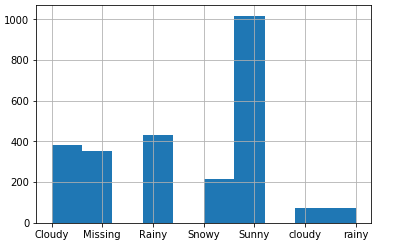
Seoul Store data analysis

Initial Data Inspection: The data has dimension 2547x23.

Handling Missing Values: There are two columns with missing values. ‘Outlook’ and ‘Japanese Tourists’

Column ‘Outlook’ represents the type of weather a particular day has but out of the 2457 rows 356 have missing values. When we see the distribution of the different categories in the column we see that “Sunny” is the most common type. We Have imputed the missing values with a new value called ‘missing’. This is the most effective way according to us because it does not change the data. Other methods such as imputing the most probable value or a predictive model would create an artificial correlation that would not be representative of the actual data.



Column Japanese Tourists records the number of Japanese tourists that come to Seoul every day. But of the 2457 rows 1166 rows have missing values which is almost half of it. The simplest and best way to impute this data would be to replace with the mean of the column. We did this on the store wise data set to create some variability in the data rather than imputing half of the dataset with the same value.

Data Labelling: Some columns in the data requires labeling them to convert into a numeric value because there are some models which require the data to be numeric. These columns are ‘Weekday’, ‘Outlook’, and the distances from stations.

Data Partition: We are dividing the store data into train and test data using K Fold splitting with number of splits set as 10. This makes the train and test data more useful because the order of split is random and thus prevents overfitting and trains the model better. We need KFold and cross\_val\_score packages for this.

**Linear Regression:**

We are using the simplest model of predicting the values of total sales per store based on the features. We are using cross\_val\_predict function from the sklearn model selection library to make it simple.

The measure of linear regression accuracy is the error term that is obtained from the predicted and actual value. There are many score but we are using coefficient of determination for the purpose.

Store A= 0.9660726237098108

Store B= 0.9427800278397543

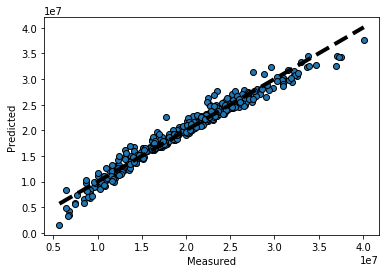
Store C= 0.9538113248981983

Store D= 0.9755066367843688

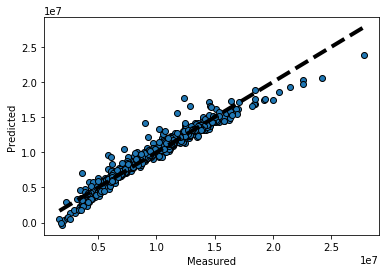
Store E= 0.9117663059602787

Evidently all models are good but we can see that we are getting very good results for store C and A.

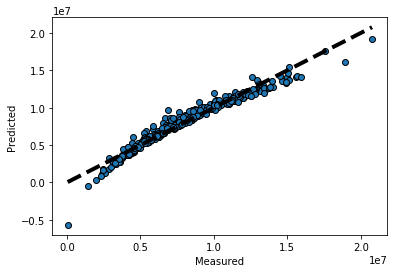
Linear Model: Store A



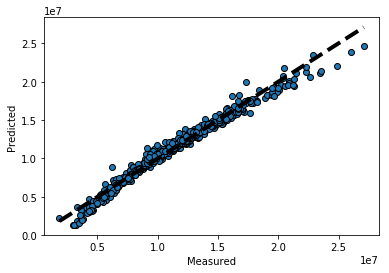
Store B:



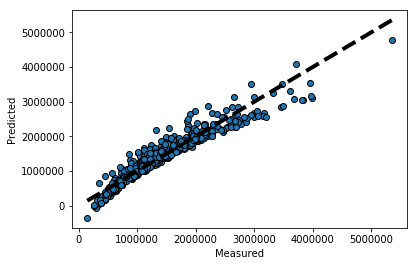
Store C:



Store D:

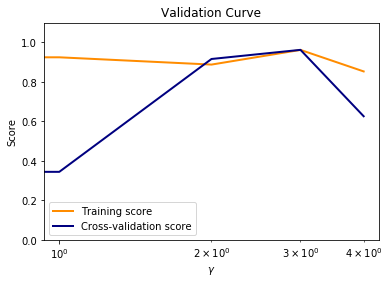


Store E:



**Cross validation score comparison:**

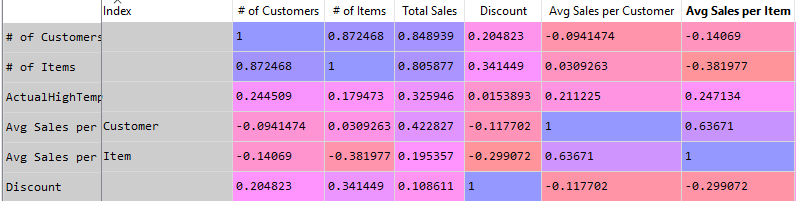
We used cross\_val\_score to identify the change in predictions that happen in train data and test data. We started out by calculating the score of every store and combining them together in a dataframe. We plot the variation in graph to see the variation.



We can see that the score in training set is fairly constant but the score of the test data is high for store 3 and 4 which are labelled form of store C and D.

**Correlation Analysis:**

We used a simple correlation function in python to find out the correlations every variable has with respect to each other.



This shows a clear correlation but expected correlation between total sales and number of people visiting the stores as well the averages. So what does total sales depend upon the most in terms of other variables?

**# of Customers, # of Items, Avg Sales per Customer, YenWonRatio**

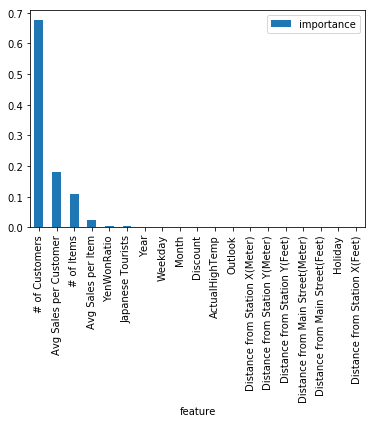
So the exchange rate affects the sales of the stores. But as we looked more closely across all the stores we found that there is an exception. Store E total sales has a negative correlation with the yenwonratio of -0.0041356. So for store E it seems that the exchange rate is not really factor that drives sales. And because of the same reason the number of Japanese tourists also have a very little effect on the store.

There are some more insights from the correlation analysis that the discount and number od items sold are closely related. Mostly because of

**Random Forest Regressor Analysis:**

We will be using the random forest regressor package from sklearn.ensemble library. This model is very effective because it creates a number of decision trees on various subsamples of the data and averages to prevent overfitting and improve accuracy.

By fitting the store A data into the model and using feature\_importance\_ method we will be able to figure out the most important features.

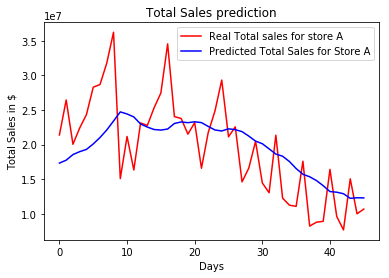


If we do the same for other stores we will be find a similar ordering.

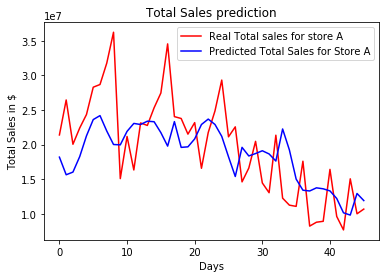
Neural Network Analysis:

So we created a recurrent neural network that predicted the total sales of store A based of features. We can compare the results obtained from different features and see which feature has been affecting sales the most and which one has not impacted sales at all.

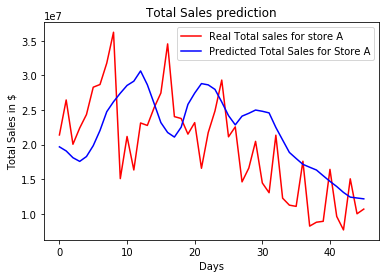
1. The first one is the most simple neural network which just selects the 10 previous total sales values and predicts the 11th value. And as expected it predicts without much variation.

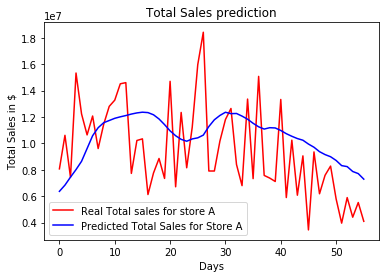


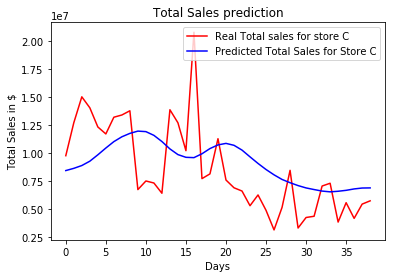
1. This one takes into account the first five features which are number of items, number of customer, avg sales/ number of item, avg sales/ number of customer and discount. This is the better model which fits the real value quite nicely and accounts for the variations too.



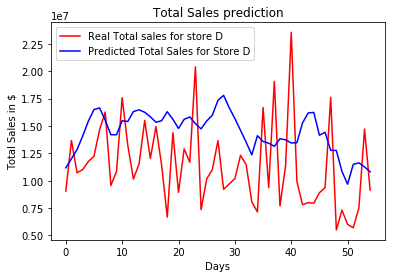
1. When we take just the Number of Japanese Tourists and Total sales into the neural network we get an accuracy of 96%. Which suggests that Japanese Tourists have a very high impact on the sales of Store A.



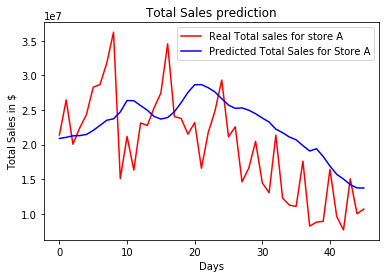
1. In store B for the Japanese Tourists and Total sales, the accuracy that we get is almost 90% which is less than store A so we can say that Store A has more correlation between these two features. 
2. Store C has the best fit with an accuracy of 98%.



1. Store D also has a good relationship between Japanese Tourists and Total Sales. Accuracy= 95.5%



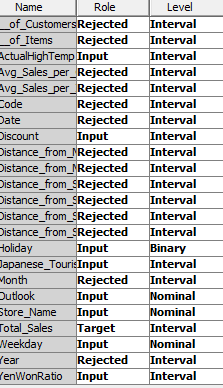
1. When we get features such as Total sales, Discount, Holiday, YenWonRatio and Japanese Tourists the accuracy of the model for Store A is 77%. This is almost same across all the stores.



Decision Tree Analysis:

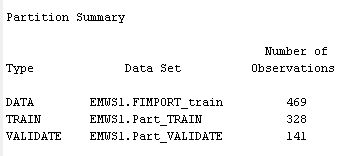
We have used the SAS Eminer tool for decision tree analysis, because it is much easier and has more options to check model variability.

To set up the data before performing model comparison we have to set up the types of feaures that we choose in the model

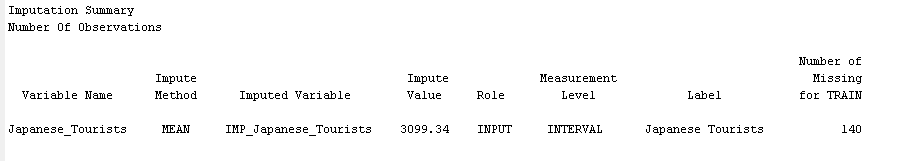


We reject some columns such as distances from station, codes, Months, Years, Avg sales etc. This makes the tree more useful because then there are more useful features to make decision on.

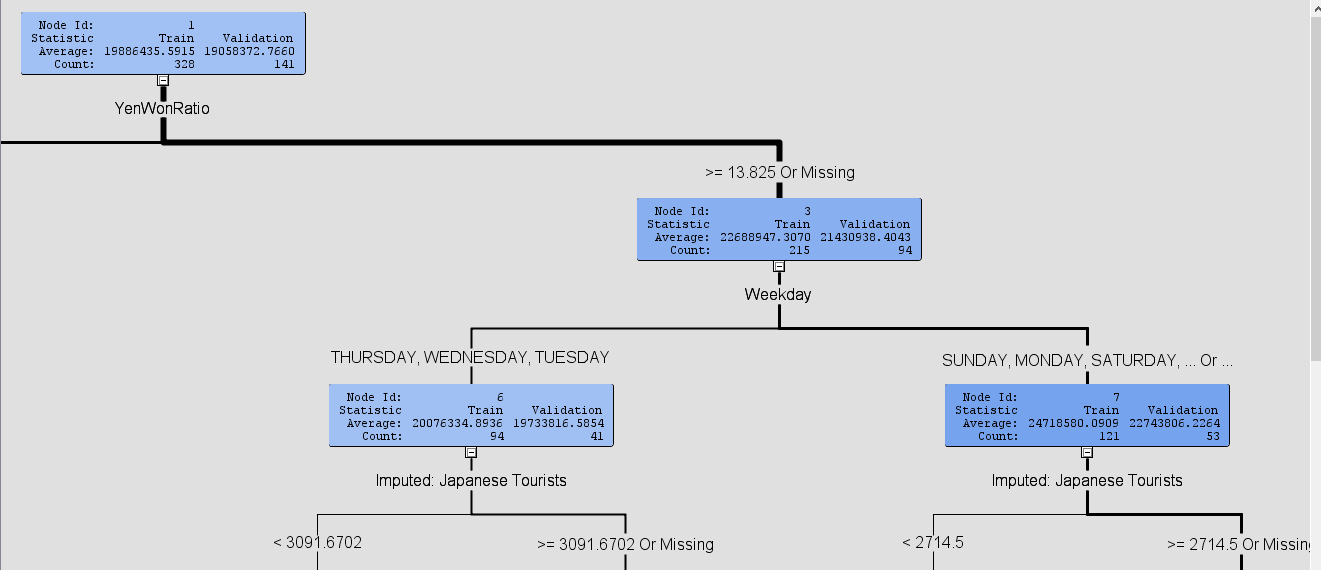
Next we perform data partition, and set the training and validation data sets at 70:30.

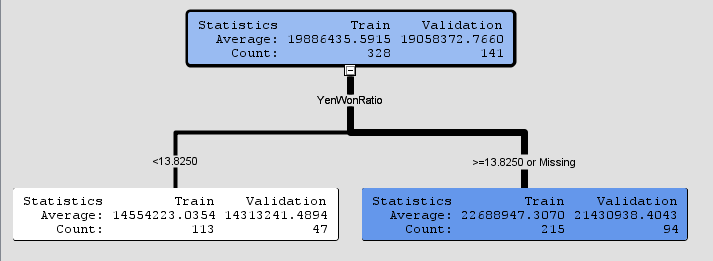


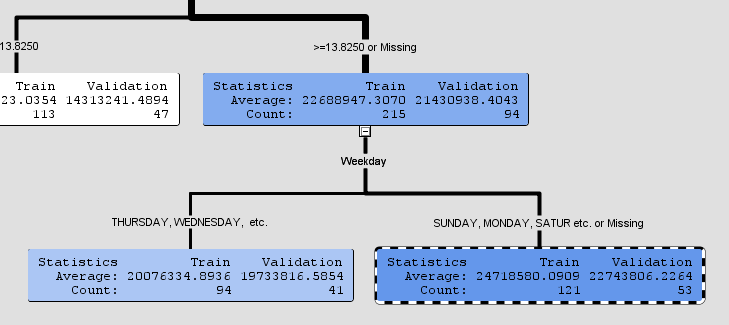
Imputation: Although decision trees can work with missing values we will be imputing the missing values.

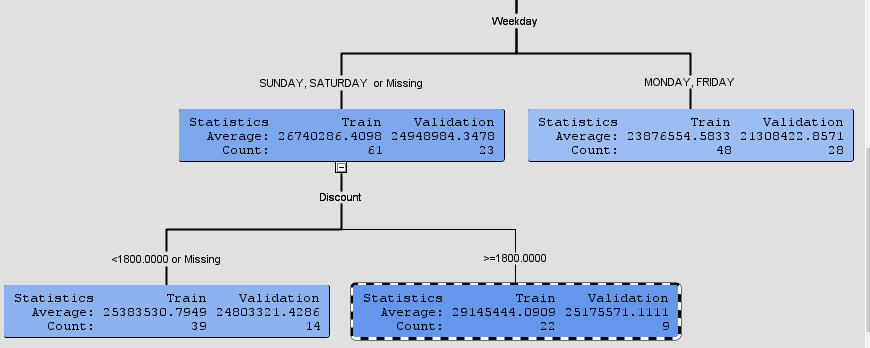


We fit the data through a decision tree now and look at the result.

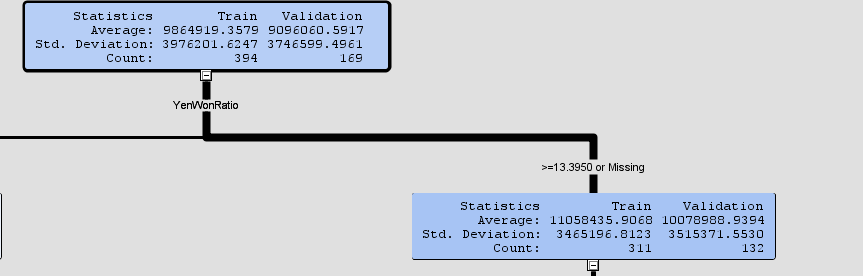


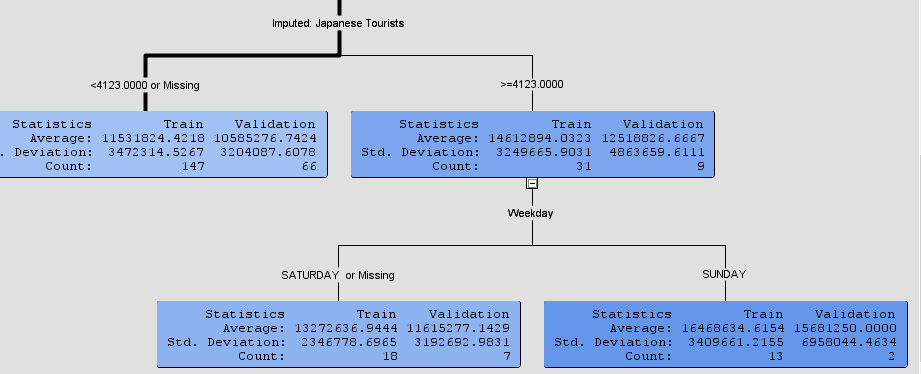
**Store A:** When we use an interactive Tree we can see that the first split is based on currency exchange rate. 

The next split is on Weekdays, 

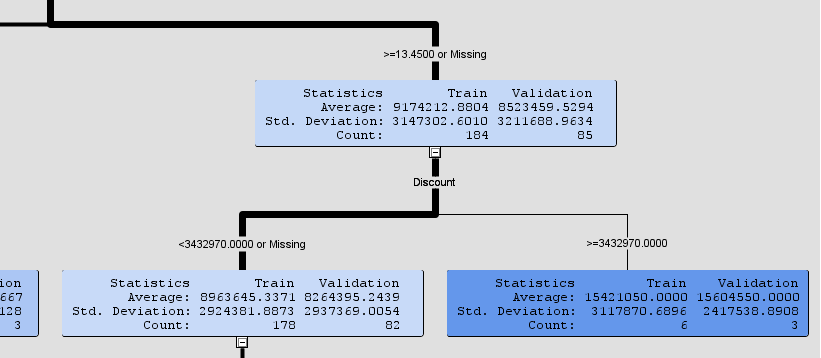
Next two split leads us to the final split on Discount, 

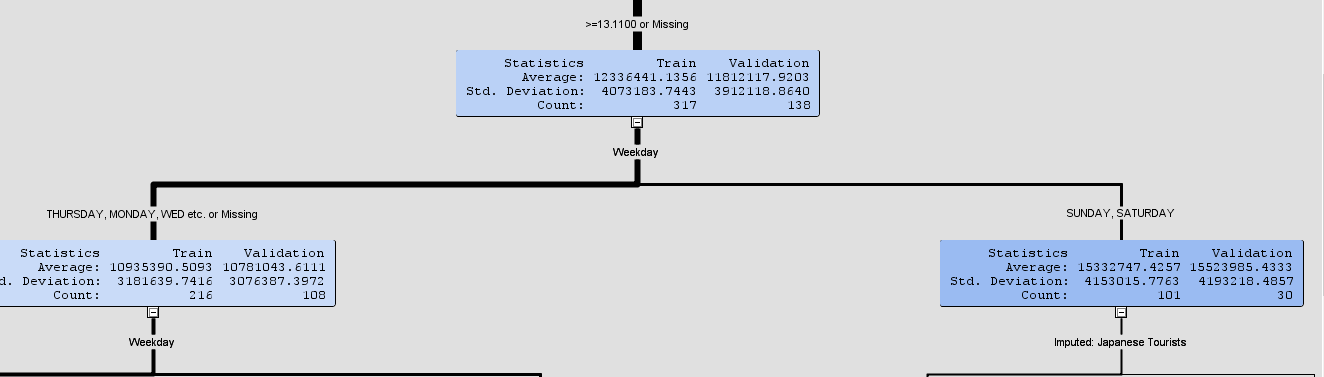
We can say that on days where the YenWonRatio is more than 13.8, on a weekend or where the number of Japanese tourists are more than 2700 we will have the highest total sales for Store A.

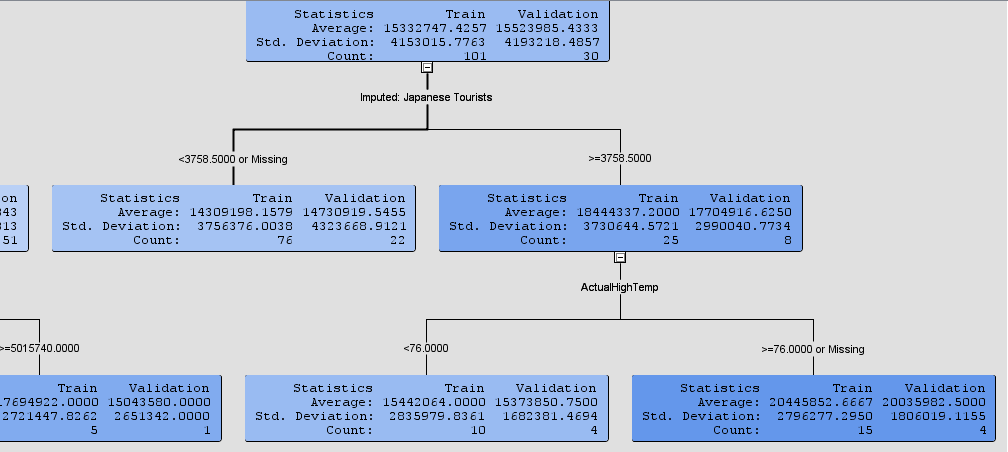
**Store B:** In store B also the first split on the interactive tree happens with the YenWonRatio, 

The next few splits are on weekday and Japanese tourists, 

We can see that the most sales happens on a Sunday where the number of japanese tourists is more than 4123. This implies that japanese tourists really affect the sales of the store B.

**Store C:** For store C the interactive tree shows us that the most sales happen when the discounts given by the store is high. 

Store D: In store D also as expected the first split is on YenWonRatio, and then on the weekday

And then the split is on Japanese tourists feature, and s surprising result is that temperature is also a splitting criteria. This maybe because of people preferring sunny days more for shopping than rainy or cloudy days, 

Store E:

Store E is mainly dependent on discounts to drive sales it seems, there are two splits consecutive for discounts which leads us to the highest sales.

