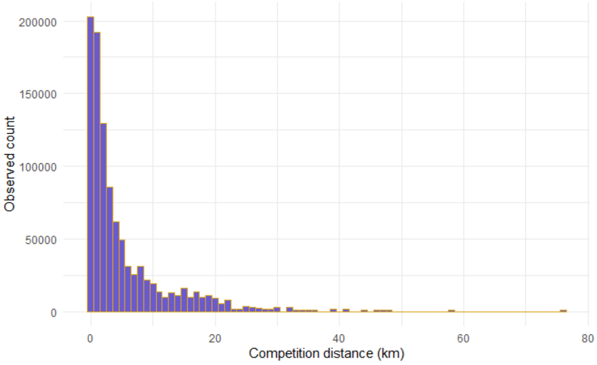


Figure 6: Competition distances counts

A correlation plot was then produced. This can be seen in Figure 7.

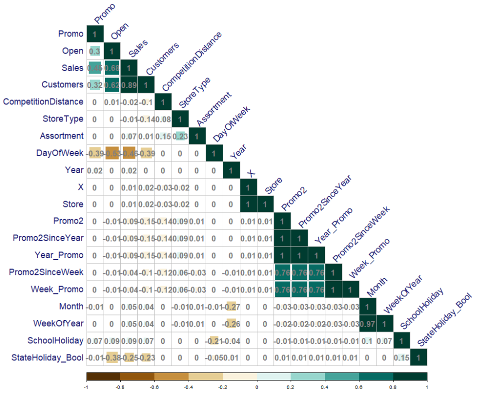


Figure 7: Correlation plot show collinearity

There was a relatively high correlation between the Customers/Sales variables, and the Promo variables. Those above 0.75 would be removed due to multicollinearity.

Another correlation plot was produced just to check the validity of our removal decisions. Figure 8 shows that all (except WeekOfYear/Month) correlations are now under 0.75.

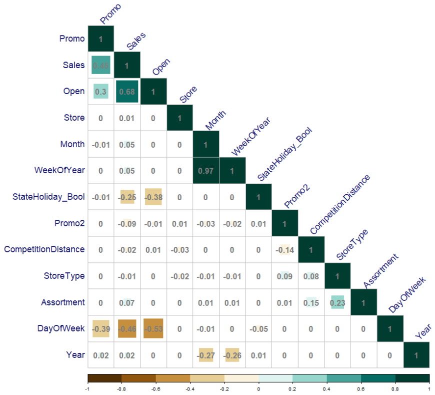


Figure 8: Correlation plot after multicollinearity was addressed

Figure 9 shows the degree of Variance Inflation Factor (VIF) and that all the remaining variables' GVIF values were under 5.

	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

Figure 9: Variance Inflation Factors for the linear model

A heat map shows the level of linearity in the relationships between the variables. and Figure 10 shows the majority have a low linearity.

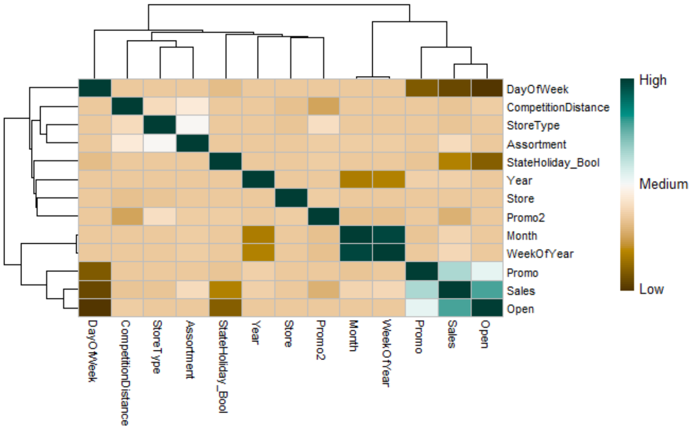


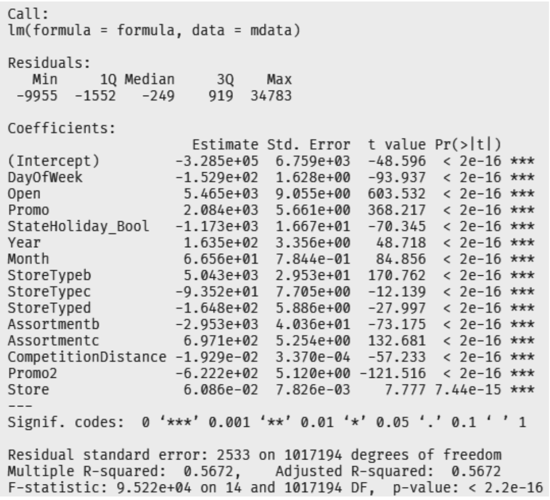
Figure 10: Heat map showing level of linear relationship

Testing the models

Once the remaining variables would not skew the results of any regression or other predictive models, we could investigate which of the chosen models was able to best predict sales.

Model 1: Linear regression

A linear regression model was used and Figure 11 shows the results. Looking at the p-values shows all variables were significant. The R^2 was around 0.56, sp 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 11: Summary of linear regression model

Figure 12 shows the distribution of the residuals model representing the difference between the actual and predicted values. This distribution show that there is a fairly large variation between the actual and predicted values.

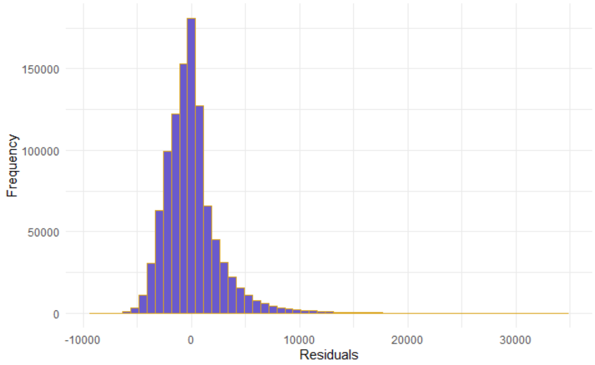


Figure 12: Plot of residuals for the linear model

The QQ-Plot (Figure 13) shows how well the residuals produced by the model match what we would expect from normally distributed data. Up to a certain point (~1.25) they match expectations, but after that, the quantiles are greater than expected. This suggests that there is a relatively large amount of data that is non-linear.

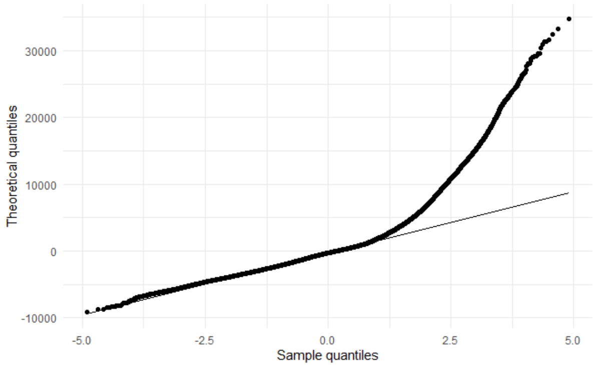


Figure 13: Q-Q Plot of Residuals

The data was then split into train and validation sets with train 70% and validation 30% of the original data.

Model 2: Decision Tree

Decision Tree models are one of the oldest computational methods, and are highly interpretable. They are non-linear, hierarchical models that use a series of decisions to produce various results, culminating in consequences based upon likelihood of, for example, the chance of an event outcome (Ville, 2013).

Firstly a `tune_grid` was created from all combinations of variables. This is part of the hyperparameter optimisation which allows for the optimal number of branches in the decision tree to be found. It was then trained using `rpart` which is a regression classification method.

The most important variables according to the decision tree can be seen in Figure 14 which shows open the day of the week, followed by 'Month' and 'State Holiday' were the most important in predicting sales.

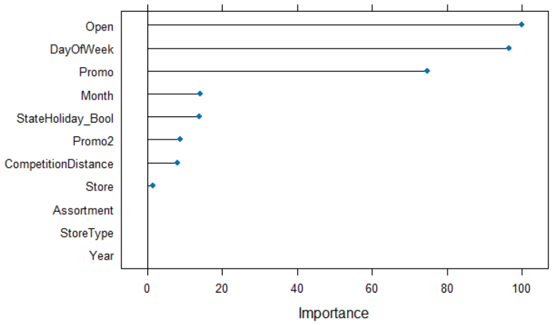
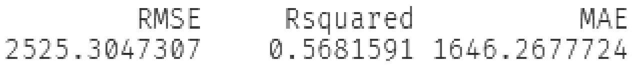


Figure 14: Importance of each variable to the model

Sales were then predicted using the validation dataset. Figure 15 shows through the R^2 that the model can explain 56% of the variability observed in sales.



```
        RMSE     Rsquared          MAE
2525.3047307    0.5681591 1646.2677724
```

Figure 15: Performance statistics from the Decision Tree model

The Root Mean Squared Percentage Error (RMSPE) value of 55.31% shows that 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 RMSPE measures the predictive power by measuring the distance between the actual and predicted values. The lower, the better the model. Figure 16 shows this model is predicting lower sales values than the actual observed sales.

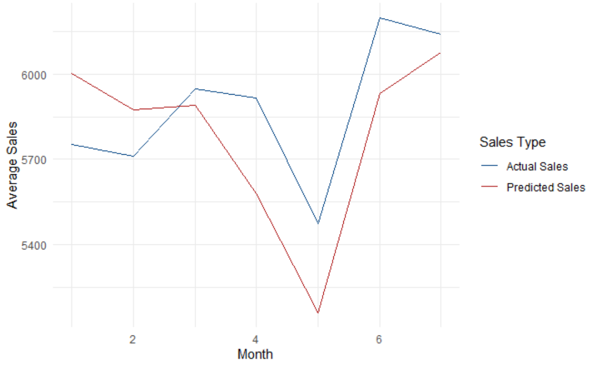
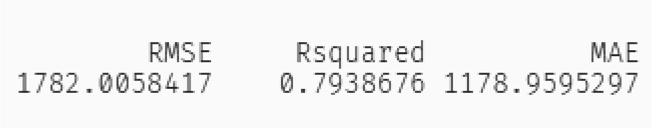


Figure 16: Average actual vs predicted sales per month using a Decision Tree model

Model 3: Random Forest

This model 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 such as this, the differences become greater, meaning that Random Forests have better predictive power in these instances. Again, the model was trained and validated.

Figure 17 shows a marked improvement in the R^2 value, with 0.79. Effectiveness of the model shows an RMSPE value of 48.41%. This proved to be a slight improvement from the Decision Tree model as it fell just below the 0.5, or 50% threshold.



```
         RMSE      Rsquared          MAE
1782.0058417     0.7938676 1178.9595297
```

Figure 17: Performance statistics from the Random Forest model

Comparing Figure 16 above and Figure 18 below, shows the gap between actual and predicted sales shrinking slightly, and significantly closer together between months 1 and 2.

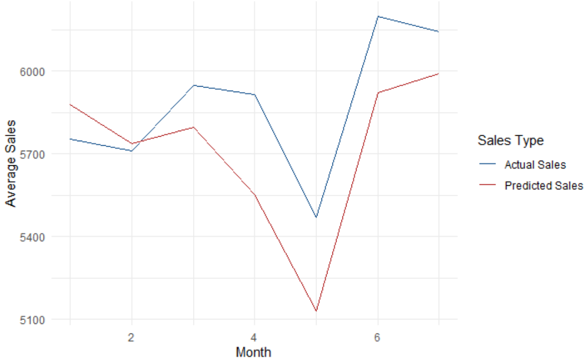
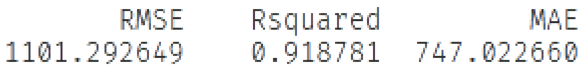


Figure 18: Average actual vs predicted sales per month using a Random Forest model

Model 4: XGBoost

The eXtreme Gradient Boosting (XGBoost) model is another non-linear tree based machine learning algorithm, similar to the Decision Tree and Random Forest. The major difference is that 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 that is sequentially trying to improve on itself (Jhaveri et al., 2019).

Figure 19 shows the resulting R^2 value was 0.92. In addition, 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
```

Figure 19: Performance statistics for the XGBoost model

Figure 20 shows that we can see the improvements of the XGBoost model. The 1st three months were relatively similar, however there is a narrowing of the gap between the actual and predicted sales for the rest of the period.

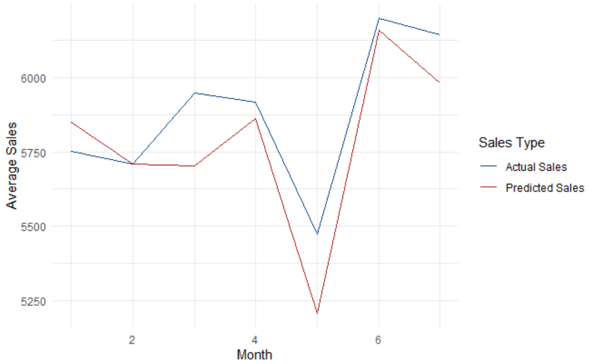


Figure 20: Actual v predicted sales from the xgboost trained model

Using XGBoost to predict sales

With the above in mind, we decided that the model that provided the best predictions of 'Sales' was the XGBoost model. Therefore we would use that in the final prediction for a period where sales were unknown.

Final XGBoost results

Figure 21 shows that for each day of the week (1-7 == Monday to Sunday), the model is very accurate in predicting sales. This is important because some stores will be open or closed on different days to each other and will allow Rossmann to better decide which stores are open on which days.

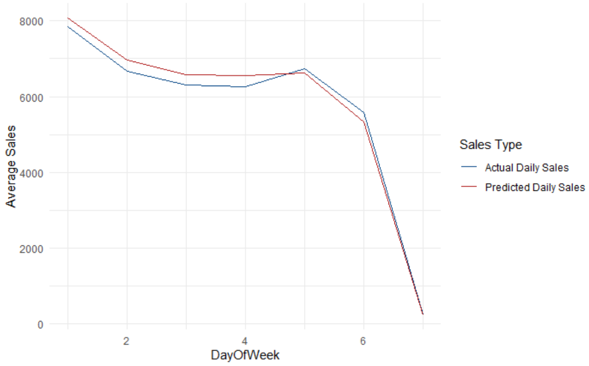


Figure 21: Average actual vs predicted sales per week of year using the XGBoost model

Further evidence of the accuracy of the model can be seen in Figure 22. Each individual store on the x axis representented by a peak, displays a value and similar pattern between the predicted average sale value that is relatively close to the actual sales value.

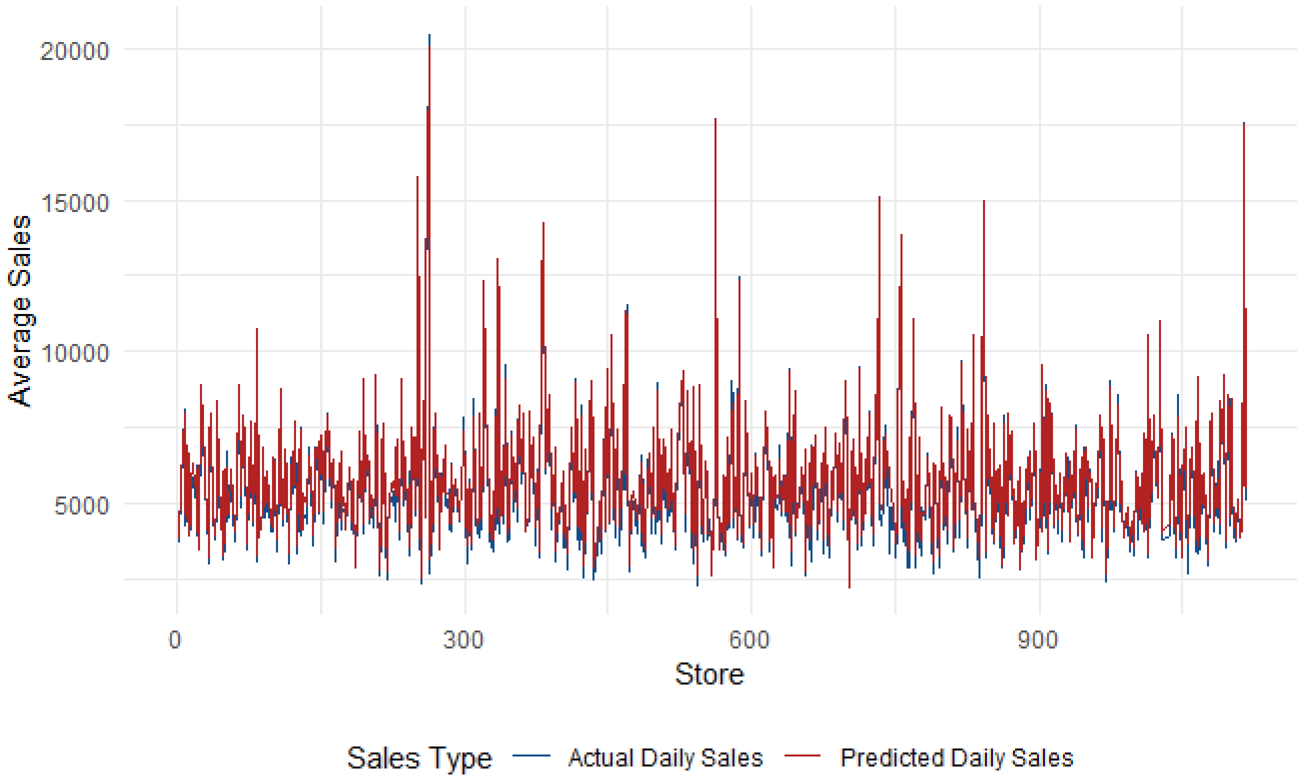


Figure 22: Accuracy of predicted sales per store

Limitations of the study

The main limitation of the XGBoost algorithm is that, it requires greater processing power (unnecessary multi-threaded optimisation) than the other models used in this report and is more time-consuming, which means it takes it longer to run to conclusion (Ma et al., 2021). This would be something for a business to consider, when thinking about how to optimise sales, but that decision would have to be based on the trade-off between time and cost.

One limitation of this study overall is the lack of processing power available. If more were able to be used, it would have been pertinent to the study to compare the results of another model called Prophet, which is especially good at handling time-series data such as this, and is especially useful for predicting sales with low error rates and displays better fitting (Kumar Jha and Pande, 2021).

Benefits of machine learning

The cost of utilising such a system of sales prediction however would be minimal because of the nature of using something like R, which is free and 'open-source' as used in this report. One other advantage of the XGBoost model is that it is very good at handling missing data; in fact it was designed to handle missing data well (Saraswat, n.d.). This could provide very useful to business when considering whether or not to open a new store, considering that sales data would be missing from a new store, making this method very useful in this context.

Implications & recommendations

As can be seen in Figure 21 and Figure 22, the model highlights the importance of machine learning and it's ability to be utilised by business to predict things like sales. XGBoost is an especially good example of this and we would advise that further research is carried out, but ultimately the business should adopt either XGBoost or another similar model to predict future sales.

References

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/.

Gudivada, V.N., Apon, A. and Ding, J. 2017. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations.

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.

Jhaveri, S., Khedkar, I., Kantharia, Y. and Jaswal, S. 2019. Success prediction using random forest, CatBoost, XGBoost and AdaBoost for kickstarter campaigns *In*: *2019 3rd international conference on computing methodologies and communication (ICCMC)* [Online]., pp.1170–1173. Available from: https://ieeexplore.ieee.org/abstract/document/8819828?casa_token=0nwz8hrCsZ8AAAAA:09DDbNgR-xtTrU81Y3NckjNftXINlhkC7mNEWQfqZK0NObPrSyn92DXaFRuvpjSpBM-R-m4.

Kumar Jha, B. and Pande, S. 2021. Time series forecasting model for supermarket sales using FB-prophet *In*: *2021 5th international conference on computing methodologies and communication (ICCMC)* [Online]., pp.547–554. Available from: https://ieeexplore.ieee.org/abstract/document/9418033?casa_token=NzJwD4VSiZUAAAAA:NqdRkyHR3ufieLAt2_tBeZp9BzcjoQTYt4VM3mQPdSi-7T9CzgguQCbjB_BXBmTB5ICN2Kk.

Ma, M., Zhao, G., He, B., Li, Q., Dong, H., Wang, S. and Wang, Z. 2021. XGBoost-based method for flash flood risk assessment. *Journal of Hydrology*. **598**, p.126382.

Saraswat, M. n.d. Beginners tutorial on XGBoost and parameter tuning in r tutorials & notes | machine learning. *HackerEarth*. [Online]. Available from: https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/beginners-tutorial-on-xgboost-parameter-tuning-r/tutorial/.

Ville, B. de 2013. Decision trees. *WIREs Computational Statistics*. **5**(6), pp.448–455.

Appendix

This link provides access to the complete project files hosted on Github. Here you will find the code used in the processing of the models. This also contains a .ipynb file that was used to to do the majority of the cleaning of the data as a group.