Applied Techniques of Data Mining and Machine Learning Coursework

Business context

Modern businesses collect and store a huge amount of data and making use of this data can be a key differentiator between successful companies and their competitors. In a supply-chain limited business environment it is more important than ever to be able to forecast customer demand effectively as global lead times ever increase. Demand planning is crucial to being able to ensure available inventory to meet customer requirements, especially for just in time (JIT) replenishment. For sales, the availability of inventory can be the difference between making and losing a sale. Not only is demand forecasting crucial for customer sales it helps high-performing organisations look after cashflow by having a higher stock turn. Further benefits include the reduction of waste for expiring material and more efficient use of storage space, reducing the need for expensive warehousing space. Forecasting demand accurately can also allow the efficient allocation of resources to the correct areas of the business and provide insights into changing sales trends. Indeed, the correct understanding of customer demand can set a company apart from the competition and provide insight to investors potentially increasing shareholder value.

This piece of work attempted to forecast the sales of a US superstore using sales data for the previous 4 years. Statistical trends can be understood during data preparation. Once the data has been analysed and cleansed, feature extraction and was performed on the sales data. The features were used to forecast overall sales for the superstore for known sales data. Subsequent rounds of modelling analysed performance of different categories of products, and finally sub-categories within. The three rounds of modelling were compared for accuracy of forecast at different levels. Once model accuracy for known data was determined, a forecast into the future sales was performed. The eXtreme Gradient Boosting (XGBoost) model was utilised due to it being quick, scalable (Dairu and Shilong, 2021) and having an easy to use python package available (xgboost developers, 2021). In addition, the XGBoost model is more easily understood than the “black box” Artificial Neural Networks (ANN) which is important in any model being presented to leadership and used for business cognition.

Forecasting is not a new problem, being the subject of numerous works which have been published with varying degrees of success. (Choi, Hui and Yu, 2011) explore several models for the fashion industry and the difficulties in forecasting such a volatile sector. Statistical (SARIMA), AI (ANN, ELM, HSNN) and hybrid models (EELM) were explored in their work with no single solution providing the desired result for that sector. (Xu, Tang and Rangan, 2017) explore B2B sales looking at customer relationship management data to understand the pipeline of the more complex enterprise sale environment. Their solution used a library of models including neural networks, probability models, time series models and opportunity score aggregation models in a forecast engine creating a robust but complex forecasting tool achieving high accuracy within 100ms. Some novel models have been suggested such as the Fuzzy Time Series (FTS) model by (Burney and Ali, 2019). Whilst the results show promise with much reduced root mean squared error (RMSE) compared to previous models, the dataset utilised by Burney and Ali is unclear. The forecast results are strong, but it is unknown how much data is used for training and there is no commentary on the tendency to overfit the data.

One common source of data for forecasting is Kaggle, and one dataset has been the focus of numerous approaches to forecast as part of a Kaggle competition: A Walmart sales dataset covering 1913 days and over 30,000 SKU’s. (Zhou *et al.*, 2021) used Light GBM model in conjunction with a grid search algorithm to refine to parameters of the model and compared this to linear regression and SVM models. The RMSE of the model was lower than the more traditional models, though no analysis was offered on the comparative speed of the models and cost for the relatively small decrease in RMSE. (Li *et al.*, 2020) Applied a neural network with three LSTM layers and a dense layer. The

backpropagation of LSTM is more complex than a normal neural network and the predictions we performed every 15th day using a grid search method to find the optimal hyperparameters. Exceptional RMSE results were achieved but the complexity of this model means businesses cannot rationalise the predictions.

(Chen *et al.*, 2021) Utilised a Deep Neural Network (DNN) model and achieved RMSE scores around half of the comparative linear regression and ridge models. Their study also included a SHAP analysis into the most important features for prediction and found rolling features, item and demand most influential on the model. Their attempts to uncover the black box nature of the DNN algorithm were interesting, but with two hidden layers there was still some mystery leading to difficult comprehension of predictions. Again, no comparison of computational effort and time is presented relative to the decrease in RMSE. A decrease from 5.78 in a SVM model to 3.20 in a DNN may not be a huge benefit given the increase in complexity. As (Choi, Hui and Yu, 2011) opined, accuracy is not the most important factor when analysing a model. Models should be analysed based on their impact to business performance. While intuitively a higher RMSE should mean worse performance, it is possible that there is only negligible impact on business performance. Risk to the business is a better measure to use, and the scalability and flexibility of a simpler model may be more useful as a result.

(Dairu and Shilong, 2021) Modelled the Walmart dataset using an XGBoost model. Due to the sensitivity to outliers, their dataset was first groomed for outliers using standard deviation and filtering to clean the data. Memory compression was achieved converting to float data type, increasing speed of prediction. Once the data was cleansed, time features and some statistical features were engineered for modelling; specifically rolling feature, lag feature and min, max and median features. Feature selection was applied to remove redundant features to aid with overfitting keep performance speed. Comparison to other models' performance for prediction of 28 days sales were determined using the RMSE accuracy measure. XGBoost had an accuracy of 0.655 compared to 0.783 for Linear Regression and 0.774 for Ridge Regression. These results evidence the performance of this algorithm is strong despite avoiding the complexity of a neural network. Ease of deployment and flexibility make it an ideal choice for forecasting time series data. No comparison is made with respect to performance speed between the models, and no analysis is offered on the significance of the RMSE value difference compared to the magnitude of the mean value of sales. The XGBoost method was chosen for this piece of work and the stages of forecasting will mirror the work of Dairu and Shilong. They concluded that XGBoost may be able to handle more hierarchical feature representations. The aim of this work is to take that step and compare the accuracy of models comprising of differing levels of product hierarchy; complete aggregation; categorisation; sub-categorisation.

Data exploration and understanding

The data being analysed is a retail dataset (Kaggle, 2022) .csv file of superstore sales over 4 years. The size of the data is 9800, with dimensionality 18. The data population is very strong with only one data point having 11 null values: Postal Code.

- Row ID

Row ID is a sequential number of interval domain. It is completely populated for the entire dataset and is purely a sequential unique identifier for each record.

- Order ID

Order ID is another sequential identifier, this time of Ordinal domain. There is an order within the values given the grouping prefix. This attribute groups instances together within each sales order. The largest order has 14 rows, with the mean order containing 2 rows and the median 1 row.

- Order Date

Order date is of ordinal domain which indicates date of sale. The busiest day had 38 sales, with the average day containing 7 sales as a mean and 6 as a median.

- Ship Date

Ship date is another ordinal domain attribute containing dates. It is the day the order was shipped and can be used to engineer the dwell between order date and shipment date. It also can be used to visualise when deliveries are scheduled. Not all days have sales equally, and a model will need to account for that.

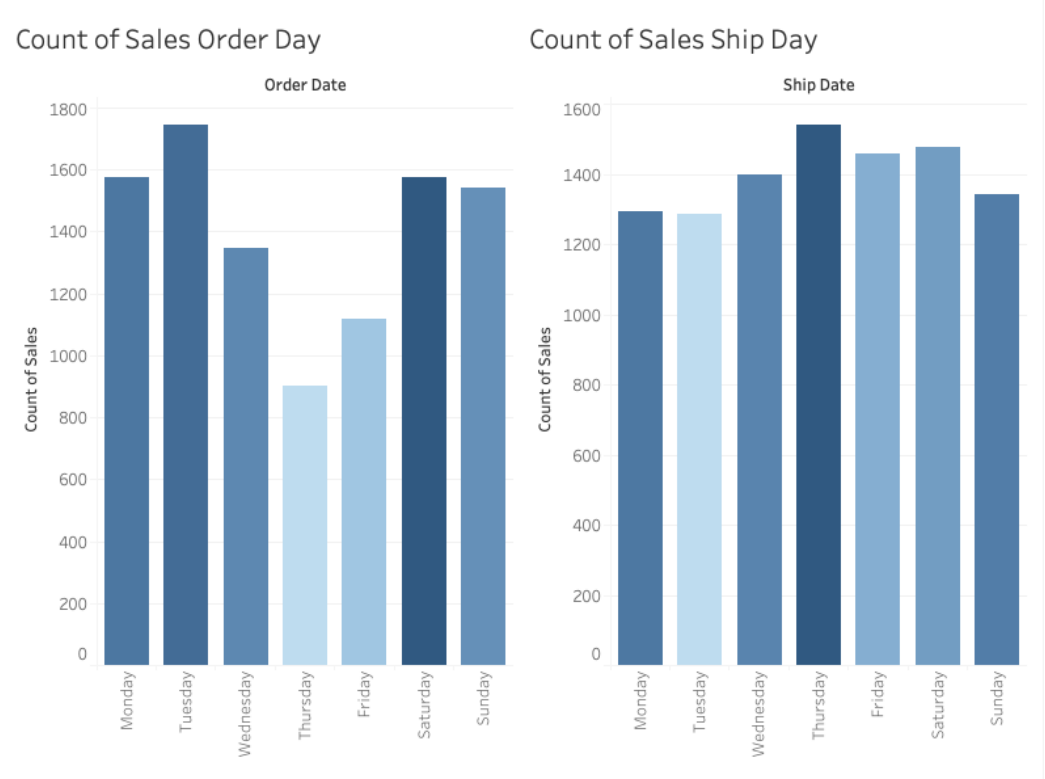

Figure 1 - Comparison of order and shipment volume on different weekdays

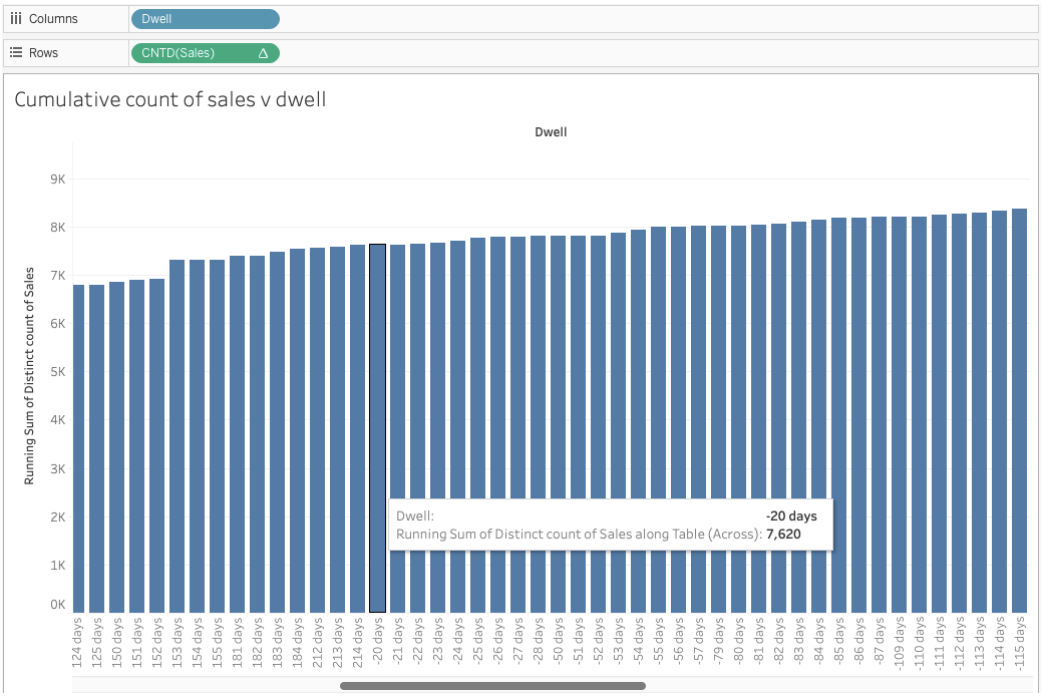

Figure 2 - Cumulative count of sales across dwell between order ate and ship date

Figure 2 demonstrates that the dwell between shipment date and order date runs into the negative, which is an implausible value. Over 2500 of the 9800 records have a negative dwell which would indicate that either shipment date or ordering date is inaccurate. In this piece of work, order date was selected to compare product sales as within that date there was evidence of seasonality that would be expected in the sales industry.

- Ship Mode

Ship Mode is a categorical data attribute. It is the method by which delivery was performed with four categorical values; Same day; First class; Second class; Standard Class. It can be considered ordinal in domain as the priority decreases with each category.

- Customer ID

Customer ID is a unique identifier of nominal domain. It is unique per distinct customer and can be used to group purchases to individuals and build profiles of customer buying. The most prolific buyer has 35 rows, while the median customer has 11 purchase instances across all their sales.

- Customer Name

Customer name is a nominal domain attribute identifying customer by name. For purposes of building a model, this personally identifiable information should be excluded.

- Segment

Segment is a categorical attribute of nominal domain. There is no order or hierarchy in the three distinct values of “Consumer”, “Corporate” and “Home Office”. The segment groups customers by their business type. There is no ability to examine quality of this attribute without intimate knowledge of the products and or customers themselves.

- Country

Country is the location of the customer and is another nominal data attribute. In this dataset there is only one value for country: “United States”. It does not provide any information and as such will not be discussed any further in this report.

- City

City is a nominal data attribute relating to the address of the customer and thus destination of the sale. There are 529 distinct cities recorded across the United States which can give an indication of the geographic sales trends. All cities were successfully mapped onto a MapBox plot confirming data validity, if not accuracy of true customer location.

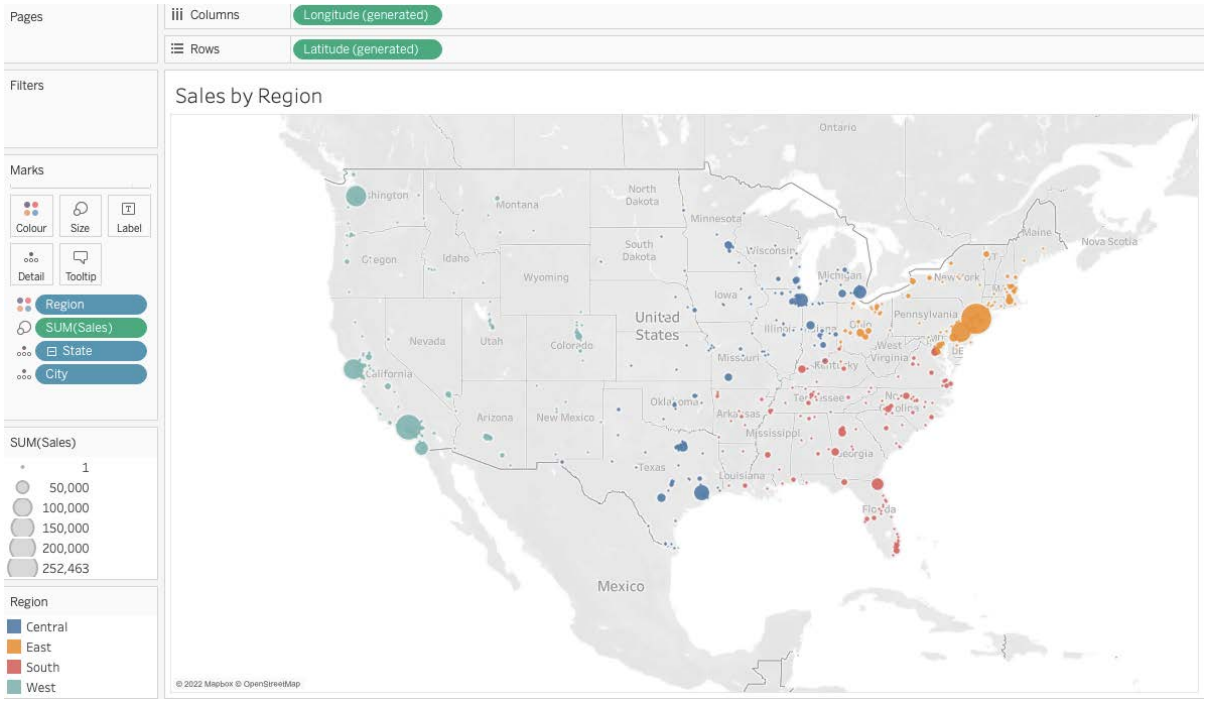

Figure 3 - Sales by city with regional colouring and sales volume bubble

Figure 3 provides an insight into the sales spread in the business. The sales along the coastal areas were higher and in West are almost exclusively coastal. The major cities dominate as expected, but the business had fair infiltration across most of America except for the central Western region from North Dakota to Idaho and Oregon. East and West regions had the highest sales, whilst there is good coverage in Central and Southern regions.

- State

State can be used in conjunction with city to understand sales location trends. State would allow aggregation at a higher level if desired. There are sales recorded in 49 of the 50 states within America with the highest number of sales being achieved in California, New York, and Texas.

- Postal Code

Postal code is another location-based attribute of nominal domain. It is the only attribute with missing data, with 11 missing values out of 9800.

- Region

Region is a nominal label assigned to geographic regions with the United States. There are four possible values: Central, South, East, West. This allows another level of aggregation for location analysis. Oftentimes business areas are split into regional management and given the size of the

United States and the volumes of business in each area it was likely the case with this superstore. Sales are dominated by the East and West regions.

Figure 4 - Sales by region

- Product ID

Product is a nominal domain unique identifier for each product sold. Product will be a huge driver in identifying the value of a sale as ultimately, that is what is being purchased. There are 1861 products in the dataset across 9800 rows, which indicates a diverse set of products. The most popular product ID was sold 19 times. This attribute's data quality will be discussed more in conjunction with Product Name

- Category

Category is a categorical nominal attribute with three values: Furniture, Office Supplies, and Technology. This is a higher grouping of products and is more applicable to model given the population of it. There is no way to validate the accuracy of this data without intimate knowledge of the domain.

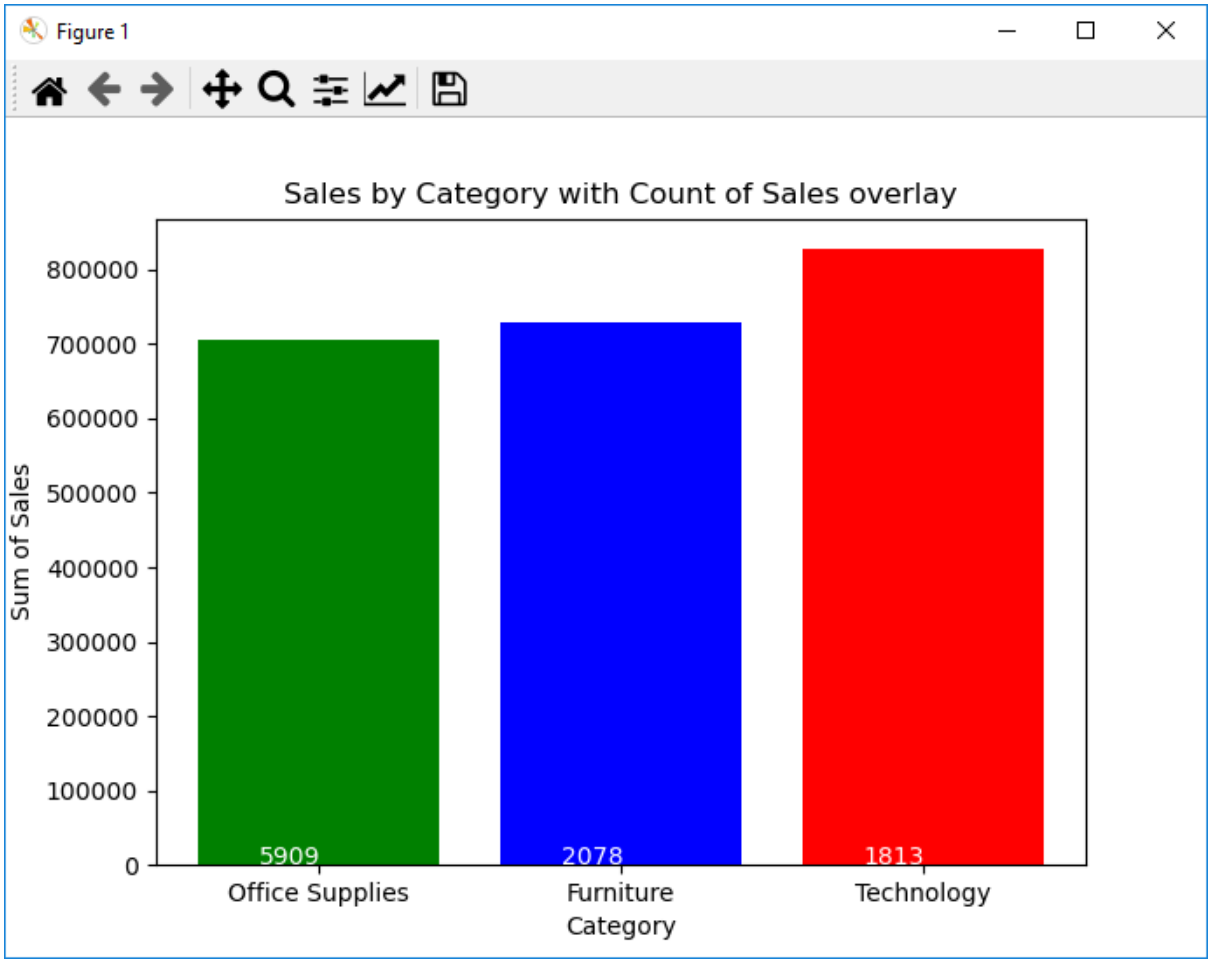

Figure 5 - Sum and count of sales by category

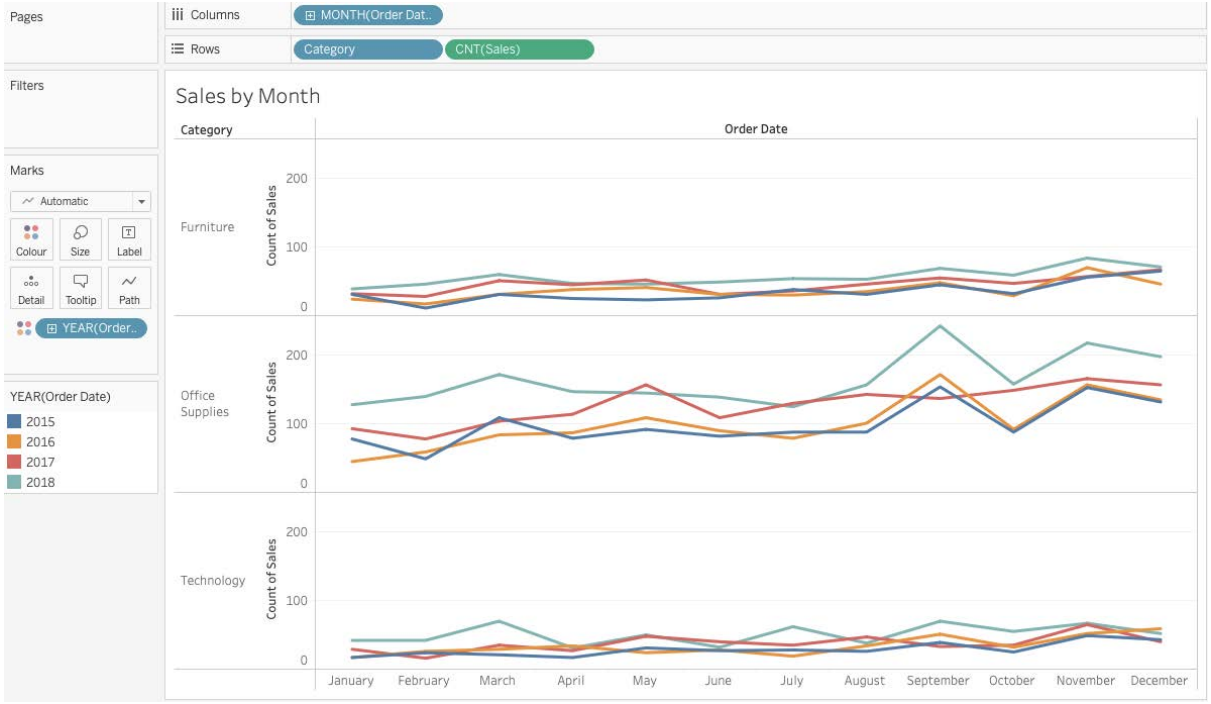

Figure 6 - Seasonality of sales by category

Figure 5 shows that the technology sales were the biggest revenue generator, whilst office supplies might have provided better cash-flow due to the much higher volume of sales. Figure 6 shows the monthly sales volumes for each category per year. Sales grew year on year, with clear seasonality displayed.

- Sub-Category

Sub-Category is another categorical nominal attribute a level of aggregation lower than category. It aims to branch products into more specific groupings within the product category. It may be more useful than product ID to understand product trends where lots of products fit into one sub-category such as “Phones”. The distinct categories and subcategories are shown the below figure with their sales performance across the 4-year period.

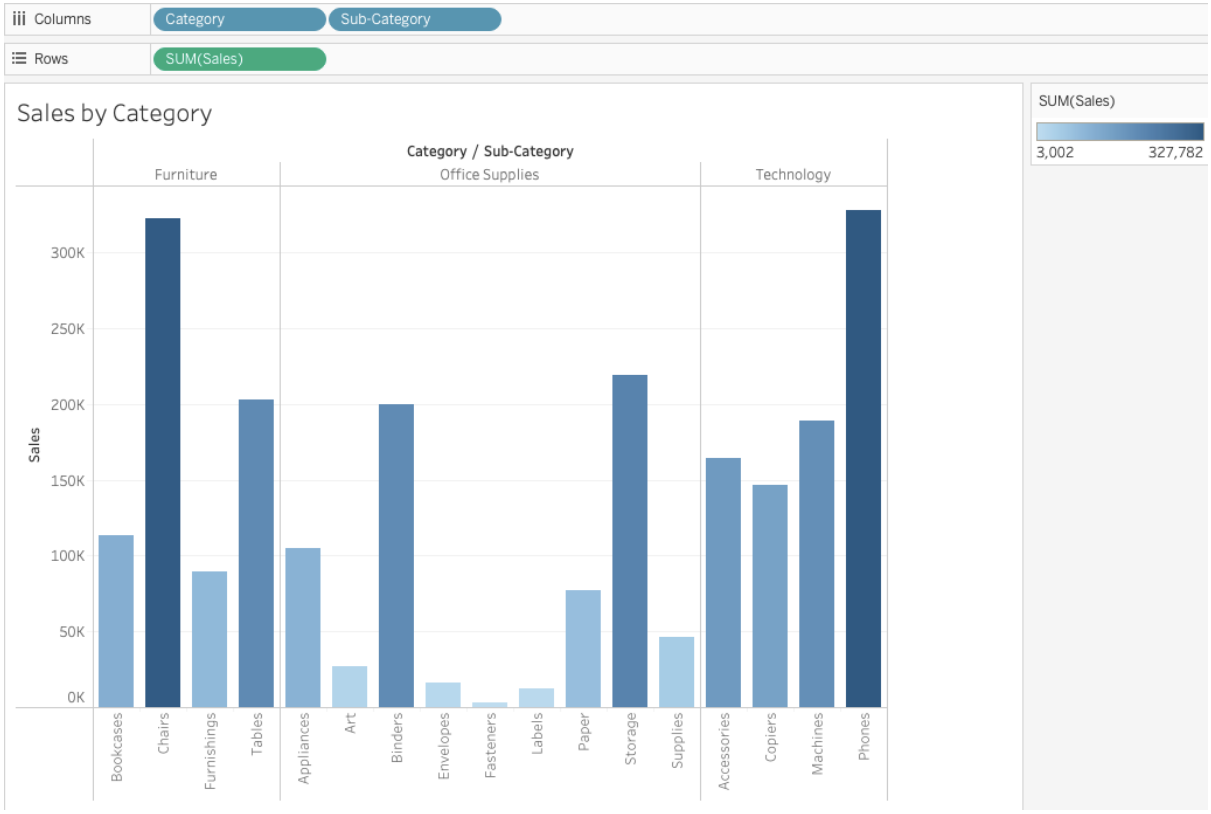

Figure 7 - Sales by category and sub-category

- Product Name

Product Name is a text representation of product ID describing each individual item, of nominal domain. There are 1861 product IDs, but only 1849 product names. This is concerning from a quality perspective. “staples” and “envelopes” were purchased a lot more than 19 times, as much as 47, indicating some product identification issues whether it is an undescriptive name, or duplicate ID. For example, “Staple envelope” has 9 distinct product IDs. Envelopes may have distinct properties not disclosed in the product name or the dataset has poor part mastering with numerous IDs being created where one already existed. Sub-category is the lowest reliable attribute.

- Sales

Sales is the target variable in this piece of work. It is a quantitative measure of ratio value. Whilst 0 is not present in the dataset, it is possible to have a 0-value sale, despite it not being practical for a business. As demonstrated earlier in this section there are numerous relationships to sales within each attribute.

The data quality of this dataset is hard to validate without expert knowledge. Other than postal code, it is completely populated across all data instances. As discussed in data exploration, the order and ship dates contain either calculation or collection error. In the case of Ship Date, it would more likely be collection error: no system would calculate a ship date in advance of an order. Product ID and Name share data quality issues. Region, Category and Sub-category are completely populated and there is no reason to discredit the quality of the data without any domain knowledge otherwise. The map of sales shows by region indicate that attribute is accurate.

Data preparation and pre-processing

Forecasting is a time series modelling issue, so one of the date fields must be retained and selected as the date to model against. Order date is that feature for this piece of work. Analysis was required into the level of aggregation that is desired. Given the business aim of the project is forecasting future trends, customer specific information or unique identifiers were not required for analysis. The sales volumes of individual products were not sufficient to project forward. Some level of grouping was needed. Row ID, Order ID, Ship Date, Product ID, Product Name, Customer Name, Segment, City, State, Region and Ship Mode were all excluded from the model. This made the model less complex and removed some data integrity concerns around the products themselves and the location data.

After feature selection, outlier analysis was performed before the undesired instances and features were removed from the dataset.

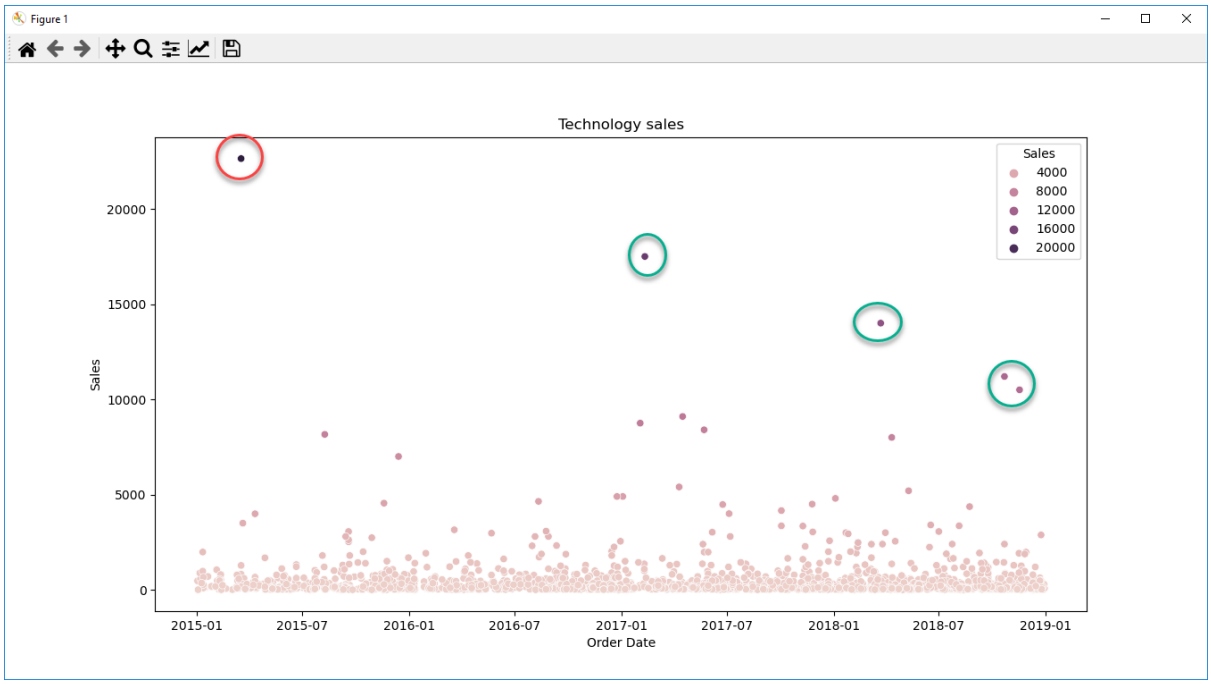

Figure 8 - Outlier analysis in Technology

There were 5 records much larger sales than the rest of the population and were considered for outlier analysis. The highest sale of all was for one product in 2015 for 22,638.5 which had no other sales in the entire dataset. As such it was removed from the dataset as an outlier. For the other 4 values, they were all for the same product that had been sold each year, at decreasing value as seen in figure 9. This is reasonable as a lot of technology becomes cheaper over time as newer models come out. The rest of the sales fell within a reasonable range.

After feature removal, new features were engineered. For the first two rounds of modelling, sales were grouped by day allowing for more datetime features to be extracted. Year, month, day of year, day of the month, day of the week and week of the year were all extracted, along with lag features. Per (Lazzeri, 2021), lag features look at the target feature at prior timestamps with the ideology that past data can predict the future. Three lag features were selected, looking at the past three years respectively to capture the seasonality of the sales trends. The first round grouped sales by date alone, and the second by sales and category.

Feature extraction returned a groomed dataset of 10 features, 11 when category was included. XGBoost allows for the use of categorical features with the experimental parameter "enable_categorical = True" selected, reducing the need for encoding to integer values for the categorical features. Instead, the datatype of these features was converted from object to category.

One of the business objectives was to forecast into the future, and to achieve this, an empty DataFrame was created with datetime features at daily intervals. Time and lag feature extractions were performed on this dataset to generate test dataset for modelling.

Evaluation of the first two rounds of modelling indicated an issue with the daily sales aggregation. As such a third round of modelling was performed with aggregation into week and year in "YY-WW" format. In addition, product sub-category was explored to see if the trends seen from the categorical model can be pinpointed into specific product groups. The data preparation for the third round had reduced features due to the loss of day of week, day of month and day of year features.

Data modelling and evaluation

Once pre-processing was complete, the dataset was split into training and test sets. Initially the dataset was split into 3.5 years training and 0.5 years training, which is an 7/8:1/8 train test split.

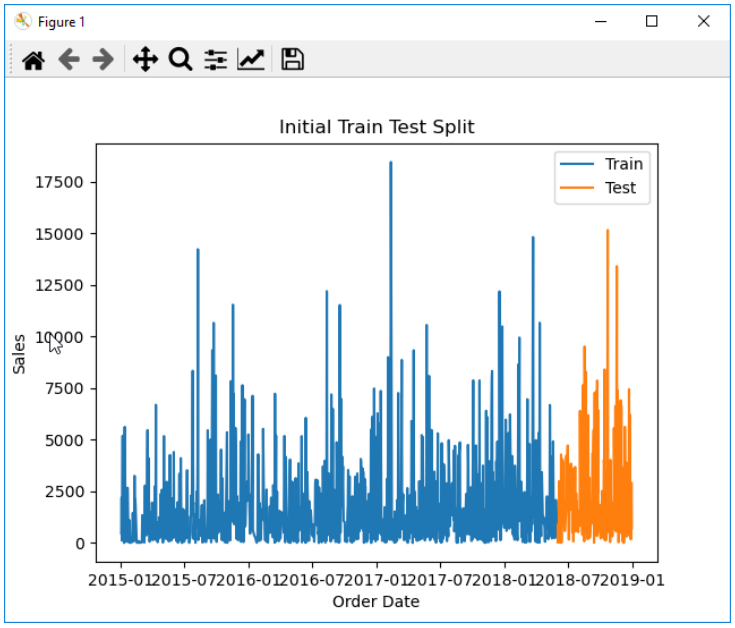

Figure 9 - Initial Train Test Split

The model selected the XGBoost regressor from the XGBoost library within the python API (xgboost developers, 2021). Parameters used included an initial learning rate of 0.3 and objective for reduced RMSE. The default base score was used with a gradient boosted tree model. The Tree booster was limited to a maximum depth of 6 nodes for initial analysis in order to prevent overly complex decision trees and save performance.

```
reg = xgb.XGBRegressor(base_score = 0.5,
                       booster = 'gbtree',
                       verbosity = 1,
                       learning_rate = 0.3,  ##learning rate imppacts how quickly we get to the endpoint of learning.
                       max_depth = 6,
                       objective = 'reg:squarederror',
                       ##n_estimators = ,
                       ##early_stopping_rounds = ,
                       seed = 1)
```

Figure 80 - Initial regression parameters

The first round of training ran for 100 trees at a learning rate of 0.3. The RMSE score compared to the training set reduced all the way down to 364.11, whereas the minimum RMSE for the test validation was not achieved. Predictions map well to the test data, but the validation set RMSE score indicated overfitting. Additionally, the model predicted negative sales figures which are impossible, confirming model tuning was required.

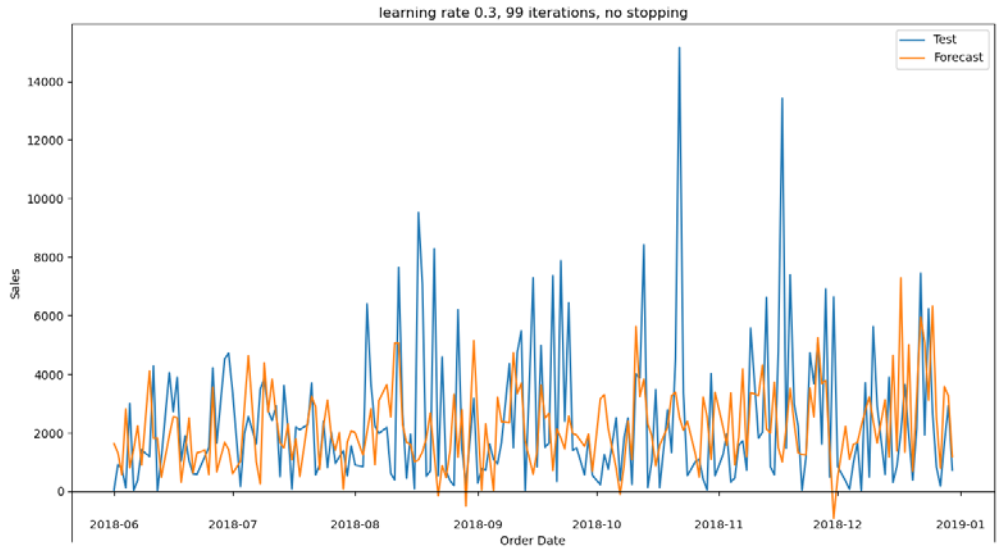

Figure 9 – Forecast vs known values for initial model training

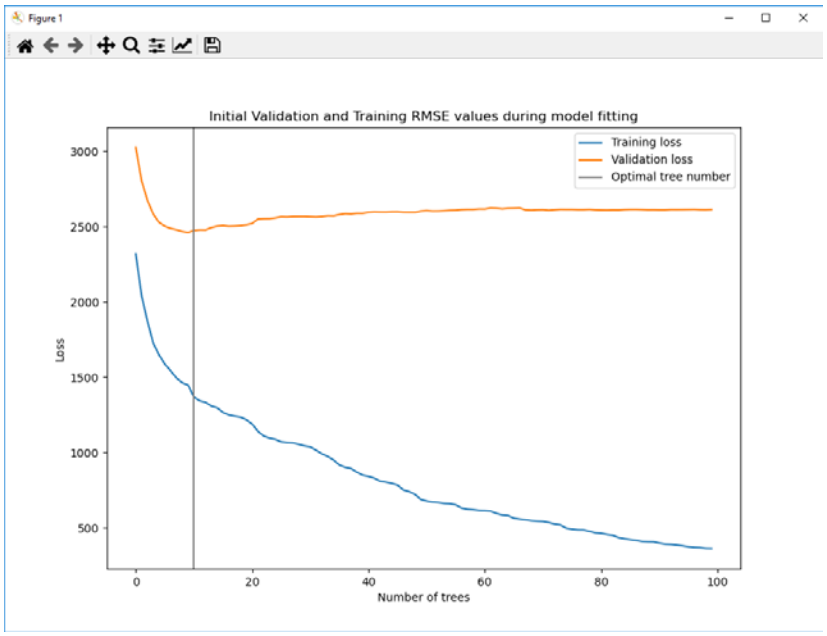

Figure 10 - Training and Validation Loss with vertical line at optimum position

The most optimal RMSE scores for the validation training set were around the tenth tree which suggests the learning rate was too high. Additionally, the early stopping measures parameter was not used to prohibit overfitting as seen in figure 12. Tuning the model, the number of trees were limited to 1000 with a learning rate of 0.01. As the learning rate was 30 times smaller, 1000 trees were more than enough to surpass the optimum value and overfitting occurred again.

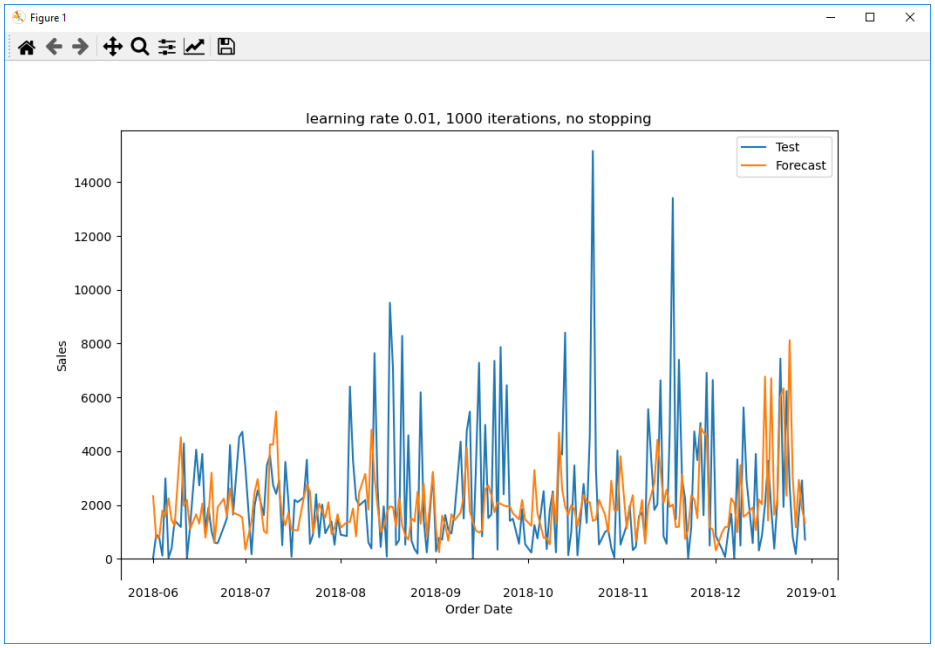

Figure 113 – Forecast versus known data for the lower learning rate

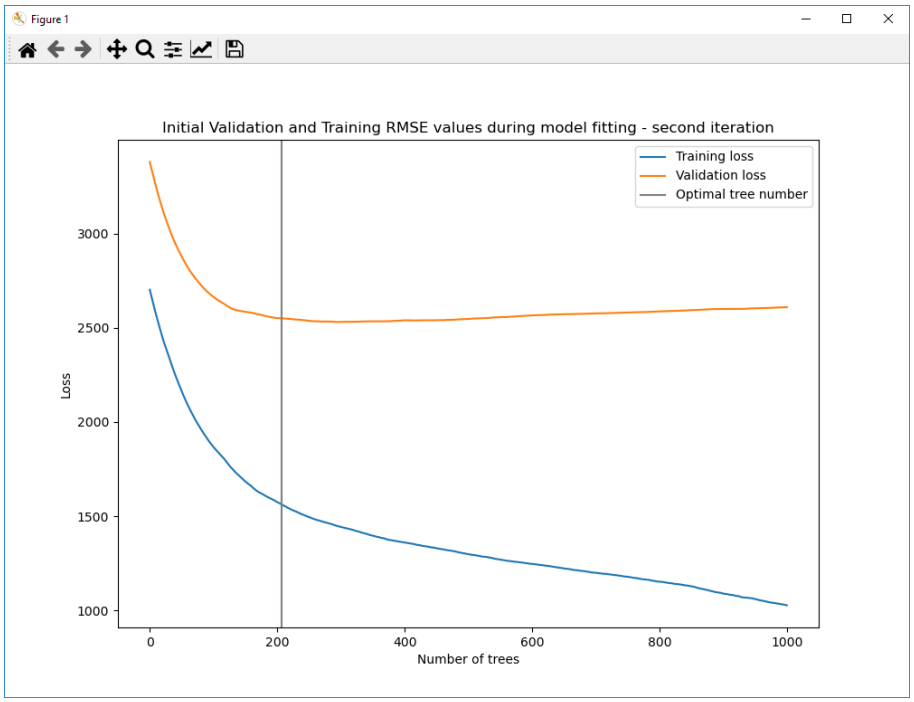

Figure 12 - Loss values with a learning rate of 0.01 for 1000 estimators.

The model still visibly tracked well to the testing data in figure 14, with the removal of negative values. The smoother loss curve in figure 16 validated that the decreased learning rate had improved the ability to reach the minimum RMSE for the validation set. More work was needed to avoid overfitting. Introduction of the parameter “early_stopping_rounds” stopped the model if the validation set RMSE did not improve after 7 new tree additions. Instead of 1000 trees, the early stopping setting ended the model at 295 decision trees.

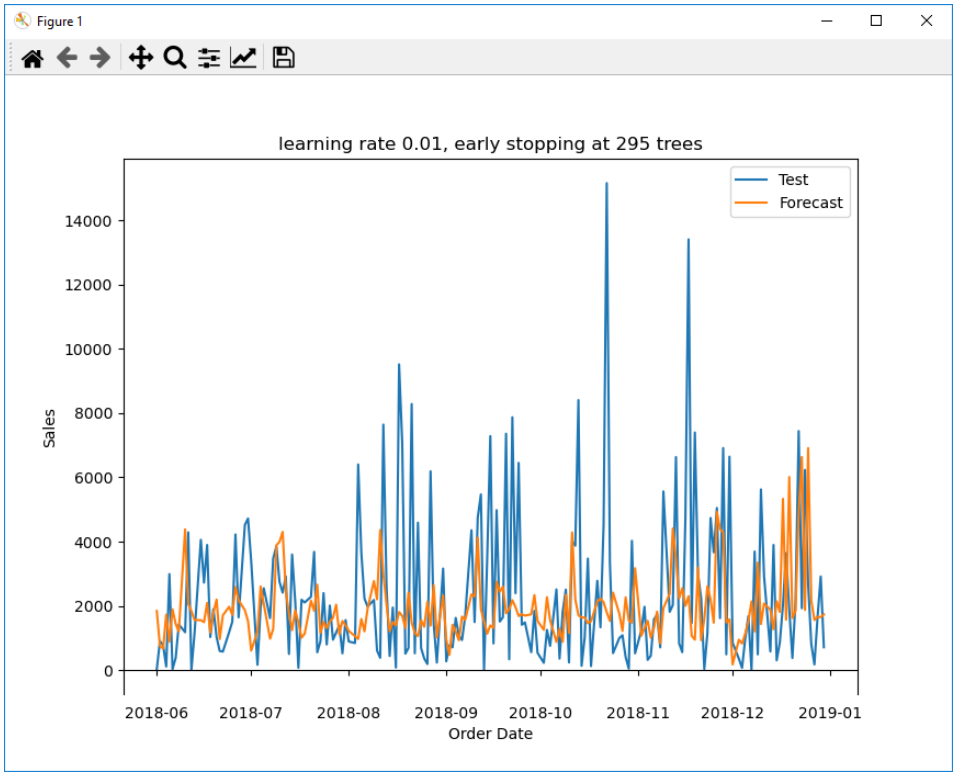

Figure 13 - Model fitting on optimal number of trees

The optimised model still visibly fit reasonably well with the threat of overfitting mitigated. The RMSE score for this model was 2531 and the mean sales value is 1820. This indicates that there was significant room for improvement. With the number of trees optimised, the model tuning focused on the training and test split. 2 years of possible split lengths were investigated with each of the 104 splits different by a week in length.

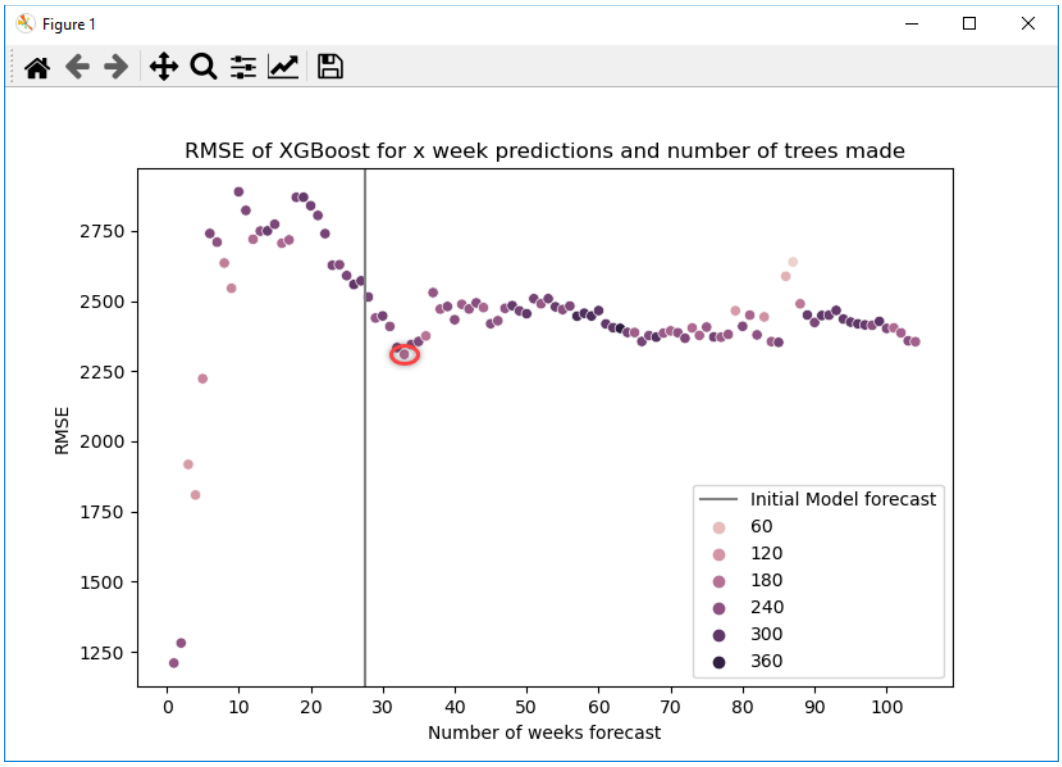

Figure 16 - Training set size accuracy and model complexity with number of trees made. Initial parameters highlighted

Figure 20 compares the forecast size, which is the inverse of the training set size. It concludes that the model very accurately predicts the next week's data however as predictions are made on larger datasets the predictions become less accurate very quickly before levelling out. The hue indicates the number of trees created, which can be considered a measure of model complexity. Short-term predictions were more accurate and less complex. The business task desired longer-term forecasts requiring a more complex model. Around 33 weeks forecast was a good balance between forecasting a long time in the future and retaining a level of accuracy and lower complexity. If model accuracy is the most important result, then predictions should only be weeklong. One interesting note is the least complex model was around 90 weeks forecast which would be interesting to investigate in future work. The optimal forecast length is visualised below.

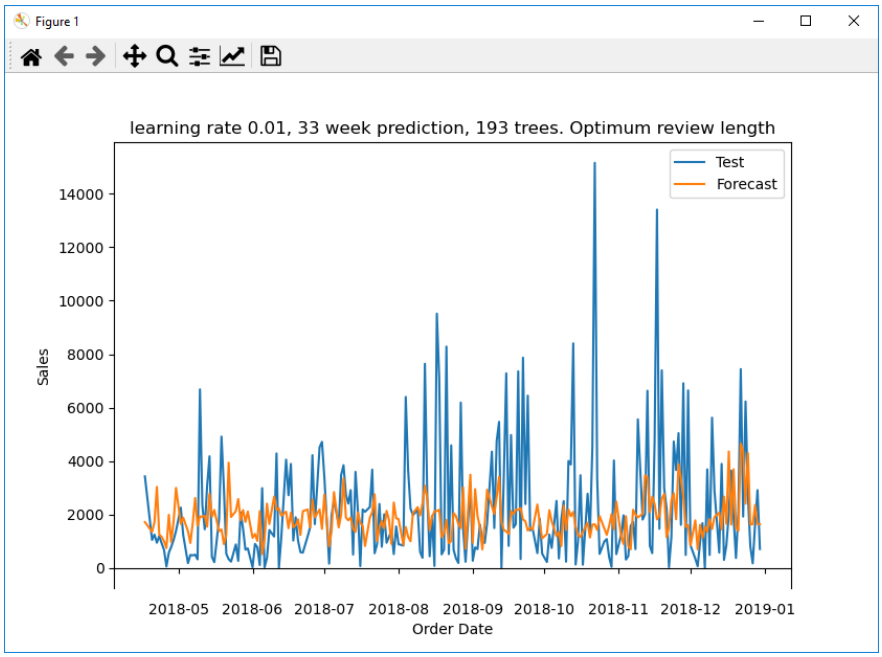

Figure 17 - Optimized train test split model

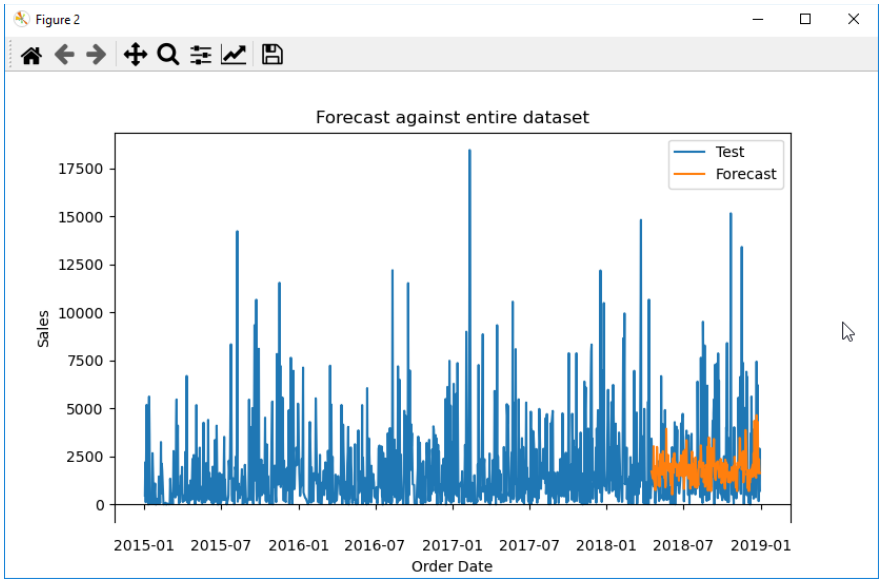

Figure 18 - Optimized model visualised against entire sales data

One feature of XGBoost is the feature importances. For the first round of modelling the most important features were day of month and day of Week.

```
In [87]: print(reg.feature_names_in_, "importance:", reg.feature_importances_)
['Year' 'Month' 'Day of Year' 'Day of Month' 'Day of Week' 'Week of Year'
 'Lag1' 'Lag2' 'Lag3'] importance: [0.08830722 0.0581697  0.1492433  0.14340152
0.09064447 0.15281366
 0.1096519  0.09822569 0.1095426 ]
```

Figure 19 - Feature Importance

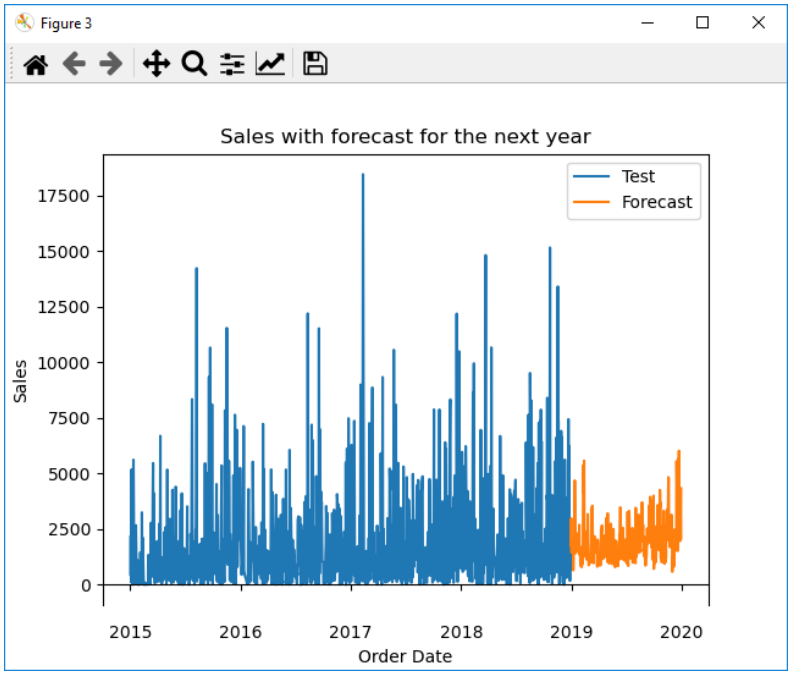

Figure 140 - One year future forecast

The optimised model was used to forecast future sales for one year, with the entire dataset used as training data. When forecasting the future sales, the model cannot rely on early stopping rounds as there is no test set to validate against. Instead, the rounds of stopping must be imposed on the model. 150 rounds or trees were selected as the endpoint as it was the point at which cross-validation on the training datasets started overfitting. The model parameters tuned for this analysis were kept constant for the other rounds of modelling to give fair results.

The future forecast highlights one problem with aggregation at a day level: the model never predicts 0 sales, which is not accurate. There were some days without sales meaning the prediction on a monthly or weekly may be more accurate than daily. This changed the business question to look at aggregation for weekly data which is seen in the third round of modelling. The second round of modelling looks to improve the accuracy by segregation of product category and to forecast into the future to understand how the business may change over time. The final accuracy for all models were graded by cross-validation.

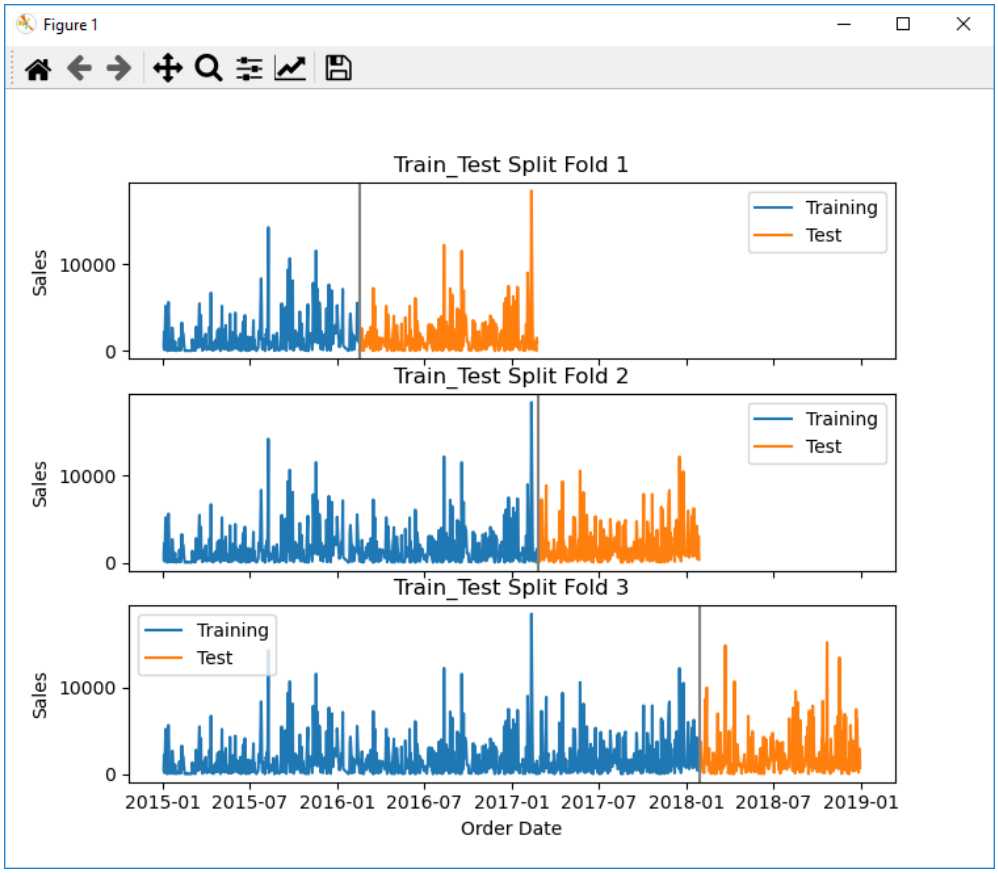

Figure 21 - Visualization of the cross validation

The categorical model when evaluated through cross-validation had a significantly lower RMSE of 1363, compared to 2332. However, the mean sale for this dataset was 788 which indicates the forecast is less accurate than expected despite a lower error. Reviewing the feature importance of this model, the category was the most important attribute, which was expected.

Figure 15 – Initial dataset and forecast per category per week

The categorical aggregation forecast has the same flaws as the first model in that no days are forecast to have 0 sales which is an issue with the date aggregation. Technology continues to have spikes in March each year and is forecast to continually grow.

Figure 16 - Product sales per year with prediction from 2018 added

Comparing the prior and forecast sales per year provided the most clarity and was the most useful outcome of the model. The forecast trends indicate a continued uptake of technology sales, with potentially more the 300,000 sales by 2021. Conversely, office supplies were predicted to be less profitable as sales drop initially before a slight recovery. Even by the end of 2021, office supply sales did not recover to the level they were at the end of the training dataset. Furniture was predicted to have continued growth but not at the same rate as technology with stagnation and decline in sales in 2021. This model, despite the questions over accuracy could be used to drive business strategy. The model concludes that business should focus their attention on technology sales and potentially divest resources from their office sales groups to support that focus. Intrinsically that shift makes sense with what is known about technological advances and the rise of working from home. Virtual notebooks and remote calls are removing the requirement for certain office supplies with email replacing mail.

Observing the results, the business question shifted to investigate if specific product lines within in each category were predicted to be successful or fail. This might dictate specific strategy or marketing campaigns to combat or accompany future growth and decline. The same model was applied to a third dataset to attempted to answer this question.

The features of the third round of modelling, grouped by product sub-category, were reduced as discussed in the data pre-processing section. In this round, week was aggregated to address the issues of 0 sale days never being forecast. The RMSE of this model was reduced compared to the categorical model, with a similar mean sale which indicates a slight improvement in performance. Given the model error is still almost twice the mean sale a lot of improvement can be made. RMSE of training data on the future forecast was lowest in the sub-category mapping task which indicates the model was able to mirror the training set more closely within the same number of decision trees. This is due to the reduced number of features.

Figure 174 – Sub-category prediction fitting

The sub-category level of prediction was problematic due to the sparsity of the data. A higher level of aggregation than sub categorical analysis is required.

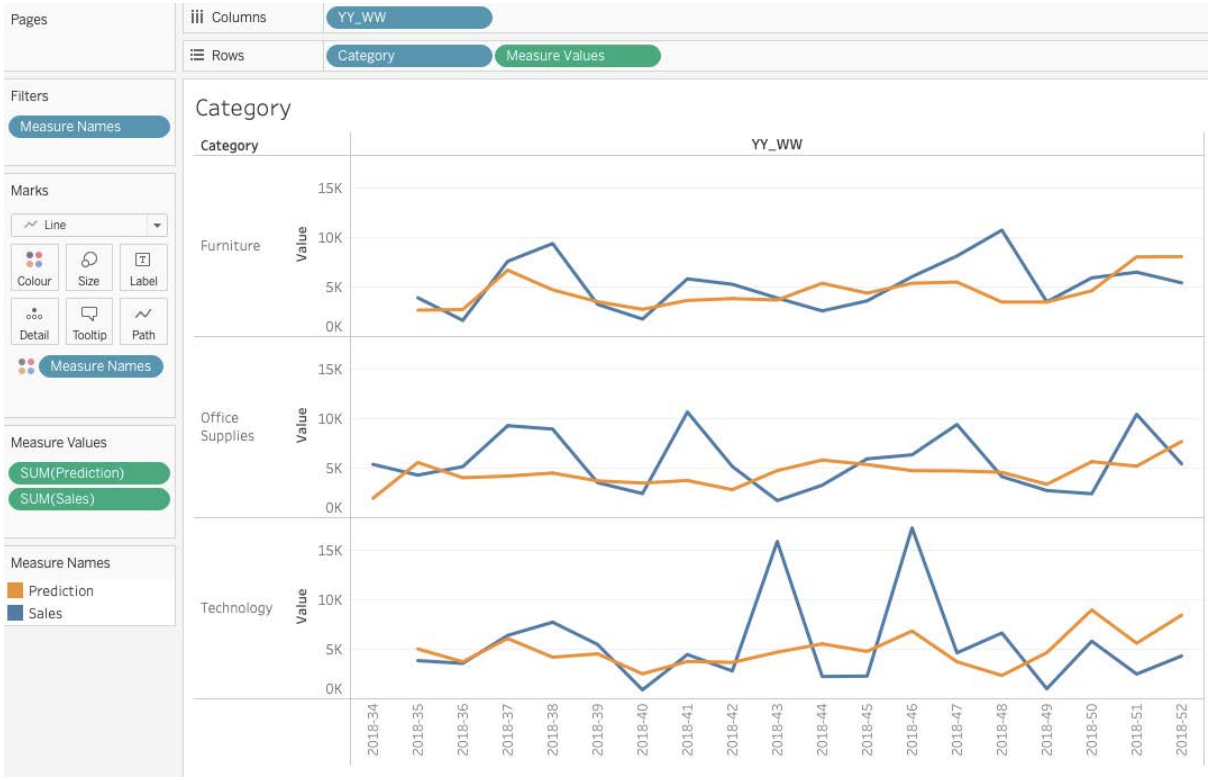

Figure 2518 - Subcategory grouped dataset fitting vs test data at category level

Within the sub-category dataset, it was possible to view how the model mapped category at a weekly level. It is a much more visually satisfying aggregation level, and the predictions more closely map to the test data. More importantly, the issue of 0 sales is reduced significantly.

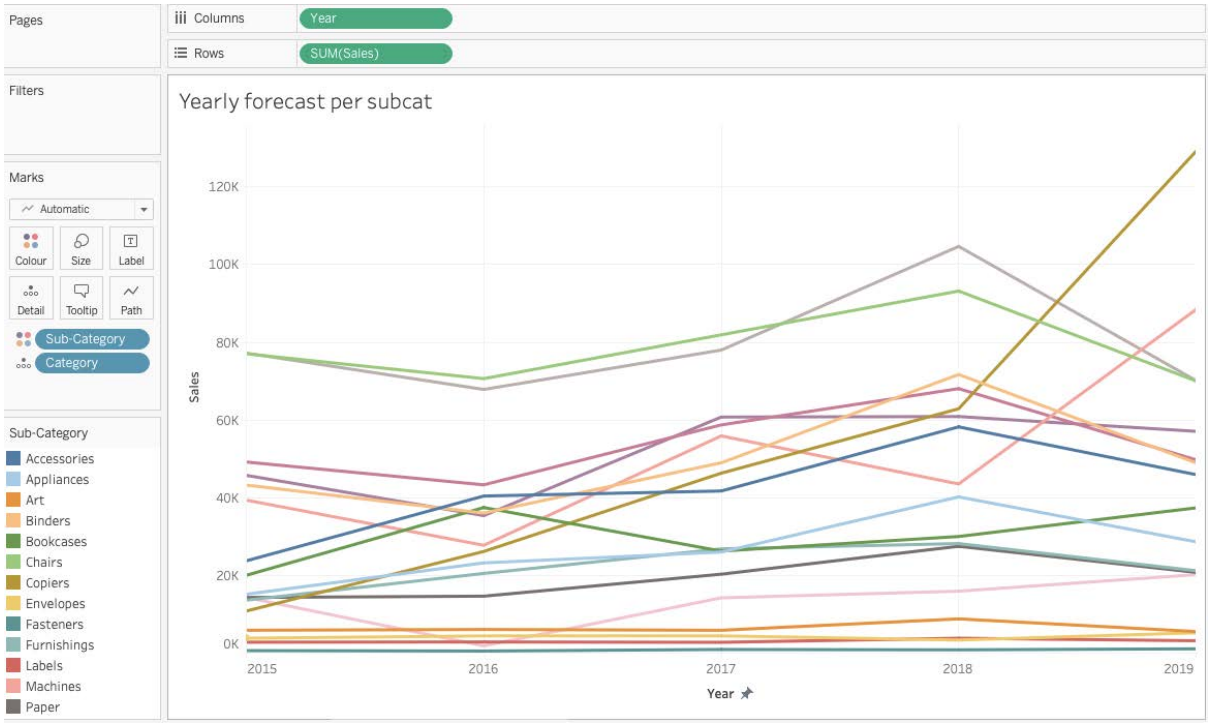

Figure 26 - Yearly sales and 1 year prediction for sub categorical dataset

Comparing all the product sub-categories appears difficult and visually unsatisfying. However, some trends can be identified. Copiers and machines were the two sub-categories with the largest predicted increase in sales, with both product groups almost doubling in predicted sales over the next 12 months. Sales focus should champion those products to increase profitability. Interestingly, the historical staples of chairs and paper were predicted to no longer be the highest revenue generating products, both with significant falls in sales.

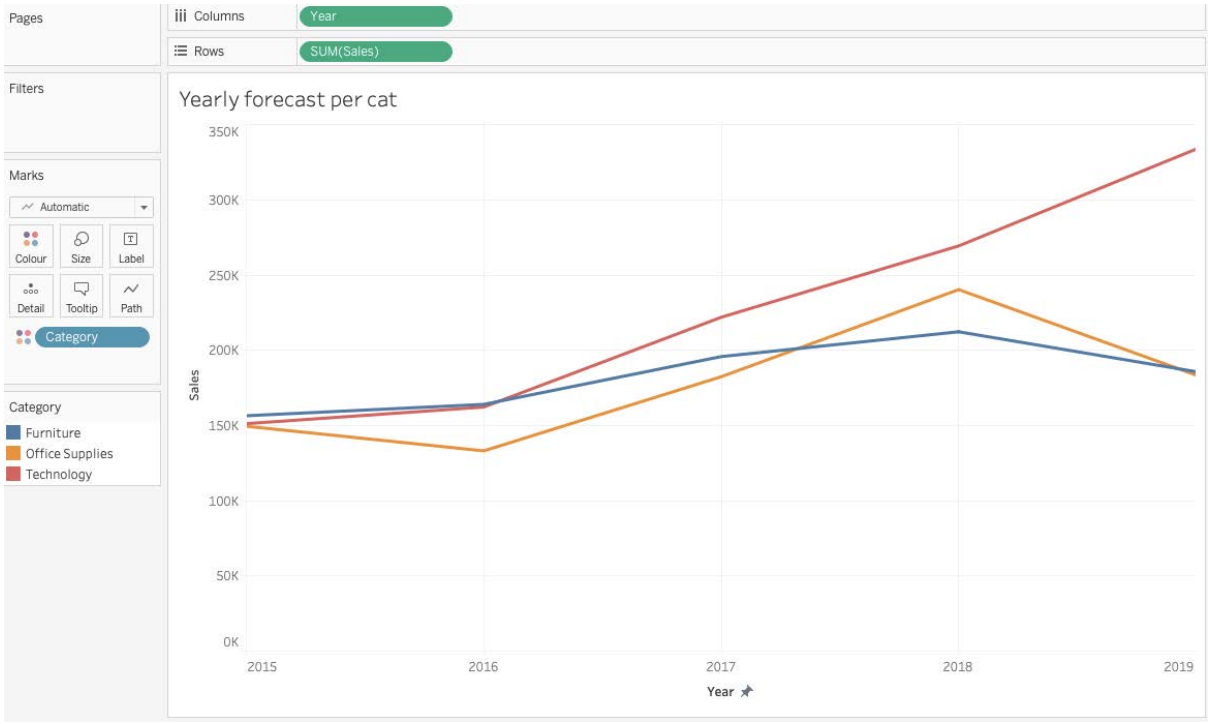

Figure 27 - Yearly Categorical estimation on Sub Categorical dataset

In both datasets, the categorical and sub-categorical grouping the yearly forecast at a categorical level show the same trend: Technology sales accelerating and office supplies sales stagnating.

Project evaluation and summary

Model Dataset grouping	RMSE (TE Validation)	Mean Sales	Future forecast RMSE (TR Validation)
Sales grouping, daily sales.	2332	1820	1064
Sales and Category grouping, daily sales	1363	788	1246
Sales, Category, Sub-Category, grouping, weekly sales	1246	766	963

Figure 28 - Modelling results

Evaluating the three models, the features applied clearly impacted results greatly. Whist the RMSE decreased, the relative RMSE vs the mean sales within each level of aggregation is concerning with error being 128, 178, and 163% of the mean respectively. In each case, the error is greater than the mean sale, which concludes that the models were unreliable. Any future work would need to address this. One method is to drastically reduce the forecasting period. With a 1 week forecast the errors dropped to 1210 for sales grouping and 811 for categorical grouping. This is a significant improvement but still very high compared to the mean. Hyperparameter tuning using a GridSearch algorithm analogous to (Zhou *et al.*, 2021) would allow the best possible tuning of the model to be performed. However, there may be some feature engineering that can be performed too to improve performance. Min, max value and rolling average have been implemented in previous studies (Dairu and Shilong, 2021) to improve forecast accuracy. This would be beneficial on a test dataset where product is defined, but in a future forecast setting where the volume of sales is not known on a daily basis it is difficult. The higher levels of aggregation to week and month eliminate this to an extent.

One of the issues with this dataset is the limited volume of sales for each product within the period, requiring a certain level of aggregation for any prediction of future sales trends. Ultimately it may be very difficult to forecast out as far as three years into the future. The model became very generic with most of the seasonal trends being smoothed out over time. Peaks in sales data contributed to the high error values in a model generalised for long-term forecast with lack of product ID and statistical values of ranges that product can be sold at.

The main findings of this report are that time features alone are not the only feature for forecasting that is relevant. In each model lag features were highly influential. More statistical data would also improve accuracy. The business findings for the project were that technology sales are the growth area for the business, and all forecasts predict continued improvement in that area.

In this report a lot of time was spent focusing on feature engineering and impact on model with feature selection. Future work needs to focus on tuning the parameters of the model. XGBoost houses a host of other boosters, loss objectives and other configurable hyperparameters. The model tuning in this report was limited to learning rate, and the number of estimators or trees. The boosted tree algorithm was not tuned, and regularization terms were not explored. Overall, this project focused on feature impact, not model tuning. Alternative models and methods of model utilisation should be explored to confirm if the long-term sales forecasting is plausible.

If the model is being tuned to short-term forecasting, employing more features such as product ID will improve performance. Analysis on regional trends could be useful to further target sales activity. The current model is very erroneous and would need to be drastically refined before deployment. From the evidence seen, deployment of this model would take the form of weekly forecast sales targets than a long-term business plan.

Including sales quantity will help understand the volume of sales and can be used for demand planning and purchasing decisions. Additionally, some measure of profit per sale would allow for profitability forecasting. Technology revenue was predicted to increase, but there were no insights into the margin of that business. Office Supplies have decreasing sales but might still be more profitable than the larger revenues seen in technology.

Deployment of forecasting models needs to be monitored as there is a risk of an “Model lead reality” whereby models predictions lead to business strategy making the models prediction self-fulfilling. If based on the evidence of this project, office supplies were massively reduced in availability, companies would need to invest in technology to overcome the barrier. This would become self-fulfilling as the demand for office suppliers drops due to low availability, creating further diversion of resources into other product sectors.

References

Burney, S.M.A. and Ali, S.M. (2019) 'Sales Forecasting for Supply Chain Demand Management - A Novel Fuzzy Time Series Approach', in *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. Karachi, Pakistan: IEEE, pp. 1–4. Available at: https://doi.org/10.1109/MACS48846.2019.9024810.

Chen, J. *et al.* (2021) 'Sales Forecasting Using Deep Neural Network And SHAP techniques', in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*. Nanchang, China: IEEE, pp. 135–138. Available at: https://doi.org/10.1109/ICBAIE52039.2021.9389930.

Choi, T.-M., Hui, C.-L. and Yu, Y. (2011) 'Intelligent time series fast forecasting for fashion sales: A research agenda', in *2011 International Conference on Machine Learning and Cybernetics*. Guilin, China: IEEE, pp. 1010–1014. Available at: https://doi.org/10.1109/ICMLC.2011.6016870.

Dairu, X. and Shilong, Z. (2021) 'Machine Learning Model for Sales Forecasting by Using XGBoost', in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. Guangzhou, China: IEEE, pp. 480–483. Available at: https://doi.org/10.1109/ICCECE51280.2021.9342304.

Kaggle (2022) *SuperStore Sales Dataset*. Available at: https://www.kaggle.com/datasets/rohitsahoo/sales-forecasting (Accessed: 18 August 2022).

Lazzeri, F. (2021) *Introduction to feature engineering for time series forecasting*. Available at: https://medium.com/data-science-at-microsoft/introduction-to-feature-engineering-for-time-series-forecasting-620aa55fcab0#:~:text=Lag%20features%20are%20values%20at,intrinsic%20information%20about%20the%20future. (Accessed: 12 August 2022).

Li, X. *et al.* (2020) 'Automatic Sales Forecasting System Based On LSTM Network', in *2020 International Conference on Computer Science and Management Technology (ICCSMT)*. Shanghai, China: IEEE, pp. 393–396. Available at: https://doi.org/10.1109/ICCSMT51754.2020.00088.

Thiesing, F.M., Middelberg, U. and Vornberger, O. (1995) 'Short term prediction of sales in supermarkets', in *Proceedings of ICNN'95 - International Conference on Neural Networks*. Perth, WA, Australia: IEEE, pp. 1028–1031. Available at: https://doi.org/10.1109/ICNN.1995.487562.

xgboost developers (2021) *XGBoost Python Package Python API Reference, Python API Reference*. Available at: https://xgboost.readthedocs.io/en/stable/python/python_api.html (Accessed: 15 August 2022).

Xu, X., Tang, L. and Rangan, V. (2017) 'Hitting your number or not? A robust & intelligent sales forecast system', in *2017 IEEE International Conference on Big Data (Big Data)*. Boston, MA: IEEE, pp. 3613–3622. Available at: https://doi.org/10.1109/BigData.2017.8258355.

Zhou, Y. *et al.* (2021) 'Sales Forecasting Using GBDT Based Model And Data Mining Method', in *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. Guangzhou, China: IEEE, pp. 398–401. Available at: https://doi.org/10.1109/ICCECE51280.2021.9342243.