**Report**

**BigMart Products Outlet Sales Analysis and Prediction**

# Data loading

In that part we load train and test data and we checked types of data and null values and some statistic information like mean, min and max ….

# Exploratory Data Analysis

The most important in that section we visualize our data. Every feature we have in our data. In the main goal to check if we have something wrong like the same attribute but written with different forms as we discover with attribute **low fat** and **LF** in the **Fat content variable.**

Secondly, we checked the problem of balanced data and also, we check the distribution of our data, especially the variable that are of types float (non-categorical).

We detect outliers in target variable and also in other variables (type float), but in this model we didn’t remove any outlier in target variable in contract we remove them in other variables (type float).

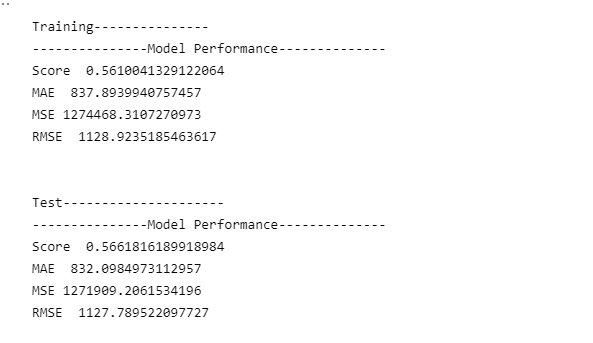
We replace all missing values in categorical variable we most often appear in categorical variable in the mean time we replace other nan variables with **median.**

Finally, we encode the categorical variable with **ONE HOST ENCODER.**

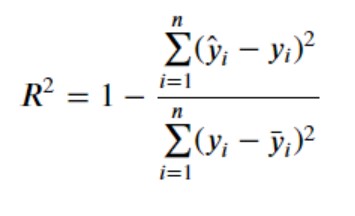
# Modeling

We used a variance of algorithm for this regression problem such as **linear regression model, random forest regression and XGBoost ..**

For linear regression we can see the result as bellow



The result of train data and test data is stable as we can see from the R2 score



The best result was given by the Random Forest Regressor and XGBoot without any transformation on our data.

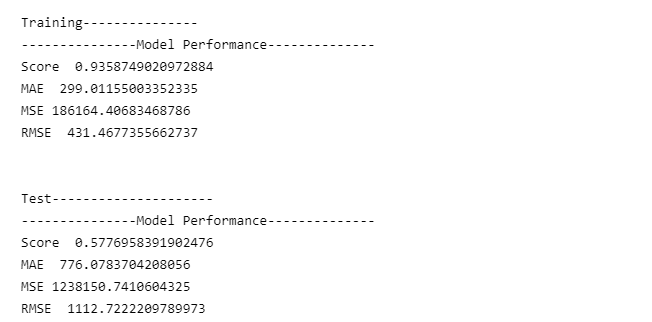


Figure : result of Random forest

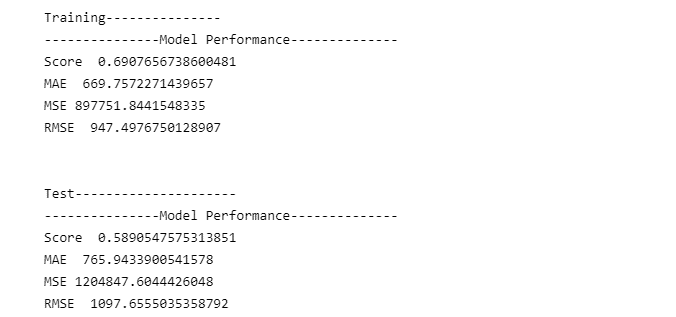


Figure : result of XGboost

The difference between to result that **the random forest** fit data well but in test part fails in the meantime the **xgboost** doesn’t fit well but give best result in testing part which is logic because we don’t have a huge difference between score of training and score of the test.

After that we apply power transform on target variable and other variable of type float.

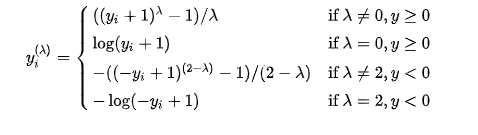


Figure : Yeo–Johnson transformation

You could see the graph of transformation in the jupyter notebook in the section of **transformation.**

The result with random forest with default parameter as bellow



Figure :The result with random forest with default parameters

We fit also the XGBoost with the train data that we got from power transform, which gave us the result bellow.

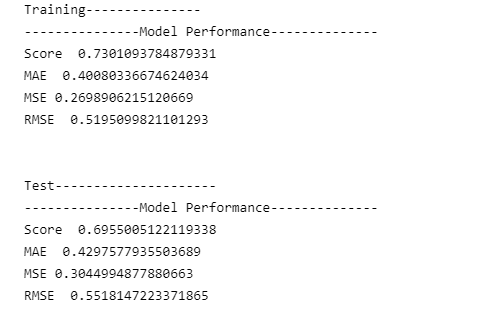


Figure : result of xgboost with power transform data

# Feature engineering

As the final step we choose the XGBoost algorithm to continue with for feature engineering. For that we started with PCA for now and in the next version of my project I will add more transformation for feature engineering.

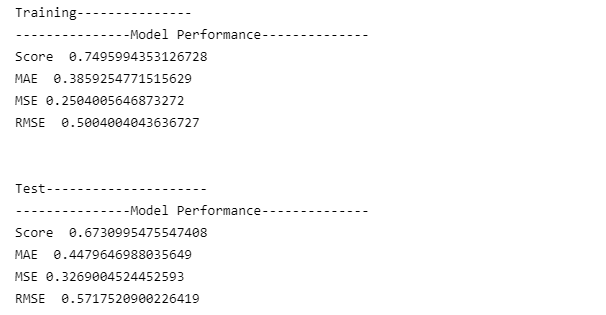


Figure : the result of xgboost after applying PCA

# Conclusion

We don’t look for fine tuning the parameter right for the XGBoost, but we will do it for the next version because takes too much time.

For now we are going to test our unlabeled data with XGBoost model without PCA.