**Machine Learning: Chocolate Bar Ratings**

Laxmi Prakhya

Srikar Mandli

May 2, 2020

**Abstract:**

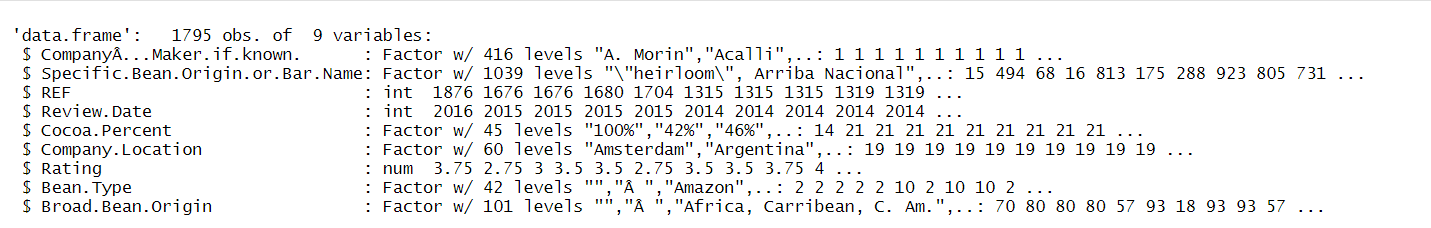
Chocolate is one of the best sweet treats and most popular among all ages and across the globe. Chocolate is prepared from a fruit called *Theobroma cacao* and the quality of this fruit can vary from region to region. Since chocolate is so widely available, in different tastes and bars, have you ever wondered where the cocoa from your chocolate bar came from or if the cocoa in the bar we consume have a good quality bean and taste to it? This is a dataset with 1795 chocolate bar ratings from across the world, where we will be looking at the origin, company maker, percentage of cocoa, bean used and some additional miscellaneous data. Hence, will use the chocolate bar rating dataset to discover which region across the world gives the highest rated chocolate bar. In this data analysis we are analyzing top quality *cocoa beans* and the relationship between *Cocoa Rating with rest of the features using Regression models*.

**Problem Definition and Goals:**

The main objective for this dataset analysis will be to determine the best chocolate bar from its rating and, in order to do so we are utilizing regression model techniques. The different variables from this Dataset are:

1. Company Â (Maker-if known) - These are the names of chocolate manufacturers
2. Specific Bean Origin or Bar Name – The bean’s origin
3. REF – A value which tells how recent the rating on the bar was given.
4. Review Date – The day on which the bar was rated
5. Cocoa Percent – Darkness percentage of cocoa in the bar.
6. Company Location – Geo location of the manufacturer
7. Rating – Experts rating on the bar (ranges 1 – 5)
8. Bean Type – Type of bean used.
9. Broad Bean Origin – The source location of the bean.

The source for dataset is available on ***Kaggle*** and the dataset is called **Chocolate Bar Ratings**.



The ratings of the chocolate bars are between one through five where ‘1’ being least and ‘5’ with highest rating. In this analysis we are trying to analyze the relationship between each variable also, we used all the related variables to measure the error rate of different regression models. The response or outcome variable of this dataset will be the ratings feature, which is a numeric feature, also there are categorical features in this dataset which will be used for error rate calculation.

**Related Work:**

<https://www.kaggle.com/allunia/how-good-does-your-chocolate-taste>

This related work from Kaggle has provided us with an idea on how to explore the data and perform different models for our analysis.

**Data Exploration and Preprocessing:**

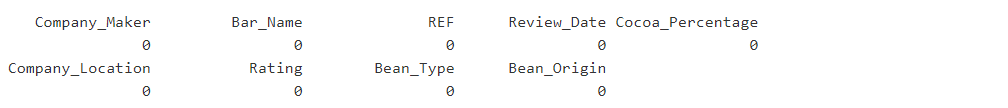
***DATA CLEANING:***

Initially we decided to change the names for columns to something short, since the column names of couple features are too long, and it would be easier for us to work on a comfortable short name.

Hence, we renamed the variables as listed below for our convention.

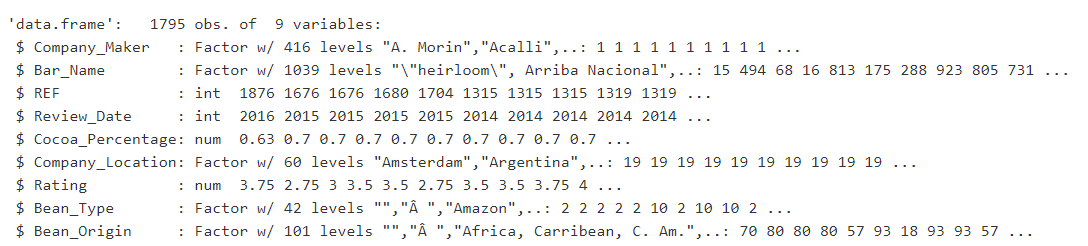


There are no NA values in this dataset.



Since, cocoa percentage is one of the main predictors we have a symbol attached to its suffix which makes it hard for our data exploration and analysis, hence, we decided to purge the percentage symbol from the dataset and convert it into a numeric percent by dividing the values of coulmn with 100.

***Dataframe after Data Cleaning:***

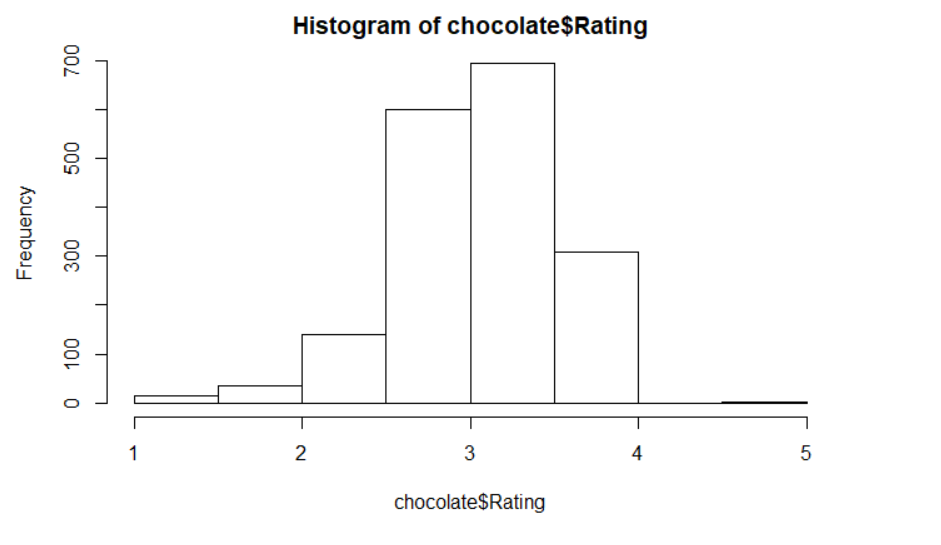


**Exploration:**

In order to understand which features in Chocolate Bar rating dataset has the most correlation with our target variable we have performed correlation tests on each variable with response variable(Rating).

The rating variable has been converted to its logarithmic value for better processing.

Initially we check to see how our data has been scattered by plotting a histogram.



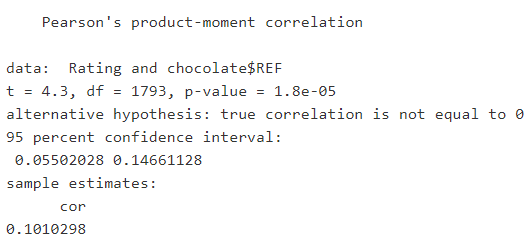
Above histogram is a frequency vs rating graph that shows the frequency of a rated bar.

Also, we noticed the mean of dataset is 3, and most part of this data we see lies after the mean. Hence, we concluded that the data is right-skewed (also known as "positively skewed" distribution) since, most data falls to the right, or positive side, of the graph's peak

Next, we performed the correlation tests for numeric features and recorded the significance value (p- value) from the tests.

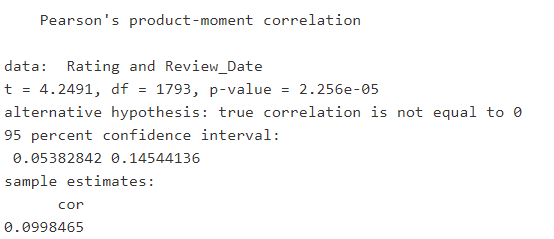
**Correlation test between outcome variable *Rating and REF***:

Since the P-Value on this feature is less than 0.01 i.e. 1.8 e-05, we considered that there is a correlation between the outcome and predictor variable.



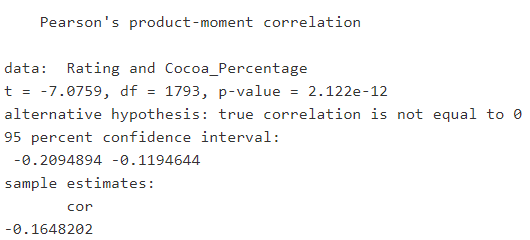
**Correlation test between outcome variable *Rating and Review\_Date***:

Since the P-Value on this feature is less than 0.05 i.e. 2.256 e-05, we considered that there is a correlation between the outcome and predictor variable.



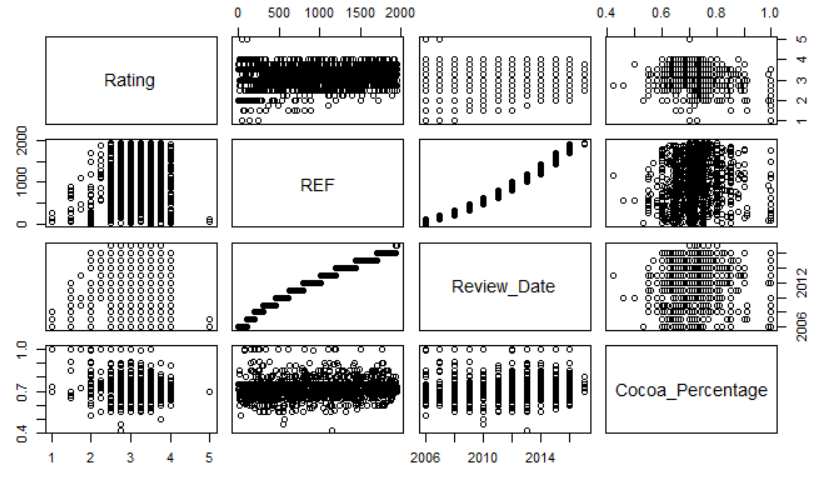
**Correlation test between outcome variable *Rating and Cocoa\_Percentage*:**

Since the P-Value on this feature is less than 0.05 i.e. 2.122 e-12, we considered that there is a correlation between the outcome and predictor variable however, this correlation is not positive.



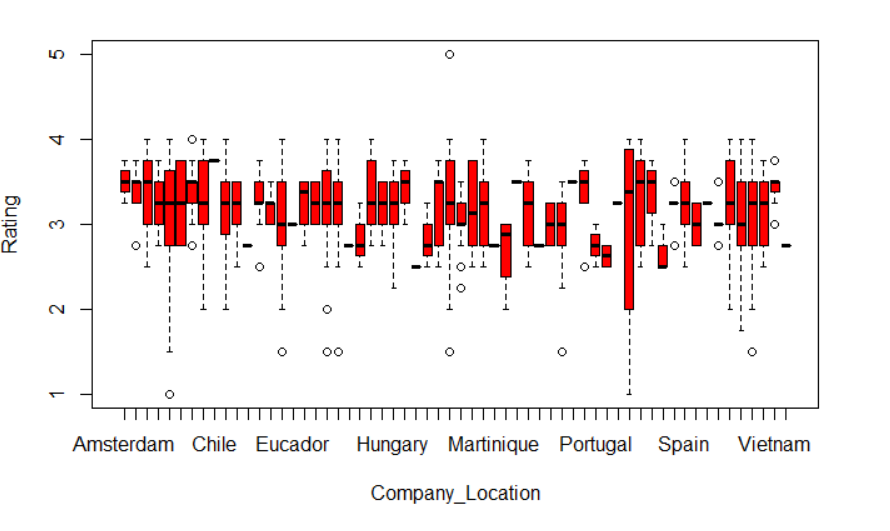
Since we have a correlation between the predictors and outcome variable, our machine learning techniques will show good effect on this dataset with the numeric predictors.

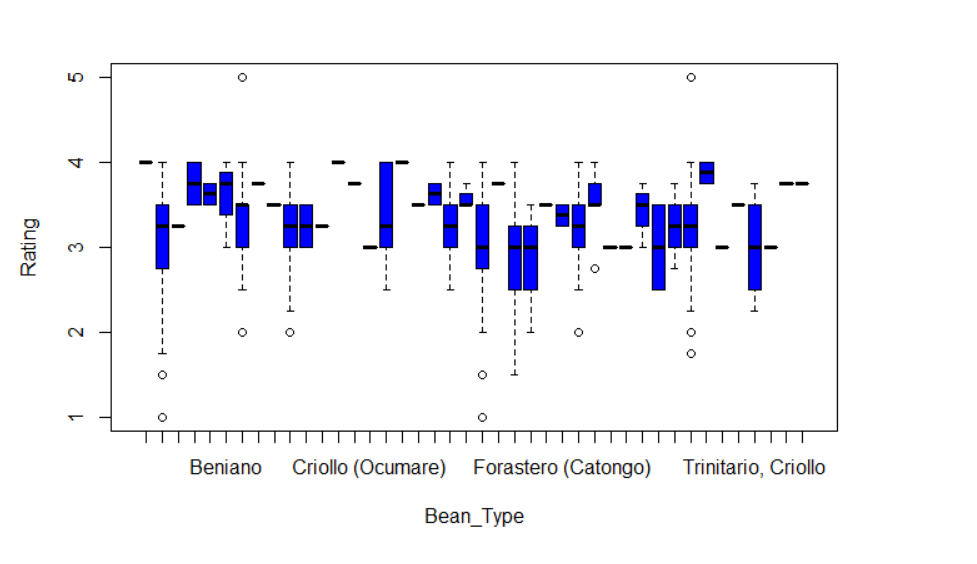
A simple correlation graph can show us a better view on how these are related for each variable combination.



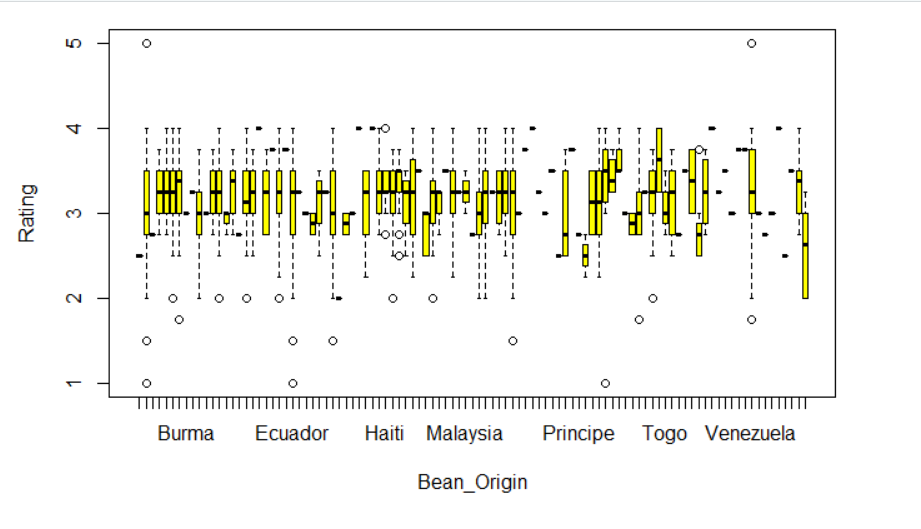
Since, we also have five categorical features in our data, we would like to know if these features have any correlation with rating feature. Hence, we plot some side by side box plots to find if there is any correlation between them. As there is significant variation between the means of response variable and other factor variables they are correlated.

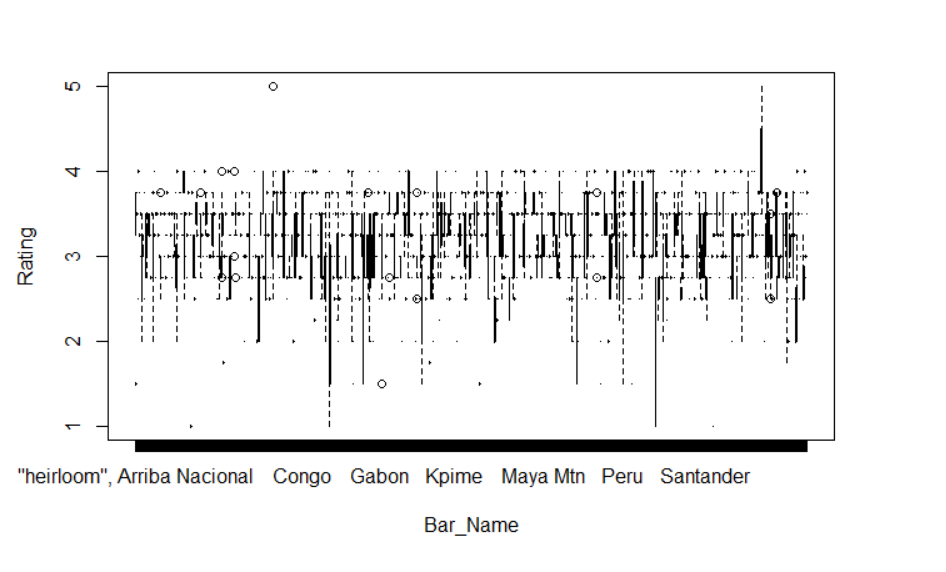
**The distribution of the following box plot is for the variables Company\_Location vs Rating.**



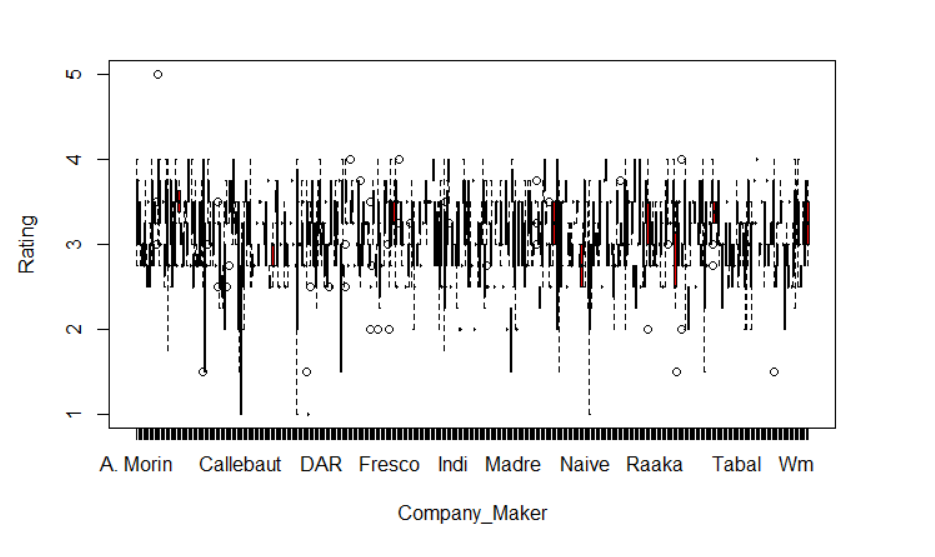
**The distribution of the following box plot is for the variables Bean\_Type vs Rating**.

**The distribution of the following box plot is for the variables Bean\_Origin vs Rating.**



**The distribution of the following box plot is for the variables Bar\_Name vs Rating**.

**The distribution of the following box plot is for the variables Company\_Maker vs Rating**.



Hence, we deduced that each element in this dataset has a significant relationship with the outcome variable Rating and therefore, we are using all the features in this dataset for our further analysis of the best chocolate bar.

Before we jump into applying the machine learning models

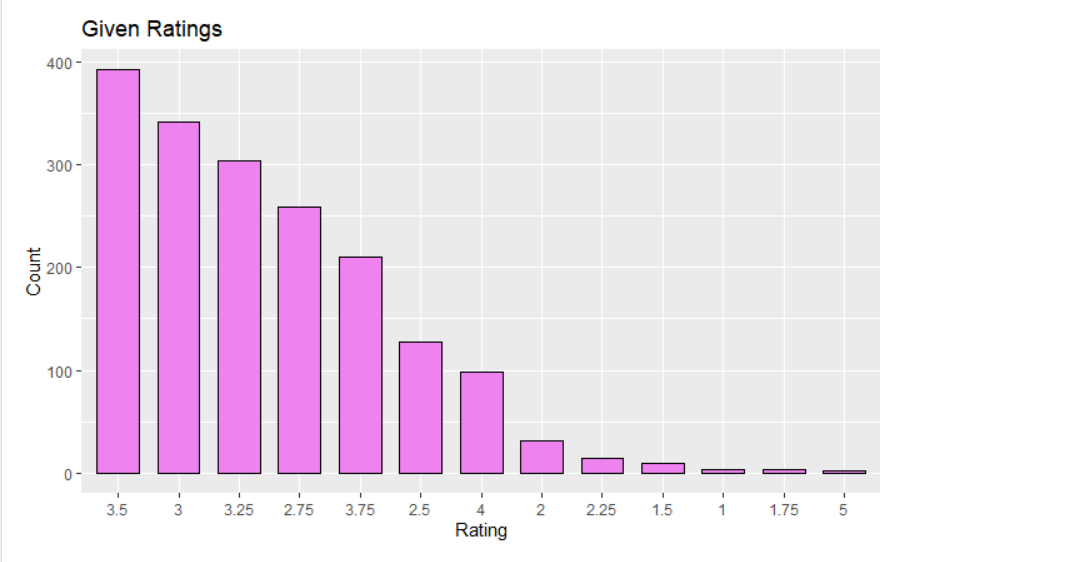
**What is the rating given by the experts?**

Below is a histogram of this dataset’s chocolate bar ratings.

The ratings are between 1 through 5 where:

1. **Unpleasant**
2. **Disappointing**
3. **Satisfactory**
4. **Premium**
5. **Elite**

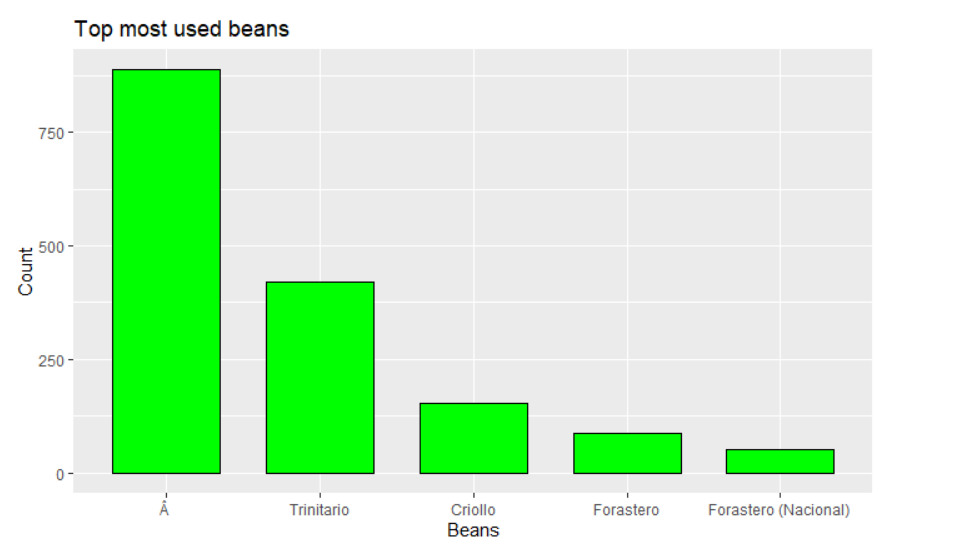
This helps us understand how many of the chocolate bars rated are bad and not suitable for eating.



**What Types of Beans are used?**

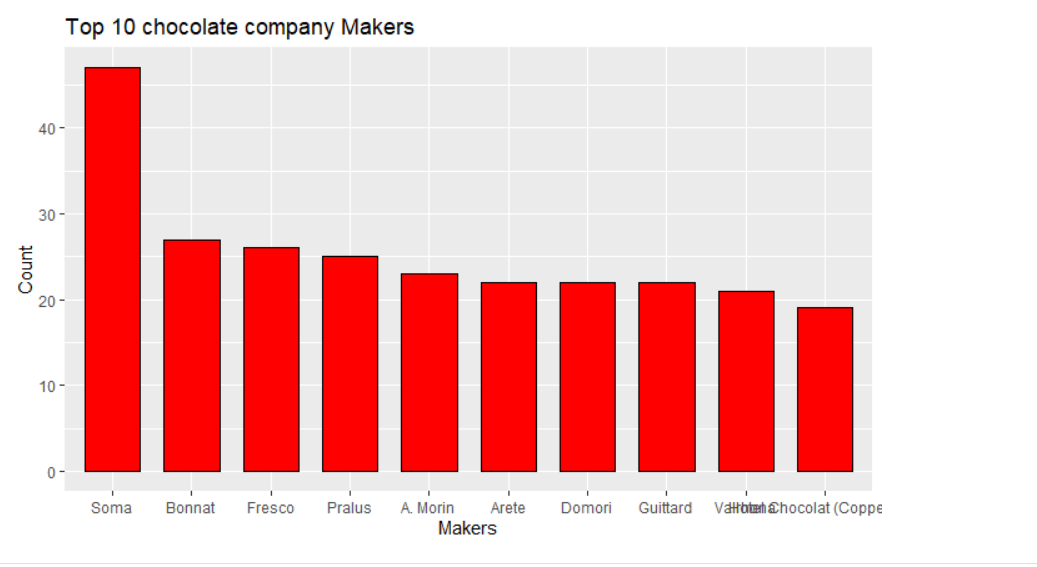
We got five main type of Beans that are being used to make the Chocolate bars across the world.

The below Graph shows us the count of topmost used beans.



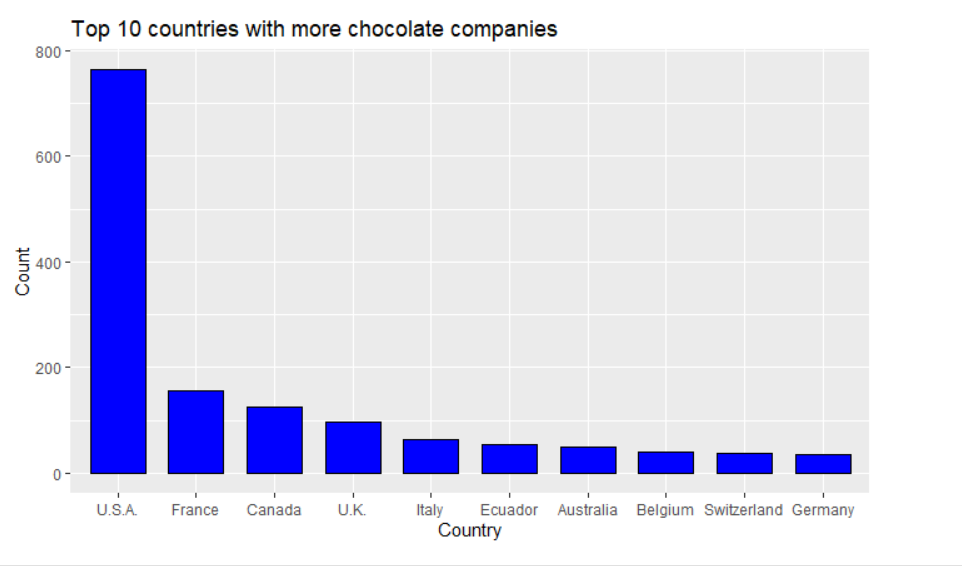
**Who makes the Bar?**

This graph shows us about the manufacturer of the chocolate bar and it includes only the top 10 makers across the world.



**Origin of the Bar?**

Precise georaphical locations of the country where the bar has originated.



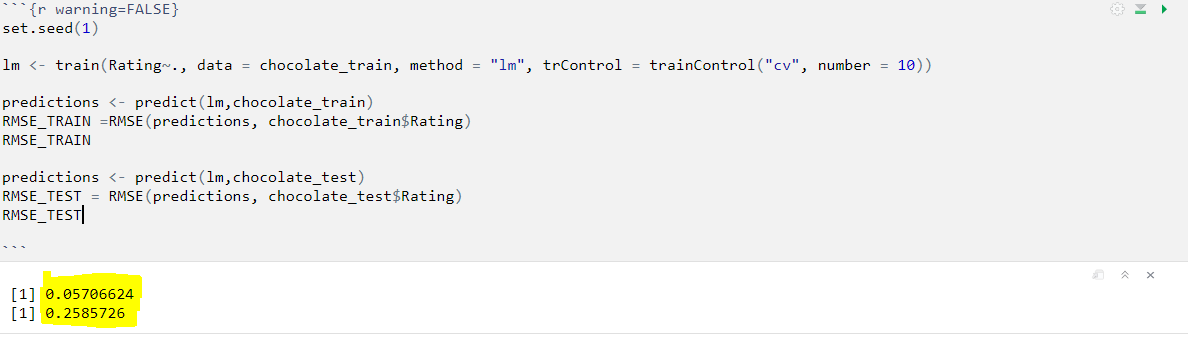
**Data Analysis:**

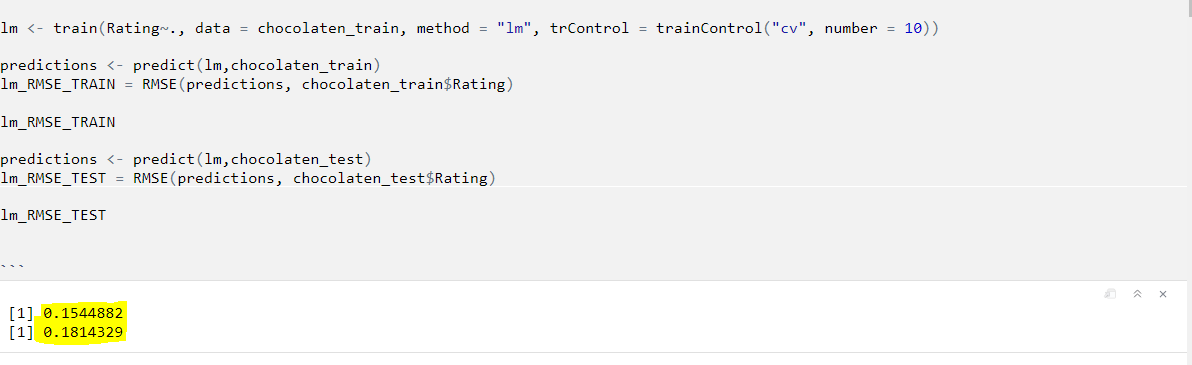
As my Response or Target(rating) variable is numeric, we used different regression models like "Multiple Linear Regression", "Bagging Tree", "Regression Tree”, “Model Tree", "Stepwise (Forward) Linear Regression”, “Stepwise (Backward) Linear Regression", “Regularized Linear Regression (Lasso, Ridge, ENET)”,, "Random Forest", "Gradient Boosted Trees", "Artificial Neural Networks". Our model has 9 variables with 1795 observations, we split the response variable into train and test with a division of 80%(train) and 20%(test) using *createDataPartition* function which was an inbuilt function in Caret package.

Later we tried to implement Multiple Linear Regression model on different pre-processed data, firstly we took the RAW data and used the model which gave us the higher error rate (0.8636943). Secondly, we tried to  convert the cocoa percent to numeric and used the model which then returned a lower value (0.2585726). Thirdly, we tried scaling all the factor variable and the model did much better with error rate (0.1692946). We decided to implement all the remaining models by using scaled data, the reasons being.

It has a lower error rate and the error rate on train and test data is almost the same, in other words it was not *Overfitting* when compared to other procedures. Hence, each model was built using *10- fold cross-validation* and had its predictions measured against the test data.

Snip 1 is data without scaling, and we observe overfitting by a greater margin when compared to the Snip 2 which has scaled data.

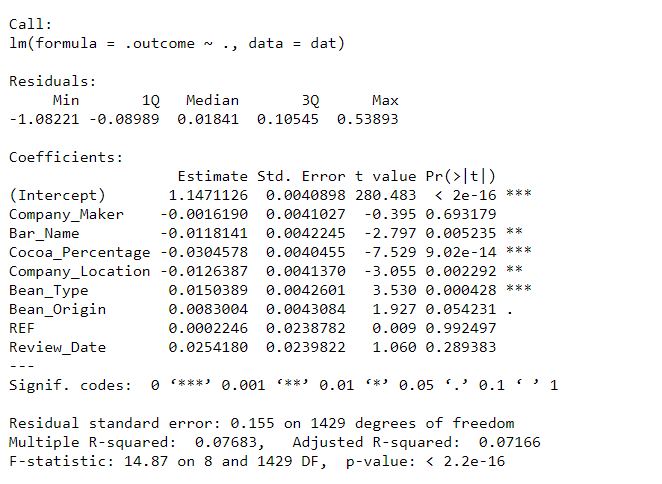


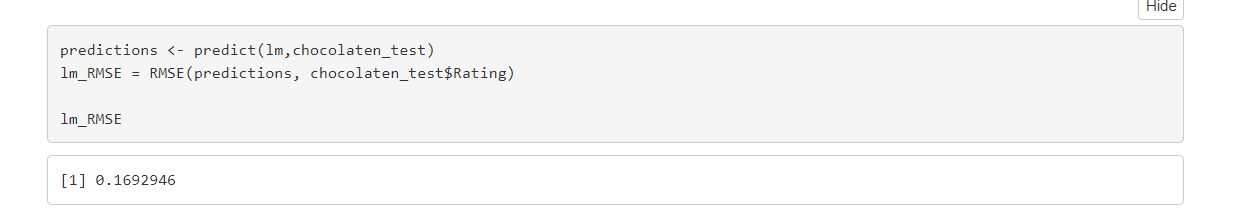


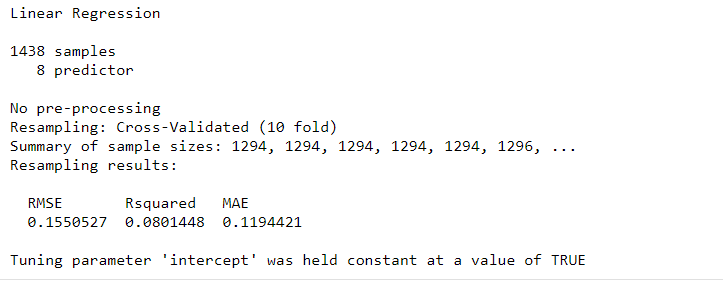
Hence, we would like to produce algorithms in out next step which will show us how RMSE is tuned and produced.

**Multiple Linear Regressing**

From the summary of Multiple Linear regression, we see that cocoa percentage, company location and Bean type are strongly correlated, whereas bar name is somewhat related. As the probability is less than 2.2e-16, indicating that our models are significant, Machine learning practitioners are mostly interested in prediction accuracy on the test (out of sample) data, the most popular measures of evaluating the prediction accuracy of a regression model are : RMSE(Root Mean Squared Error) and MAE (Mean Absolute Error). Thus, lower RMSE results in better prediction accuracy of the model.



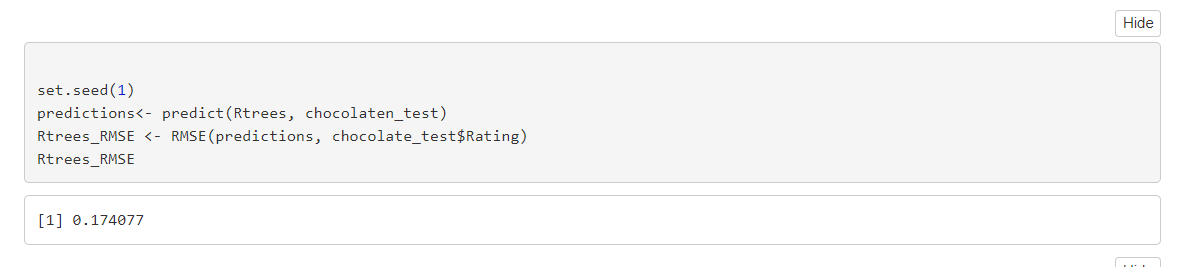


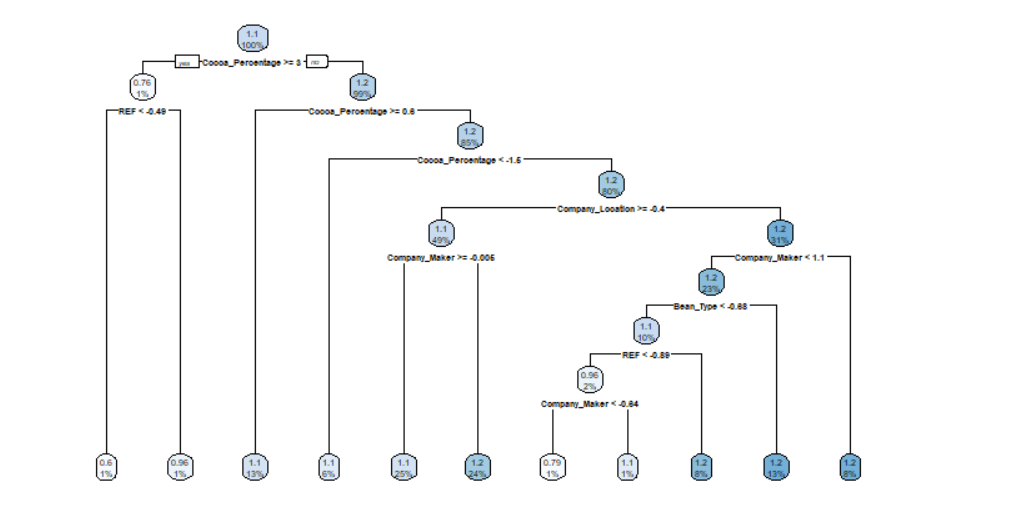


**Trees (Regression, Model, Bagging)**

**Regression Trees:**

When we compare to linear regression models tree models dint perform well, i.e. in multiple linear regression we see that RMSE is lower when compared to the Regression tress. Regression tree provides a full mutually exclusive partition of the predictor space into regions with boundaries that are parallel to the predictors’ axes, due to the form of the tests.  Regression trees can be conveniently visualized using the *rpart.plot* package, by using rpart.plot(Rtrees$finalModel, roundint = FALSE).



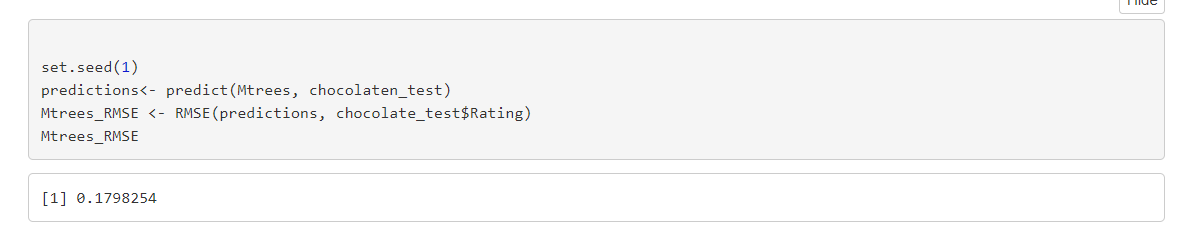


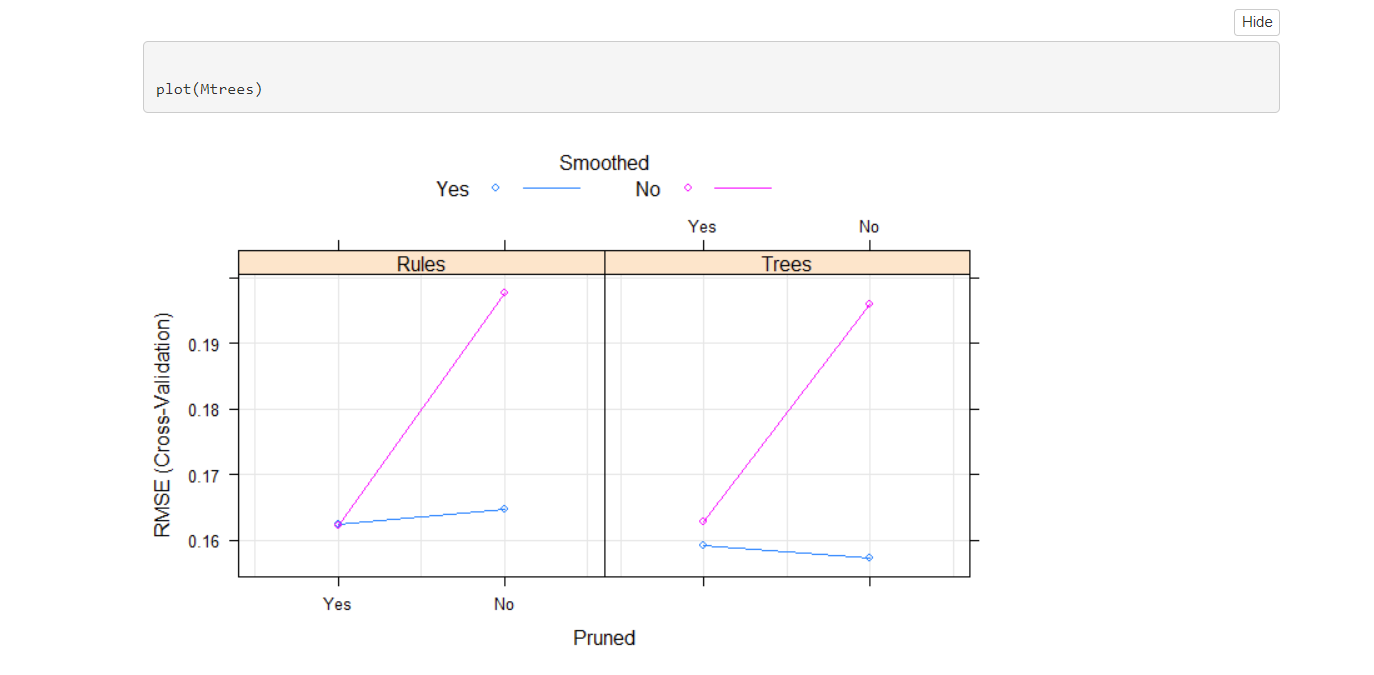
**Model Trees:**

Model tress on the other hand gave a similar performance like Regression trees.

Model trees combine linear models and decision trees to create a hybrid model that produces better predictions and leads to better insights than either model alone.

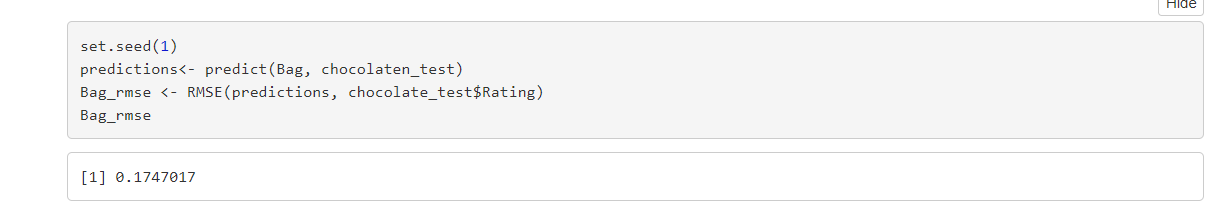
Models used in the tress increase the computational complexity and should have perform better than the regression tress, but it almost got the similar RMSE’s.





**Bagging Trees:**

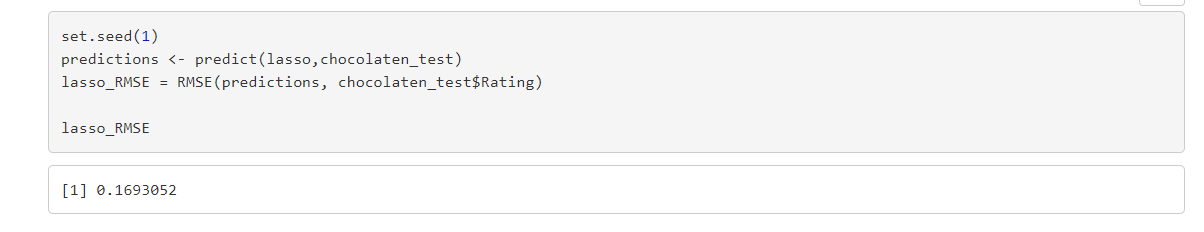
Bagging tree has got a similar performance just like Model and Regression trees, the RMSE lies in between Model and Regression Trees.

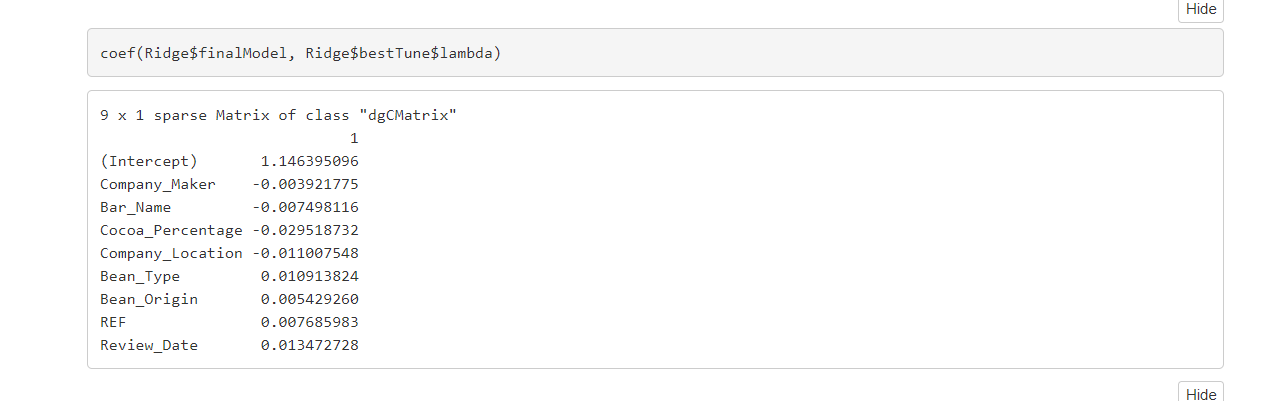


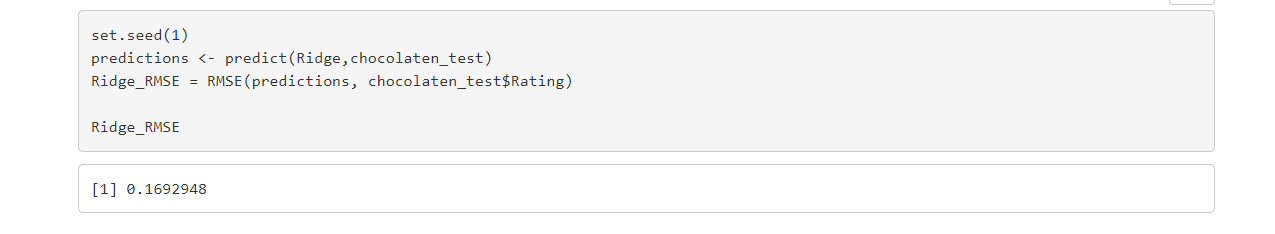
**Regularized Liner Regression:**

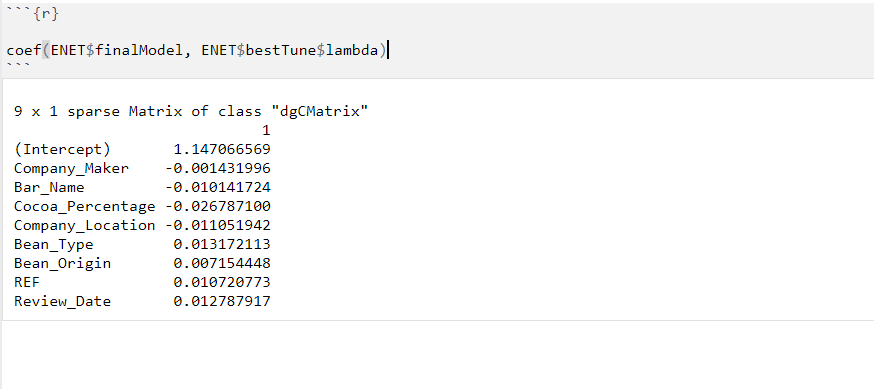
Regularization is a technique which attempts to adjust the model complexity to strike a balance between overfitting and under fitting. We used three types of regularization L1 regularization (also called **Lasso**), L2 regularization (also called Ridge), and Elastic Net where we can tune both alpha and Lambda. When compared to model trees all three models (L1,L2, ENET) performed like one other. Likewise, we see which coefficients have been used. The one with dots are zeros which are not being used.

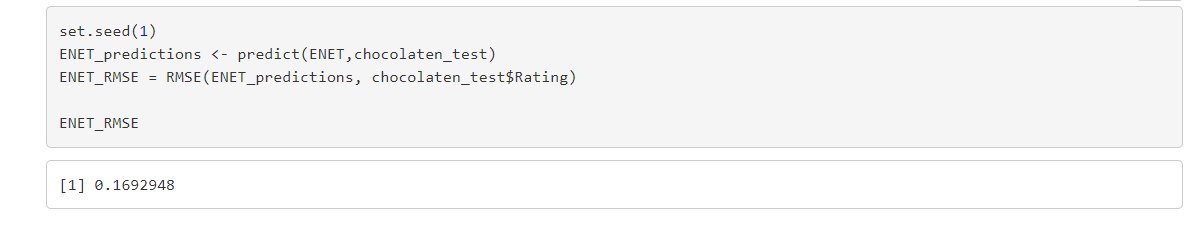






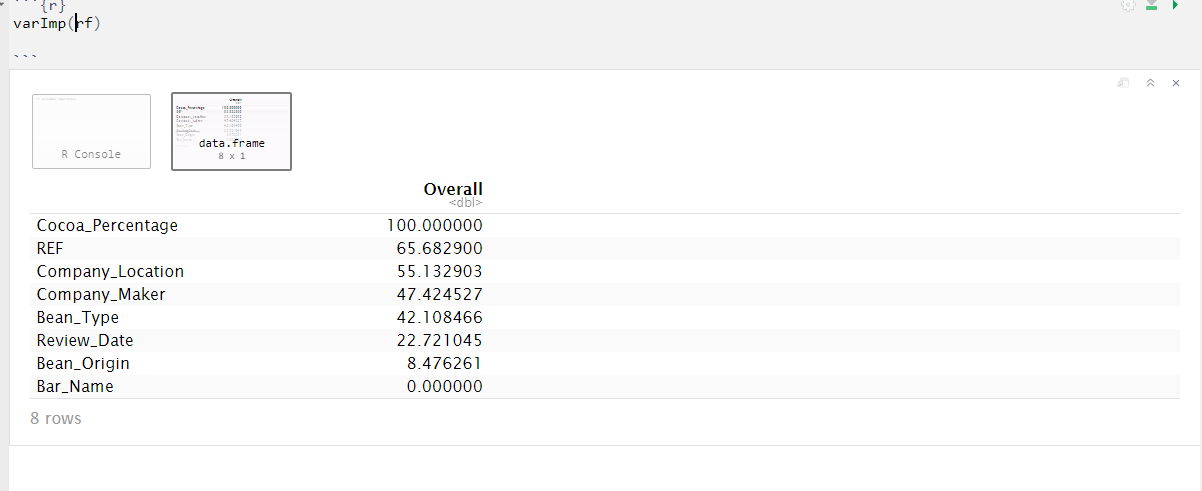






**Ensembling Methods:**

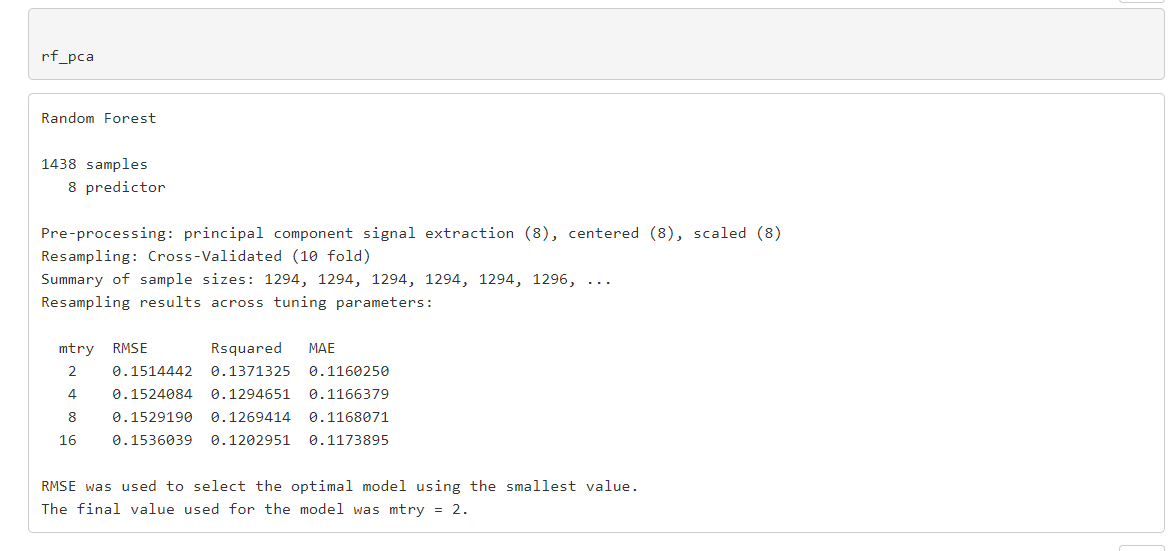
All the ensemble methods are based on the idea that by combining multiple weakerlearners, a stronger learner is created. Ensembling Methods used are Random Forests and Gradient Boost. The greatest advantage of Ensembling is it improves performance on massive or miniscule datasets, likewise it has produced best results on Random forest compared to Gradient Boost. *Varimp* function in Random Forest gives the variable importance as shown in below.

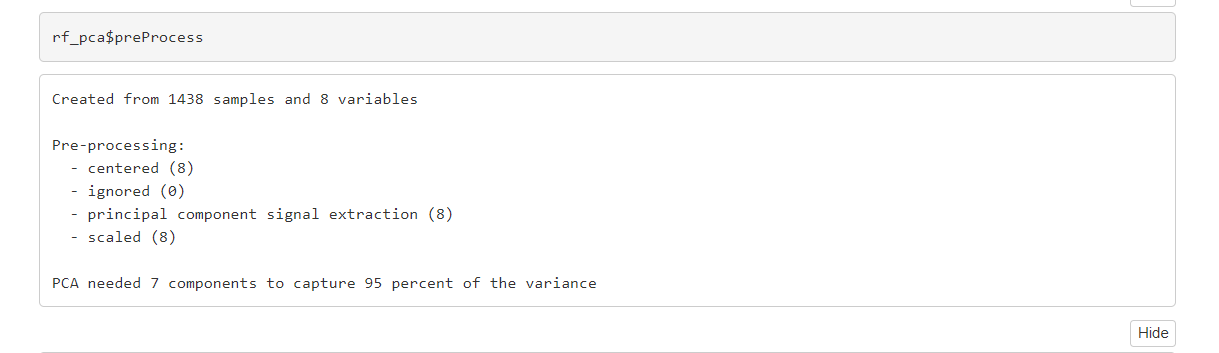




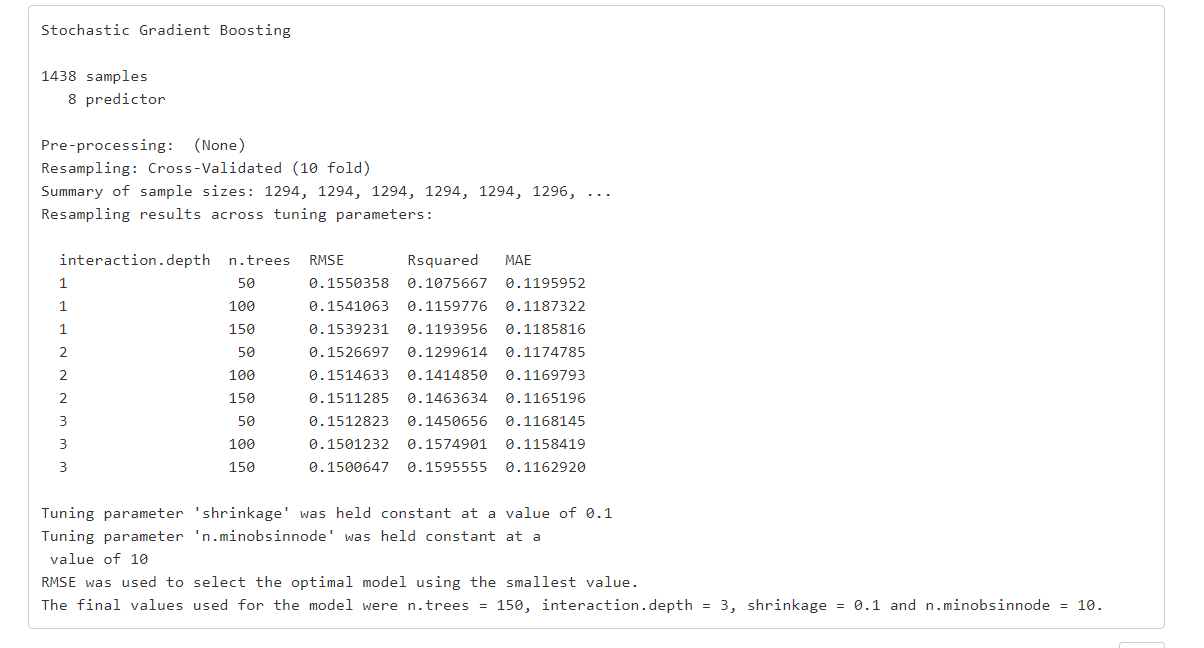
The random forest models built using principal components found 2 predictors to be the optimal amount.

Seven principal components were needed to explain 95% of the variance.



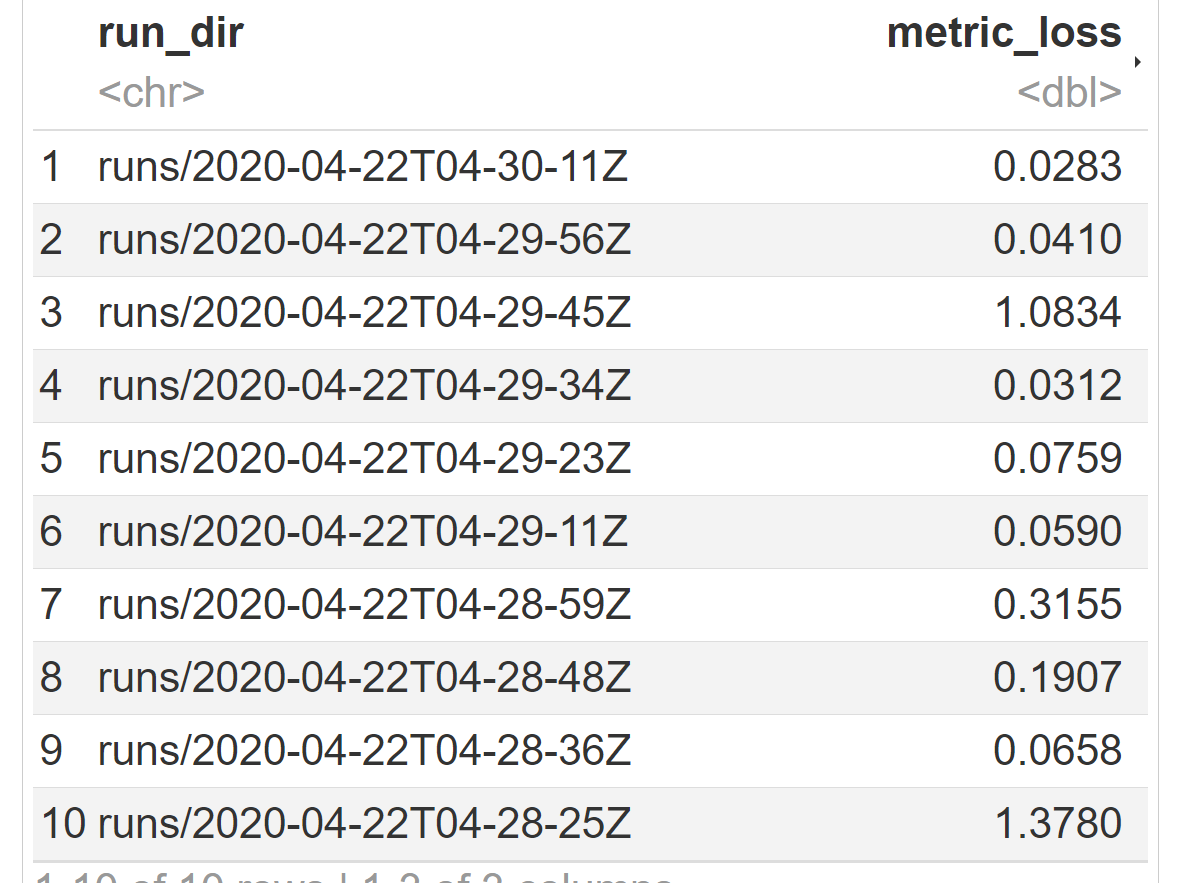


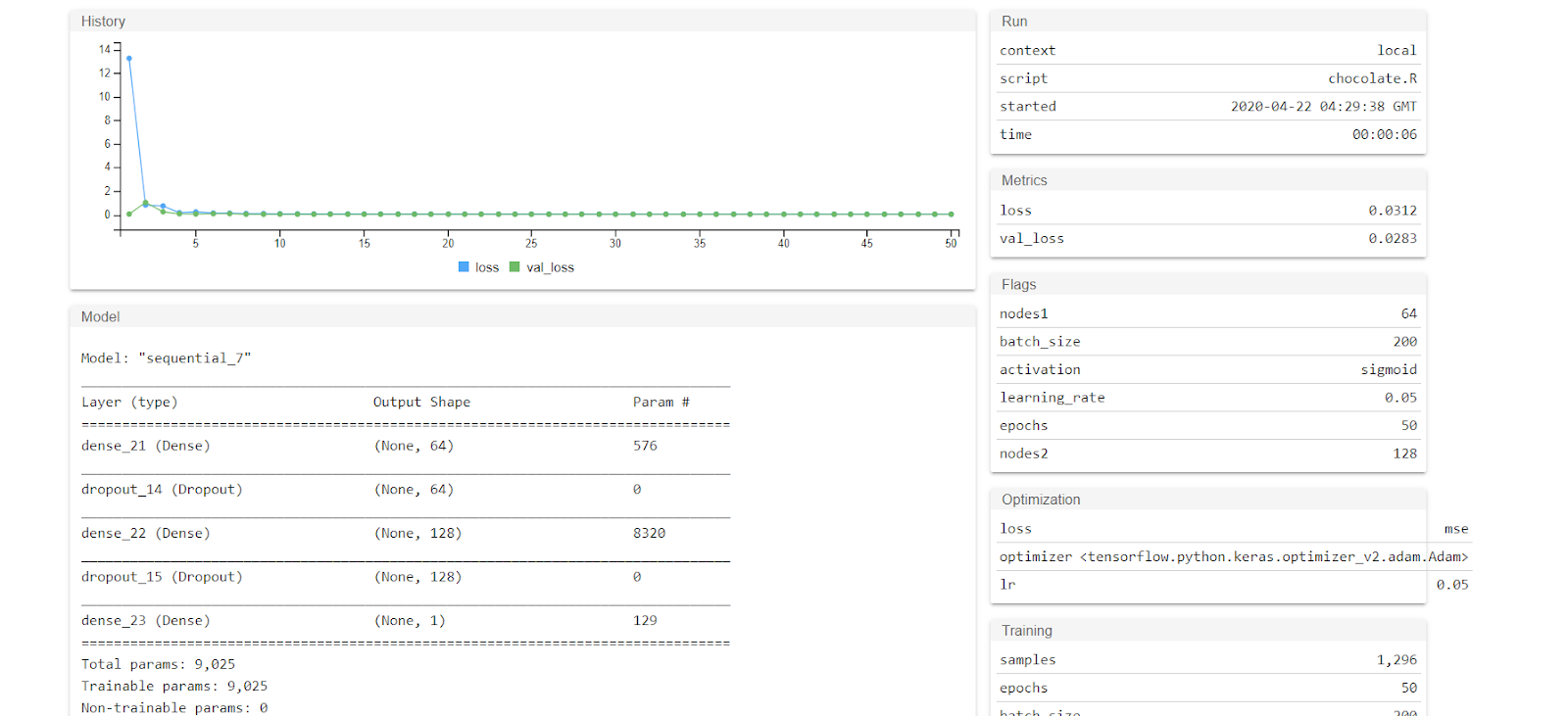
In Gradient Boost we used three iterations and the number of tress used are 150 to get the best RMSE.

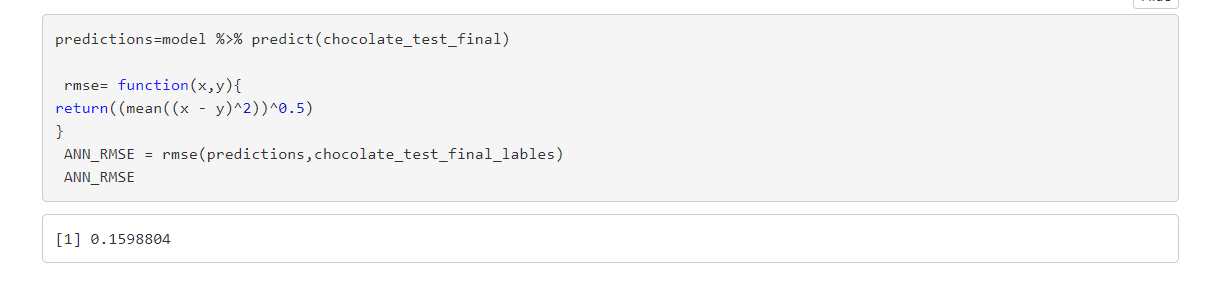


**Neural Networks:**

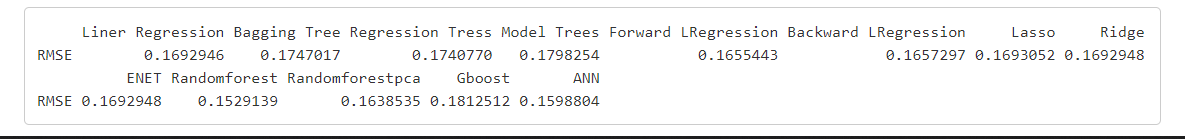
An Artificial Neural network, ANN, consists of simple processing units, the neurons, and directed, weighted connections between them. Firstly, we have partitioned the train data into train and validation sets as 90:10, later we have scaled down all the variables leaving out my response variable after which we performed Random Hyper-Parameter tuning using “tfruns” package, We then have Three hidden layers which include the output layer as well, used two flags one for each layer with different combination of nodes, batch size, activation function, learning rate and epochs, similarly added a drop out layer for those layers with a values of 0.5. Later implemented the best combinations on training data (train + validation) to train a model. Considered the best metric loss (0.02)







Finally created a table of the out of sample RMSE Predictions.



**Conclusion:**

Initially when we were implementing a model it showed terrible overfitting, then we tried to use different combinations to lower error rate. It was evident that the response variable has a correlation with other variables by seeing the P-value and graphs like scatter plot, histogram, boxplot of all the data. Despite of Cleaning, implementing different pre-process methods we see that the models are still overfitting and observed a lesser R-squared value. We might overcome this by adding more variables to the problem and also try different other combination where we can try to get a better performance of the model, there is room to improvement and we will definitely look forward for more refining the data and models. However, we see ANN and Random Forest which performed well with less error rate. We are comfortable in saying that the models did well with RMSE < 0.18 and my goal has been reached.

**References:**

<https://www.kaggle.com/allunia/how-good-does-your-chocolate-taste>

<https://www.kaggle.com/tibhar940/chocolate-bar-ratings-python-eda-dataviz>