BMAN73701 Programming in Python for Business Analytics Report

Student IDs: 10560694, 10806957, 10449979, 9920372, 10946957, 11453061

1 Introduction

In the fast-evolving world of modern trade, change is the only constant and that is why it is crucial to adapt to the dynamic world of sales forecasting due to its immense strategic significance. To survive in today's dynamic and competitive business environment and to ensure continuous growth, organisations need to make informed decisions by analysing potential market information with the help of forecasting. This report explores the symbiotic relationship between sales forecasting and revenue considering the significance of forecasting in business growth, inventory management, potential business revenue and organisational strategies and relationships.

In this report, we will focus on using the dataset 'Products_Information.csv' to produce a supervised machine learning model to make sales predictions. This model will be built using Python, using the sklearn module, will be trained on historic data and then used to make predictions over 16 days. When building the model, we will justify its selection, and choice of parameters, as well as evaluate its performance using metrics. As our goal is to make predictions of a variable, this is a regression task and therefore we will use a suitable algorithm to reflect this.

1.1 Objectives of Forecasting:

Forecasting is the systematic approach of predicting future trends and outcomes by analysing historical data and events using various models and methodologies. Sales forecasting not only provides mere predictions of future sales, but also helps in major strategic decisions by analysing the market trends and growth as well as considering competitors' strategies and positions. The main objectives of forecasting are:

- *Predicting Sales Volume:* Forecasting helps to predict the future sales volume and provides a basis for inventory management, budgeting and production planning. This helps the organisations to optimize their inventory management and reduces the chance of missed sales opportunities which can tarnish the brand reputation.

- *Operational Proficiency:* It provides operational proficiency by accurately predicting the market demand which prevents situations like overproduction and stockouts. This in turn helps the organisations to reduce production costs that can happen due to operational inefficiency and contributes to overall profitability.

- *Marketing Strategy:* Accurate sales forecasting helps in designing effective marketing campaigns and tailor the endorsement of promotional activities to capitalize market demands and consumer interest. By gauging the sales forecasts correctly, organisations can decide on the new product innovation and launching and pursue strategically profitable endeavours.

1.2 Significance of Forecasting:

Sales forecasting, nowadays, plays the role of a strategic imperative as it permeates all sorts of decision-making in the organisations which eventually shapes the organisational strategies and operations. Apart from its functional applications, sales forecasting helps in the decision-making process by navigating the complexities of the market. Some of the core significances of forecasting are:

- *Risk Mitigation:* Organisations are prone to various risks due to the inherently volatile market dynamics ranging from economic shifts to ever-changing customer demand. Forecasting provides organisations with a proactive mechanism to identify and assess potential risks and come up with proper mitigation plans.
- *Capitalising Opportunities:* With the help of forecasting, organisations can pinpoint potential market opportunities by analysing consumer behaviours, competitors' strategies and market trends and can take prior measures to capitalise these opportunities in their favour.
- *Supply Chain Resilience:* By predicting potential disruptions and hindrances, sales forecasting contributes to maintaining the supply chain resilience of the organisations. They can come up with agile supply chain strategies and various contingency plans to deal with the anticipated disruptions.
- *Competitive Edge:* Organisations with strong and effective forecasting tools can place themselves ahead of their competitors in the market by anticipating consumer preferences and aligning strategies to seize the right opportunities at the right.

2 Data Preparation and Feature Selection

One of the most crucial parts of predictive modelling is processing the raw data into a tidy, characterised dataset to help with the proper meaningful analysis of the data. This process includes intricate data processing and meticulous feature selection that subsequently fuels predictive modelling and allows us to build our Machine Learning model.

2.1 Data Exploration

The given dataset includes essential data regarding sales and associated promotions of 33 product types across 54 stores, dated from 01/01/2013 to 15/08/2017. To get the desired output and predict the future sales accurately, it is very important to understand the origins and the characteristics of the data. At the very beginning of the analysis, the data collection methodology and tools used for the analysis should be elucidated to avoid further confusion and ensure accurate predictions.

- *Huge Dataset:* Being a huge dataset containing data dated as early as 2013, it demanded a very careful approach to keep the integrity and usability of the data intact. To reshape the data in a systematic process, a preliminary analysis was done which highlighted the robustness of the data presenting a wealth of information of a considerably huge number of product types in 54 different stores spanning almost 5 years.

- *No Missing Value:* Another important aspect of data exploration is analysing the quality of the data, which in this case was immaculate and robust. Using the pandas module, we were able to see that there was no missing in the dataset which helped to ensure the dataset's reliability and laid a strong foundation for further analysis. This also streamlined the analytical process and helped to gain confidence regarding the integrity of the data.

2.2 Data Pre-Processing

A meticulous pre-processing is necessary for reshaping the raw data into actionable insights and this process includes data cleaning, filtering, and structuring to get the best result and accurate predictions. Due to the vastness of the given dataset, it was important to follow a strategic approach to pre-process the data enhancing its efficiency without compromising its richness.

1. *No Duplicate Rows:* First and foremost, a search for duplicate rows was conducted using Python, and no duplicate rows could be found. This reassured the reliability and integrity of the data for further analysis. It was important to search for duplicate rows as its presence may lead to biases because it is learning repeated patterns.

2. *Filtering and Creating Dataframes:* The vastness of the dataset compromised the accuracy and time required to analyse the data; hence the decision was taken to filter the data. Dates before July 31st, 2016, were not considered since the objective was to focus more on recent data to get more accurate results. Another reason we decided to filter these dates was to reduce computational costs when building a model, as having a dataframe will be very demanding and require a lot of processing power and will likely have diminishing returns. After that, the data was partitioned into two segments to create two dataframes- a training dataframe and a prediction dataframe. The training dataframe consisted of data spanned over a year (from 31/07/2016 to 30/07/2017) which we will use to build a machine learning model, whereas the prediction dataframe encompassed the data of the last 16 days which we will use to make predictions.

3. *Lagging Data:* As we delved deep into the pre-processing of the data, lagging data resulted in missing values for the first 10 rows of the training dataframe. This happened because the historical data for this period was unavailable. To deal with this issue and to maintain the integrity of the subsequent analyses, these 10 rows were dropped from the training dataframe.

4. *Transforming Data:* Another important part of the pre-processing of the data involved transforming the data of the 'products' column which contained categorical data. Using LabelEncoder from the sklearn library, the categorical data of this column was transformed into numerical data so that this can be further used in the subsequent modelling process seamlessly and efficiently.

2.3 Feature Selection

Feature selection is a very important step in the pipeline of modelling the data because it enhances the predictive performance of the machine learning model used to predict sales. In the sales forecasting in hand, the target was to predict the future sales of different product types and in doing so, identifying the correct historical features and choosing the right model were very crucial.

1. *Target Variable and Correlation Analysis:* The first and foremost step of feature selection was identifying the correlation of the target variable 'sales' with the other variables in the dataset. Creating a correlation matrix of the dataset in Python (which can be seen in Figure 2.1, reveals a significant relationship between 'sales' and 'special_offer' could be noticed which made the variable

‘special_offer’ eligible to be considered as a crucial historic feature. This also makes sense intuitively, as it is logical that the presence of a special offer leads to an increase in sales. As the remaining variables did not hold any significant correlation with sales, we will not include them.

2. *Selecting Historic Features:* Both ‘sales’ and ‘special_offer’ were considered as historic features for the dataset considering the impact of special offers on sales.

3. *Determining Optimal Time Window:* To avoid over complicating and overtraining the model, a window of 10 previous days of sales and special offers were selected, as well as the current special offer of the date, making the total number of features 21. This window of 10 days was considered as optimal to avoid excessive complexity of the model. If the model was built using a larger training window, it would risk overtraining the data making the model over complex, computationally costly and inefficient.

4. *Dataframe Modification and Lagging:* A lagging process was used to modify the existing dataframe so that the selected time window could be implemented. For each observation, the previous 10 days of sales and special offer were shifted back using a for-loop in Python, iteratively shifting the dataframe. The purpose of this was to ensure having all the features within the same row after the lagging process. Since the given data is in a long format, we grouped the data by its Product Type and Store Number in our loop, to ensure that the lagged values shift according to their characteristics This prepared to data to use in our machine learning model.

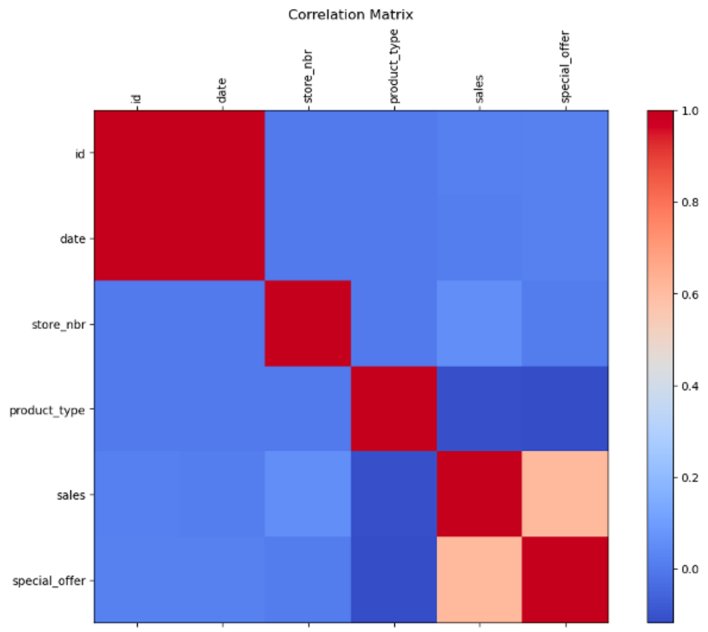

Figure 2.1: Correlation matrix visualisation between 'sales' and 'special_offer'.

3 Model Selection

For this paper, we have selected the Random Forest model. The Random Forest model is an ensemble learning technique that builds multiple decision trees during the training phase and combines their predictions; it could be used for both classification and regression.

Decision trees that make up a forest work by recursively splitting the dataset into smaller sets based on the most significant value of each node individually or in a forest.
There are several advantages to selecting this model over other models, especially a normal decision tree. Firstly, Random Forest has reasonably high accuracy as it combines multiple decision trees in the model, as a model the random forest will give results that fit quite well to the data. This is due to the fact it uses multiple decision trees; therefore, it gives more accurate results to using the singular decision tree model due to the randomness of both the subsets and features of the forest. Whilst also being very accurate, it is also quite good against overfitting values as the data is segmented to each decision tree for data training. Regular decision trees tend to overfit to the training data, as such, it would not be as useful as a predictor model as the model will be biased to any skewness in the data from the noise in the data. As the decision trees are random, it considers the noise in the dataset which reduces overfitting if enough trees are used.

Furthermore, random forest handles missing values very well. As the data set was relatively large, despite all the preprocessing that as a group we've done some NA values could slip through as random forest automates the missing NA values present in the data despite this, and if we had used a neural network, we would require a much larger training data frame - which is highly inefficient. The Random Forest model is also a lot easier to implement than many other machine models but gives results similar to that of deep learning methods (Yang, 2023). As deep learning is considered very computationally heavy, extremely expensive, and complicated to implement, Random Forest as an alternative that performs similarly to deep learning means that it is time-efficient and cost-efficient due to less high-end GPU chips being required. Furthermore, the data is more structured so a random forest would be sufficient in constructing the prediction model.

Decision trees are also considered to be non-paramedic models. Non-parametric do not rely on any assumptions or distributions from the data, as such can be very flexible in changes in data patterns for prediction models is very valuable when dealing with diverse and unknown data patterns, which fits the required usage as there are 60 product categories and 9 stores with the products sold in each store. These product categories do not have a known correlation against each other and can be completely unrelated therefore fitting the use of diverse and unknown data patterns means that the automated feature selection of the model becomes important. This also means that the decision tree model is robust against outliers in data.

The Random Forest model is also quite robust against data leakages, however, to ensure it does not happen, it was decided to use the K-fold cross-validation method. This ensures that the training and test data sets were still strictly split to avoid cross-contamination of data sets. Ensuring that there would be a less likely chance of overfitting and any adjustments can be made if there is a high variance and sensitivity. Due to this data set representing time series, it is necessary to handle temporal data carefully as there could be risks of overfitting due to the data used. As such, it is necessary to split training using past data and validate using present data to aid against overfitting.

When constructing a Random Forest model, we are required to select parameters to define the number of trees we will use, as well as the maximum depth we will allow per tree. We wanted to measure the performance of each model to see if fitted well with the data, therefore we found the cross-validation scores. Our cross-validation used 5 folds from the training data that were shuffled randomly, and each fold was used to evaluate the model using mean squared error. Mean squared error was our choice of metric as it is insightful in a regression problem as it measures the average squared distance between true and predicted values. Using our training dataframe, we have attempted different models, each of which used different parameters resulting in many inconsistencies across the cross-validation scores. This indicates the models were very sensitive, therefore we narrowed down the models that would have more consistent cross-validation scores.

Table 3.1 shows the performances of the random forest model using different parameters. Our first random forest model had 100 trees, each of which had a maximum depth of 10, and this revealed an R^2 value of 0.943 which at face value suggests a strong fitting model, however it also produced wildly inconsistent cross-validation scores on each fold, which suggests the model is highly sensitive. In this case, the train MSE is 84339.67 and the test MSE is 93631.5, when considering the nature of the data, where we are predicting sales which have large variations across different stores and product types, this is acceptable and could possibly suggest the model is somewhat fitting.

We then decided to increase the number of trees to 120 and keep the maximum depth of each tree fixed. This choice of parameters has as many problems as the previous one did, with a very similar R^2 value of 0.945 which again is very good, but the cross-validation scores did not improve and remained inconsistent, as well this the train MSE decreased significantly to 74369 whilst the test MSE remained similar. This indicated that the added trees are causing the model to memorise patterns within the training data, and it is becoming overtrained.

In conclusion, both the hyperparameters show both models are highly sensitive, and the second model is potentially overfitting our data, though that extent is moderate. This makes both unsuitable for usage. Although this does make sense, as in complex models and deep trees, it is quite common to have slightly overfitted values.

Model Parameters	Train MSE	Test MSE	R2	5-Fold Cross Validation Scores
n_estimator=100 max_depth=10	84339.67	93631.50	0.94319	[-256531.89, -151189.65, -148065.23, -91821.47, -98945.27]
n_estimator=120 max_depth=10	74369.00	90782.29	0.94512	[-253814.957, -145297.77, -143232.79, -87895.10, -93297.20]

Table 3.1: Trialing Random Forest performance using different parameters.

Learning from the previous hyperparameter tuning, we were able to make educated adjustments now to finalise hyperparameter selection. We saw that adjusting to increase the number of trees from 100 to 120 leads to overfitting, so we will choose to keep it fixed at 100. So, to ensure the model does not overfit, and to decrease model sensitivity, we will lower the maximum depth to 8. Therefore, by using our training data and setting n_estimators=100, and max_depth=8, we produce a model with metrics that can be found in Table 3.2.

Model Parameters	Train MSE	Test MSE	R2	Cross Validation Scores
n_estimator=100 max_depth=8	85160.28	93405.46	0.9474	[-99734.95, -97931.88, -99828.43, -100318.19, -197227.75]

Table 3.2: Final choice of parameters for Random Forest model.

4 Model Training

Training the model in sales forecasting is important in ensuring the correctness of predictions done because, machine learning model training based on real past sales data provides a more reliable accuracy (Ensafi et al., 2022). To train the supervised model, the dataset was segregated into *'training_data'*, which was data up to July 31st 2017, and *'predict_data'* – data beyond July 31st 2017. This split can also be interpreted as being 80% of the data for the former, and the remaining 20% used for the test set. This ratio was utilised to provide a sufficient amount of data to train the model whilst maintaining an essential part to validate the predictions as, generally, although a single data point would suffice to apply the model, the more sales data included in training, the better the forecasting accuracy (Elalem et al., 2023). Considered a fundamental step in machine learning approaches, a train-test split technique was done based on this partitioned dataset to validate the model, where '*X_train'* and *'X_test'* are features for train and test, followed by *'y_train*' and *'y_test'* which are values of the corresponding sales for training and testing (see Figure 4.1). The 80:20 train test split was particularly used because it is crucial to have a significant amount of data to train the algorithms on and allow them to capture patterns (Elalem et al., 2023). This is considered a common split as it has been utilised in various machine learning models by researchers Ensafi et al. (2022), Avuçlu & Elen (2020) and Arnoux et al. (2017). Hence, suggesting the commonality and efficacy of this approach. Using the '*fit*' method with the train sets, a Random Forest Regressor is initialised with hyperparameters of a maximum depth of 8 per tree, and 120 trees altogether (see Figure 4.2).

```
#Partioning data to use for training and making predictions
training_data=df[df['date']<'2017-07-31']
predict_data=df['2017-07-31'<=df['date']]


#Train Test Split
X_train, X_test, y_train, y_test = train_test_split(training_data[features], training_data['sales'], test_size=0.2, random_state=42)
```

Figure 4.1: Train-test Split partition.

```
#Random Forest
model=RandomForestRegressor(max_depth=8,
                            n_estimators=120)
model.fit(X_train,y_train)
```

Figure 4.2: Random Forest Regressor.

5 Model Evaluation

With reference to the dataset, the Random Forest model underwent an elaborate evaluation procedure to gauge whether its predictions are capable of forecasting sales or not. Appropriate metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE) and coefficient of determination (R^2) values were used to assess the accuracy of the model's performance. Altogether, these commonly used metrics provide an understandable model evaluation of the different types of errors encountered in predicting sales, whereby MSE is used for model accuracy, MAE for robustness, and R^2 values to explain the percentage of variances in data. However, it is imperative to consider that these metrics alone are not enough to explain the model's efficiency due to other factors like dataset biases of human behaviour and in the machine learning models themselves.

Based on the MSE values, which is the average squared difference between the predicted and real sales, results obtained from this evaluation suggested that the Train MSE and Test MSE were 85160.28 and 93405.46, respectively (see Table 3.2). The rather higher Test MSE value than Train MSE implies that the model may be marginally overfitted, but not necessarily problematic because the small difference may still suggest that it generalises well to new data. Potential overfitting should however be addressed as it could lead to a less robust prediction in real-time data, thus affecting the model's performance. Theoretically, the smaller the difference between the train-test MSEs, the more balanced the model is.

Therefore, an area of improvement which could be done to mitigate the difference is through approaches like fine-tuned hyperparameters and regularisation. For example, through a grid search method or random search method which could assist in finding the most suited combination for the forecast in terms of depth of trees. Moreover, utilising ensembles with multiple Random Forests of varying hyperparameters and ensemble approaches may also facilitate better model predictions. A k-fold cross validation (k =5) was performed on the training set and the Cross Validation (CV) scores represent the consistency of the model's performance in the training data. The CV scores of [-99734.95, -97931.88, -99828.43, -10031819, -197227.75], suggested that it was generally consistent except for the final value (-197227.75). This anomaly could be due to several reasons and could indicate slight sensitivity in our model or could be due to differences in data distribution within the fold itself. This could have been further investigated during the modelling process but due to time constraints, we were not able to find the cause of this inconsistent score. Ideally, the lack of consistency should be explored because it impacts the robustness and reliability of the model across subsets.

Next, the sales were predicted through the trained model which facilitates understanding on how the model is run on unseen data, using our *'predict_data'* dataframe. Results suggested a Prediction MSE of 71685.83, and Mean Absolute Error of 82.19 on new data, which indicate moderately accurate predictions (Figure 5.1). The mean absolute error is particularly insightful as it measures the average difference between the real and predicted data, which is more intuitive than the MSE. In addition, the model score (R^2 value) assesses the goodness of fit on the test datasets by suggesting how well the model explains variances in data. The R^2 value of 0.947 shows that the model is determined by 94.7% of the variance in our model's sales data. In other words, a great extent of variability is explained by the model hence indicating a strong fit between data and the model.

```python
#Making sales predictions
y_pred=model.predict(predict_data[features])
y_true=predict_data['sales']

#Performance Evaluation
mse=mean_squared_error(y_pred, y_true)
print('Prediction MSE = ' + str(mse))

mae=mean_absolute_error(y_true, y_pred)
print('Prediction MAE = ' + str(mae))

score=model.score(X_test, y_test)
print('Model Score (R^2) = ' +str(score) )
```

```
Prediction MSE = 71685.92528482924
Prediction MAE = 82.18826466144003
Model Score (R^2) = 0.9473841898617944
```

Figure 5.1: Insightful Metrics.

In summary, the accuracy of the model was evaluated through the chosen metrics. Potential overfitting or data leakage were also investigated through the MSEs of train-test sets and other metrics like cross-validation. Although this procedure suggested that the model showed significant performance in forecasting sales, a more thorough modelling is pivotal for more accurate and reliable predictions. This can be overcome by allowing more time for the modelling process where improvements can be further done through exploring which models and features may be most efficient in prediction-making and investigating anomalous values.

Moreover, despite the popularity of selecting Random Forest as a sales prediction model, other models could have also been explored, tested, and compared with the MSEs, MAEs and R^2 to understand its strengths and limitations which potentially contribute to the reliability of the model. For example, Martins & Galegale (2023) in their forecast of 1,510,563 sales transactions in a supermarket agreed that Random Forest showed a 65% accuracy based on MSE comparisons with Support Vector Machines (15%) and traditional Linear Regression (6%). However, in a comparative study of machine learning models applied onto retail sales forecasting, Mitra et al. (2022) found that hybrid methods of Random Forest and XGBoost together can address overfitting issues to a much greater extent than individual models like RF alone, providing greater accuracy to sales predictions. This implies the importance of different model selections and evaluations. Contrary to that, however, a limitation worth considering is that the potential overfitting might also be due to the complex model, therefore a simpler model should have also been attempted to mitigate it. For instance, Shalev-Shwartz & Ben-David (2014) covered the implications of complex models which lead to overfitting. Lastly, using different time points to validate the model could have also been useful to check if it performs consistently throughout the sales period.

6 Results

The results of the predictor model tracks quite well to the actual sales results over the aggregate data set as seen in Figure 6.1. At the beginning of August 2017, the true sales were higher than the predicted sales on the 1st and 2nd of August being the highest deviation for true sales being higher than the predicted sales. For the rest of the time period, the predicted sales and the actual sales tracked quite well with each other. The predicted sales are slightly higher than the true sales. The predictor model had an MSE of 71,685 lower than the training and test data of the training set's 85,160 and the testing data's 93,405 meaning that the predictor model generalises better to our new data set than the training and test data. The prediction MAE of 82.1 is quite consistent with the training and test MAE of 78.94 and 81.45 respectively meaning the prediction model with the amount of data points that is predicted is quite accurate and the error is quite low. Expanding on the model, with individual product categories in stores, the model tracks quite well as seen in Figure 6.2 where the general changes in sales quantity are reflected very well from the predicted model. Even though the true sales quantity is generally less than what was predicted, this trend is very consistent and tracks very well towards any changes in true sales quantity with the largest deviation being from the 1st, 2nd, and 12th of August in terms of absolute values.

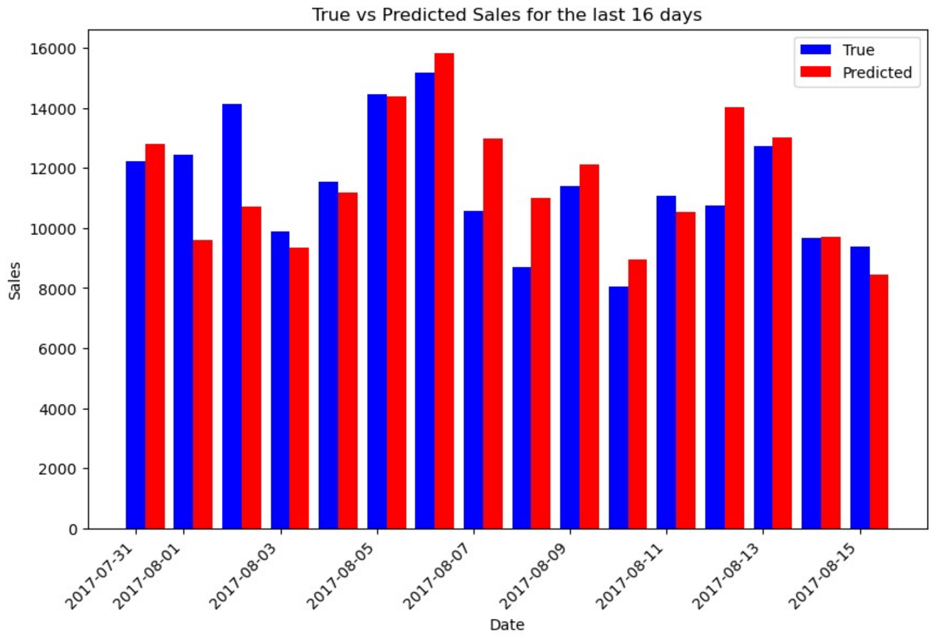

Figure 6.1: Barplot showing the aggregated sales across the 16l days, where the blue charts are the true sales and the red plots have been predicted.

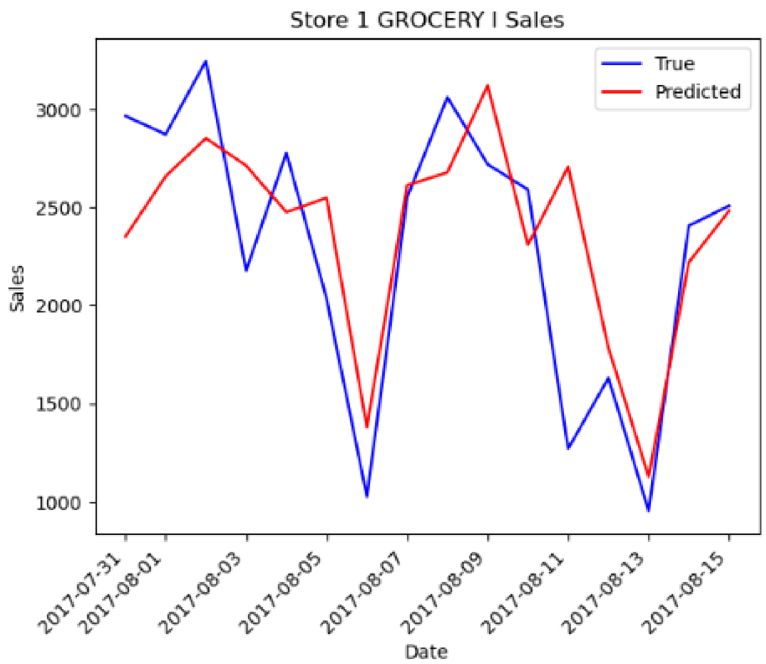

Figure 6.2: Example plot produced by the model, focusing on store 1 GROCERY I sales.

7 Conclusion

In this report, our comprehensive analysis has shed light on the relationship between predictive modeling and informed decision-making for company ABC. The fast-paced and ever-changing nature of modern trade demands adaptive strategies, such as the use of machine learning models. In this instance, Random Forest emerged as a potent tool for mostly accurate sales predictions and allowed for operational proficiency, the design of effective marketing campaigns, and risk mitigation.

Accurate and reliable predictions would not be possible without appropriate data preprocessing, and feature selection. In preprocessing, the dataset was searched for missing or duplicate values considering strategies such as removal of rows or imputation, but these were not needed as the dataset proved to be complete. The data was filtered to the last year to cut computational costs and categorical data from the 'products' column was transformed to numerical data. Feature selection was informed mainly through the use of a correlation matrix between the target variable ("sales") and other columns of the data frame. columns with the highest correlation were selected.

The choice of the Random Forest model was justified by, offering advantages such as high accuracy, resilience against overfitting, and robust handling of missing values. A brief comparison was made between random forest and other models such as neural networks. Several combinations of parameters were compared for the model, considering trade-offs between complexity and consistency, and the final configuration demonstrated promising results.

The carefully partitioned dataset proved useful in model training. An 80:20 train-test split facilitated robust training whilst allowing for effective validation. The model showcased strong predictive capabilities (maximum depth of 8 per tree, and 120 trees altogether), as evidenced by the Mean Squared Error, Mean Absolute Error, and R^2 values.

In the evaluation phase, we critically examined potential overfitting and model sensitivity. The mean squared error, mean absolute error and R^2 values were observed and interpreted. The slightly higher Test MSE indicated a marginal overfitting, prompting consideration for further refinements. The R^2 scores showed a strong fit between data and model. The k-fold cross-validation scores, though generally consistent, highlighted a need for exploring the source of an inconsistent fold, an area that warrants attention in future iterations. Further limitations of the model were highlighted and discussed.

The results were visualised and discussed, revealing that the model tracks closely to actual sales data. While minor deviations were observed, especially on the specific days of 1st and 2nd of August, the overall predictive accuracy was high and the trends on the bar chart were generally aligned, instilling confidence in the model's efficacy. The chosen metrics, coupled with visual representation, offer a nuanced understanding of the model's performance.

In conclusion, this report not only highlights the technical aspects of building a predictive model but also underscores the strategic implications for ABC Company. The steps from data exploration, preprocessing, and feature selection to model selection, training, and evaluation have equipped us with valuable insights and a model with reasonable predictive power. Further refinement and exploration of alternative models can be used for future enhancement, ensuring that the sales forecasting system evolves in tandem with the needs of a dynamic business landscape.

Bibliography

Elalem, Y. K., Maier, S., & Seifert, R. W. (2023). A machine learning-based framework for forecasting sales of new products with short life cycles using deep neural networks. *International Journal of Forecasting*, 39(4), 1874-1894.

Ensafi, Y., Amin, S. H., Zhang, G., & Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning–A comparative analysis. *International Journal of Information Management Data Insights*, 2(1), 100058.

Martins, E., & Galegale, N. V. (2023). Sales forecasting using machine learning algorithms. *Revista de Gestão e Secretariado,* 14(7), 11294-11308.

Mitra, A., Jain, A., Kishore, A., & Kumar, P. (2022). A comparative study of demand forecasting models for a multi-channel retail company: a novel hybrid machine learning approach. *In Operations Research Forum* (Vol. 3, No. 4, p. 58). Cham: Springer International Publishing.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Treyhan, D. (2020). "Why Choose Random Forest and Not Decision Trees." Towards AI. Retrieved from https://towardsai.net/p/machine-learning/why-choose-random-forest-and-not-decision-trees

Singh, P. (2022). "Data Leakage in Machine Learning: How Could It Be Detected and Minimize the Risk." Towards Data Science. Retrieved from https://towardsdatascience.com/data-leakage-in-machine-learning-how-it-can-be-detected-and-minimize-the-risk-8ef4e3a97562

Yiu, T. (2019). "Understanding Random Forest." Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J., (1984). *Classification and regression trees*

Speicer, J.L, Miller, M.E., Tooze, J., & Ip, E., (2018) A comparison of random forest variable selection methods for classification prediction modeling