Retail Price Prediction XGboost Vs Random Forest Comparison

1.0 Introduction

Data is growing at an exponential rate with 90% of the world's data being generated in the last two years alone. Businesses need the ability to track the data they have, understand the trends and patterns so its can be used for effective distribution of resources and revenue generation. Machine learning is a field of AI (Artificial Intelligence) by using which software applications can learn to increase their accuracy for the expecting outcomes. Here machine learning will be used for price prediction.

2.0 Objective

One challenge of modeling retail data is the need to make decisions based on limited history. Holidays and select major events come once a year, and so does the chance to see how strategic decisions impacted the bottom line. In addition, markdowns are known to affect sales the challenge is to predict which departments will be affected and to what extent.

The aim is to forecast weekly sales from a particular department. The objective of this case study is to forecast weekly retail store sales based on historical data. This model will be able to inform shareholders given a set of variables approximately what kind of sales they could hope to get. How these features interact with each other and what the sales outcome would be with these interactions. This is essential as it can help shareholders in determining what type of stores to invest in, their location and

3.0 Methods

This entails two analysis which are Descriptive analysis as well as predictive analysis. In Descriptive Analytics, statistical methods of data collection, analysis, interpretation, and data visualization to look at what's happened in the past this will be covered in the data exploration section. While in Predictive Analytics the data past data will be trained so that patterns can be learned by the algorithm to predict the possibility of future events.

3.1 Data

The historical sales data for 45 stores located in different regions - each store contains a number of departments. The company also runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of which are the Super Bowl, Labor Day, Thanksgiving, and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks. The data set consist of three tables called stores, features and sales respectively.
The store dataset contained the size in sq.ft. and type of store which was a nominal categorical variable. Called A, B or C. This table consists of 45 samples with 3 variable. The features data set contained 8190 instances with 12 features. Contains additional data related to the store, department, and regional activity for the given dates. The variables include :

- Store - the store number
- Date - the week
- Temperature - average temperature in the region
- Fuel_Price - cost of fuel in the region

- MarkDown1-5 - anonymized data related to promotional markdowns. MarkDown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA
- CPI - the consumer price index
- Unemployment - the unemployment rate
- IsHoliday - whether the week is a special holiday week

The sales data set consists of 421,570 instances and 5 variables. Which are:

- Store - the store number
- Dept - the department number
- Date - the week
- Weekly_Sales - sales for the given department in the given store
- IsHoliday - whether the week is a special holiday week

3.1 Data Preparation

Here first set is to merge the 3 data sets. Which produced a final table size of 421,570 instances and 14 features. Since the date column was not in the correct format that was changed into date time object which is a module recognized and python and can be used for manipulation of dates and times. In this study we will not use time in chronologically (time series) but to create a new variable called "month" to determine the sales by month. Nulls were visualized, it was found the only missing values were in the dates related to markdown which are expected as sales are not run daily. These were changed to 0 numeric values. Duplicates were removed.

3.2 Data Exploration

The objective of data exploration is to visualize the data to see which algorithm may more suitable. After looking at the pair plot we see that the variables are not correlated and do not seem to have a linear pattern. Also the distributions are not normal hence a linear would not be suitable. there for a tree based model will be a better fit. In this dataset all features except markdown1 and markdown4 have low correlation. From the Pearson correlation heat map we can see there is a 0.84 correlation between these two variables.

For many machine learning algorithms, using correlated features is not a good idea. It may sometimes make prediction less accurate, and most of the time make interpretation of the model almost impossible. GLM, for instance, assumes that the features are uncorrelated. Fortunately, decision tree algorithms (including boosted trees) are very robust to these features. Therefore we have nothing to do to manage this situation.

One hot encoding was performed on variables: IsHoliday, Department, and store. Their were over 80 departments and 40 stores so this action increased the features significantly. This is to change the variables to nominal numeric features.

Also from the data exploration we find that department 77 and 99 with very low weekly sales seems to have very high markdowns of all types While departments 38, 95, 92 and 40 have high weekly sales low markdowns. Also it was noticed that the holidays increase the sales for type C stores more than the other store types. Dept_92 has the highest gross weekly sales followed by Dept_95.

In fig 1 we can see that department 99 has low weekly sales since most data points are below the 2000 sales range. The size of the points indicate markdown4 size. This implies that more markdowns are given to this department since it has low sales. In fig 2 that is a depiction of weekly sales to the store size with the size of the points displaying the markdown4 sales it is also seen here that markdowns are most prevalent where the weekly sales are low.

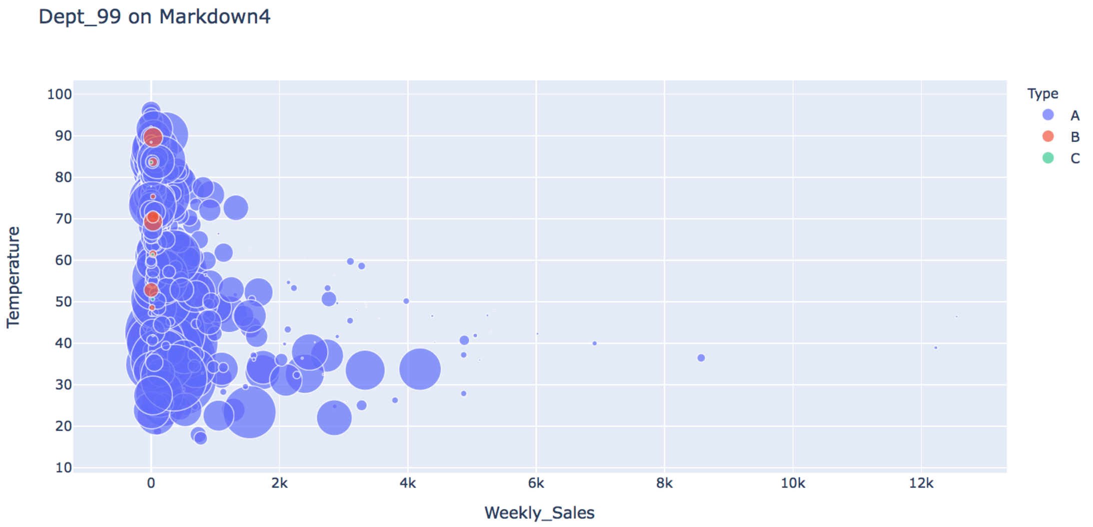


Fig1

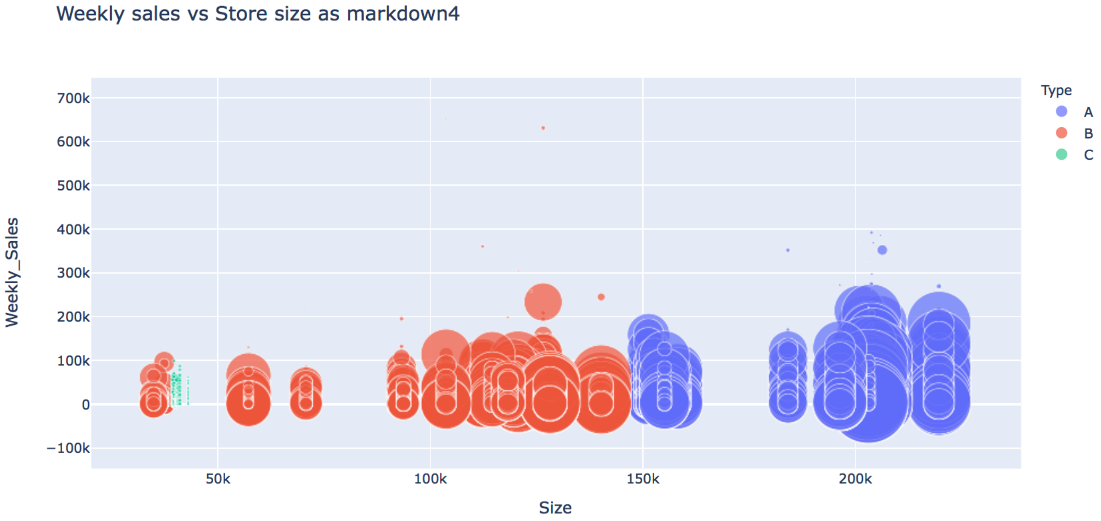


Fig2

3.3 Hyperparameter tuning

XGBoost Regressor hyperparameters checked included max_depth, n_estimators and learning rate. max_depth indicates the depth degree of the estimators (trees in this case). This parameter should be tuned with caution as it can cause the model to overfit. n_estimators which are the number of estimators the model will be built upon. Learning rate is used to control and adjust the weighting of the internal model estimators. The learning_rate should always be a small value to force long-term learning. An sklearn auto tuning package (GridsearchCV) was selected to iterate over the chosen values and the best value was selected, which was 100 number of estimators, 1.0 learning rate, max depth of 5 trees.

While the hyperparameters chosen for Random Forest Regressor were n_estimators which are The number of trees in the forest, max_depth which specifies the maximum depth of the tree, max_features which is the number of features to consider when looking for the best split. Max features is useful for reducing the training for data sets with large number of features, or can be helpful in curbing overfitting since a higher selection of trees can help reduce overfitting. The maximum tree variable is another metric that reduces overfitting.

3.4 Model

sklearn XGBoostRegressor
sklearn RandomForestRegressor
(Demo) AWS built-in XGBoost algorithm

XGBoost is a supervised learning algorithm and implements gradient boosted trees algorithm. The algorithm work by combining an ensemble of predictions from several weak models. it gives more importance to misclassified observations. Intuitively, new weak learners are added to concentrate on the areas where the existing learners are performing poorly. The contribution of each weak learner to the final prediction is based on a gradient optimization process to minimize the overall error of the strong learner. The algorithm reduces the residual error by comparing previous tree. Hence, the tree that grows next in the sequence will learn from an updated version of the residuals. Boosting happens to be iterative learning which means the model will predict something initially and self analyses its mistakes as a predictive toiler and give more weightage to the data points in which it made a wrong prediction in the next iteration.

Random Forest is a bagging technique that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.
Bagging decreases overall variance by averaging the performance of multiple estimates. Aggregate several sampling subsets of the original dataset to train different learners chosen randomly with replacement, which conforms to the core idea of bootstrap aggregation. Bagging normally uses averaging for regression.

Note: Sampling method used was 5 fold crossvaligdation to ensure that every portion of the data was tested and the average of the final result was used as the accuracy.

4.0 Results

	RMSE	MSE	MAE	R2	Adj R2
XGboost	7730.35	59758320.0	3579.4214	0.9208	0.92046
Random Forest	18142.333	329144236.59	11968.37	0.5638	0.5619
Sagemaker XGboost	7318.029	53553548.0	4309.23	0.89331	0.89285
Sagemaker XGboost After HPO tuning	4316.144	18629094.0	1806.656	0.96289	0.9627

4.1 Metrics

RMSE: Root Mean Square Error is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2}$$

Root Mean Squared Error

MSE: Mean square error is calculated as the mean or average of the squared differences between predicted and expected target values in a dataset. One problem with this metric is that it is not robust to outliers.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

Mean Squared Error

MAE: Mean absolute error is a very simple metric which calculates the absolute difference between actual and predicted values. MAE is not sensitive towards outliers and given several examples with the same input feature values, and the optimal prediction will be their median target value. This should be compared with Mean Squared Error, where the optimal prediction is the mean. A disadvantage of MAE is that the gradient magnitude is not dependent on the error size, only on the sign of y - ŷ. This leads to that the gradient magnitude will be large even when the error is small, which in turn can lead to convergence problems. Most useful when there are outliers in the data.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|$$

Mean Absolute Error

R2: R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable.

An R-squared of 100% means that all movements of a security (or other dependent variables) are completely explained by movements in the index (or the independent variable(s) you are interested in.

$$R^2 = 1 - \frac{SSE}{SST}$$

R2 Score

Adj R2: Adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared increases when the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected. Typically, the adjusted R-squared is positive, not negative. It is always lower than the R-squared.
Adding more independent variables or predictors to a regression model tends to increase the R-squared value, which tempts makers of the model to add even more variables. This is called overfitting and can return an unwarranted high R-squared value. Adjusted R-squared is used to determine how reliable the correlation is and how much it is determined by the addition of independent variables.

$$R^2_{adjusted} = [\frac{(1-R^2)(n-1)}{n-k-1}]$$

Adjusted R-Squared

4.2 Interpretation

The Packages used for interpretability is SHAP. SHAP (SHapley Additive exPlanations) is a method to explain individual predictions. SHAP is based on the game theoretically optimal Shapley values. TreeSHAP, an efficient estimation approach for tree-based models. SHAP comes with many global interpretation methods based on aggregations of Shapley values. The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalition game theory. The feature values of a data instance act as players in a coalition.

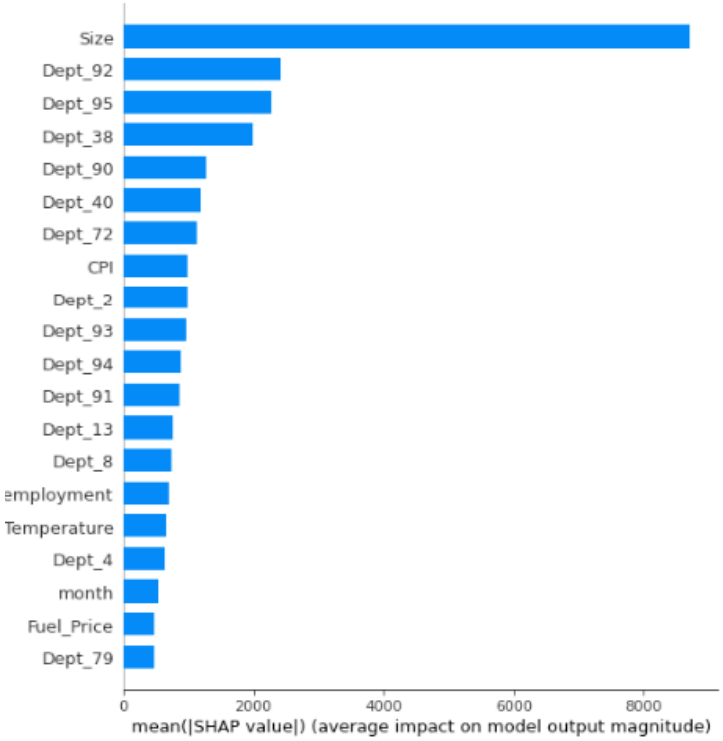


Fig3

Shapley values tell us how to fairly distribute the “payout” (= the prediction) among the features. The plot below shows the feature importance of the most influential features in the model. The model chosen for interpreting is the Xgboost model as this is the best performing model. These value showed on the figures are predicted values from the test set.

Fig3 shows that the feature size was the most important feature in the Xgboost model. This model depicts that the size of the store change the store price prediction averagely at a rate of $8000. On the x axis. Following behind is the Dept_92 which seems to change the price predictions averagely about $2200.

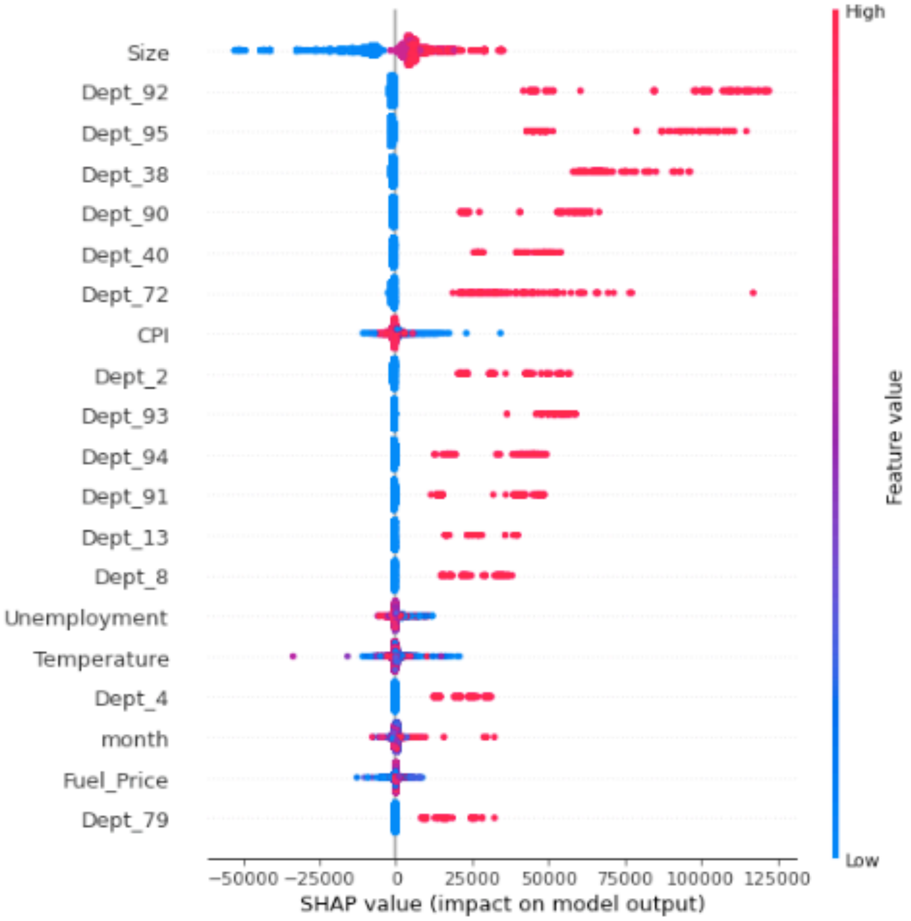


Fig4

Fig4 is a summary plot which combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The color represents the value of the feature from low to high. Overlapping points are jittered in y-axis direction, so we get a sense of the distribution of the Shapley values per feature. The features are ordered according to their importance.

Fig4 smaller size of the store (Size Feature) the reduce the price predicted for that store. For the next most important metric the Dept_92 means that when the store has department 92 (ie the binary variable is set to 1) then it increase the price the prediction.

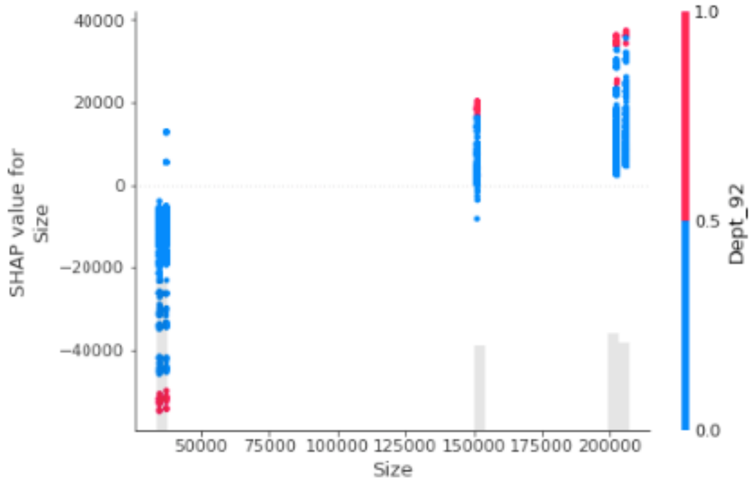


Fig5

Fig5 is called a SHAP feature dependence plot. The interaction effect is the additional combined feature effect after accounting for the individual feature effects. This features automatically colors the feature with the strongest interaction that variable has. Fig5 shows that at very low sizes of the store (Size feature) the store having department 92, reduces the size predicted of that store, also at very large sizes of the store the store having department 92 (Dept_92 = 1) it increases the probability that the store is large in size.

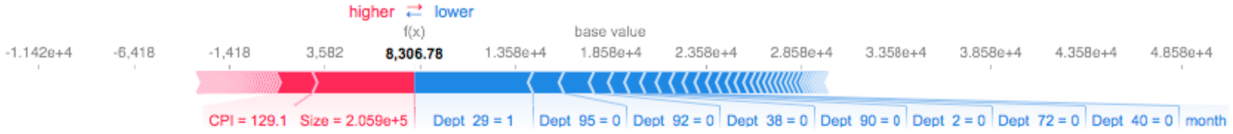


Fig6

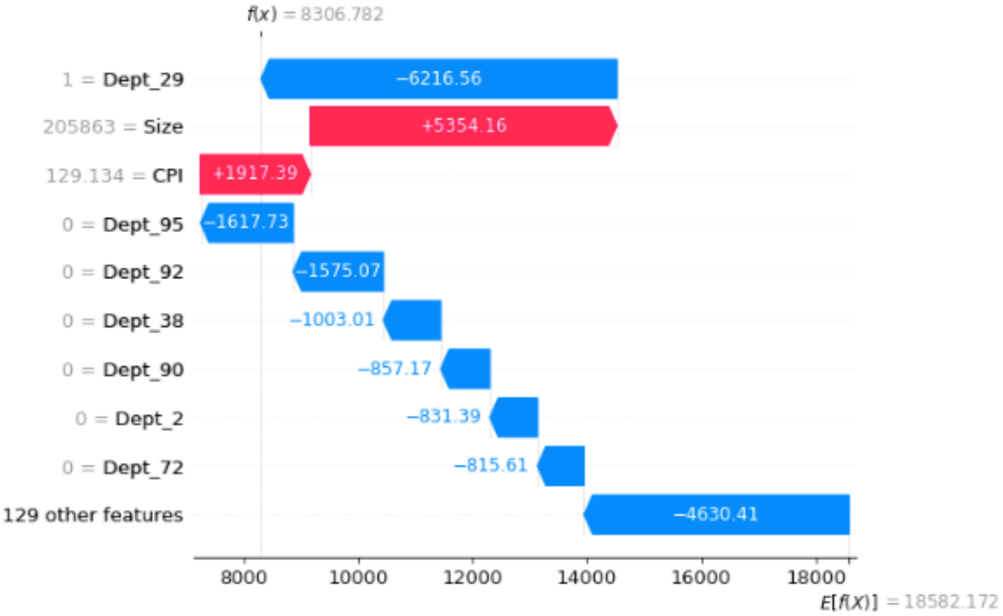
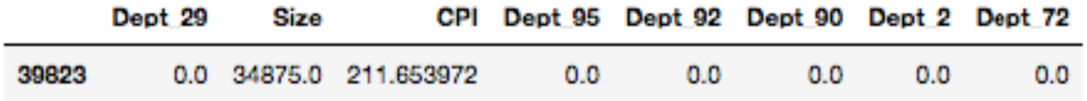


Fig7



	Dept_29	Size	CPI	Dept_95	Dept_92	Dept_90	Dept_2	Dept_72
39823	0.0	34875.0	211.653972	0.0	0.0	0.0	0.0	0.0

Fig8

Fig6 and Fig7 are a depiction of one individual test point from the predicted values, The same point was selected for both figs. Also these plot are portraying the same message in different ways. The selected test point values are shown in Fig8. It shows that the baseline the average predicted probability is $18,582.17. however this store had a sales price prediction of $8,306.78. this was majorly because the Dept_29 which had the highest effect on this store was absent (binary variable set to 0) which subtracted $6216.56 from the baseline. The size the store increased the sales prediction by $5354.16, while most other features subtracted from the baseline causing this store to have a low sales prediction compared to the baseline.

5.0 Conclusion

Xgboost model outperformed the Random forest model by and also had a faster run time during the hyper-parameter tuning. The the most important features in deterring the sales outcome are the size, some departments, CPI, unemployment and temperature. Size being the most important feature followed by department 92 that influences the sales potential of a store.

References

1. https://christophm.github.io/interpretable-ml-book/shap.html
2. https://www.kaggle.com/manjeetsingh/retaildataset/activity
3. https://shap.readthedocs.io/en/latest/example_notebooks/tabular_examples/tree_based_models/Force%20Plot%20Colors.html
4. https://medium.com/geekculture/xgboost-versus-random-forest-898e42870f30