**RANDOM FOREST**

Hello everyone, we are very much aware of current technologies thriving in the world and how to use it. So while I was trying to understand Machine Learning algorithms, Random Forest is the one that fascinated me a lot. Before beginning:

**“What is Random Forest?”**

**“Why is it used?”**

**“How does it work?”**

Have you ever thought when you do something, it can be anything as in decorating the house or throwing a house party, you consider a lot of things that needs to be taken care of?

Take an example of house-party, the factors that are important when to have a party, the place where it will be, and how many people needs to be invite, items to serve and what not. Diversity is the most important factor to be considered because not every person invited will be the same or have similar food habits.

**What is Random Forest?**

In Machine Learning, the techniques are classified into two:

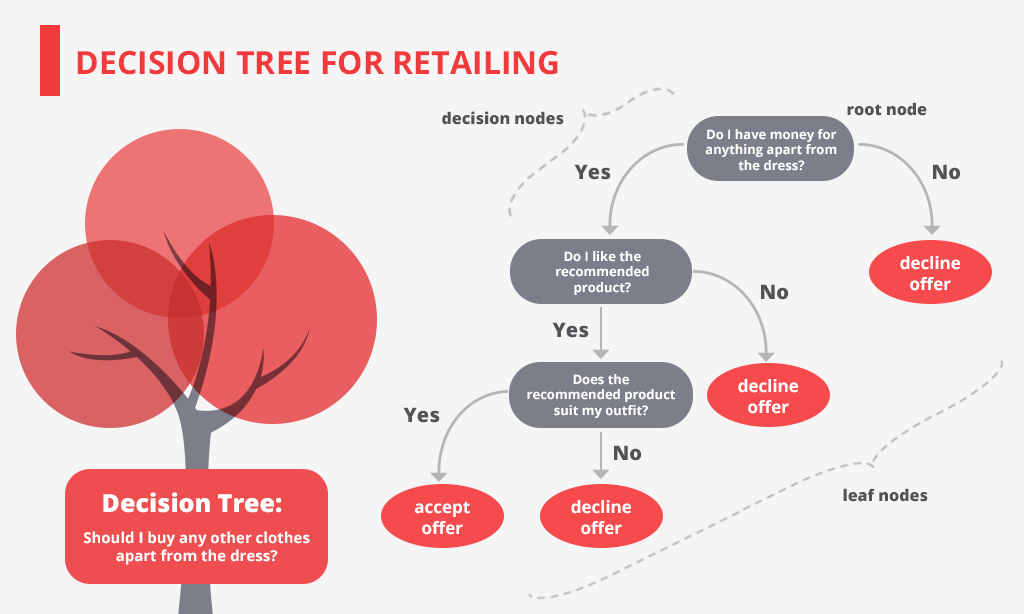
**Supervised Learning**

**Unsupervised Learning**

Random Forest algorithm falls under the Supervised Learning Algorithm. The foremost advantage is that Random Forest is used for both Classification and Regression. Right now, we are interested in how does it work in Classification? By name, we can figure out it is about forest that means bunch of trees and random means without any pattern or reason, just by chance.

Therefore, to know how Random Forest works, one should know what Decision tree is.

A **decision tree** graphically represents all possible solutions to a **decision** based on certain conditions. It is said to be a tree as it starts from a root and later splits into branches. Decision tree is said to be the building-block of Random Forest, can be understood as a series of questions (Yes/No) asked about the data leads to the prediction. Decision Tree is the interpretable model that does the classification as we humans do. For Ex:-



Random Forest is an Ensemble Learning Method, made up of many such decision trees.

**How does it work?**

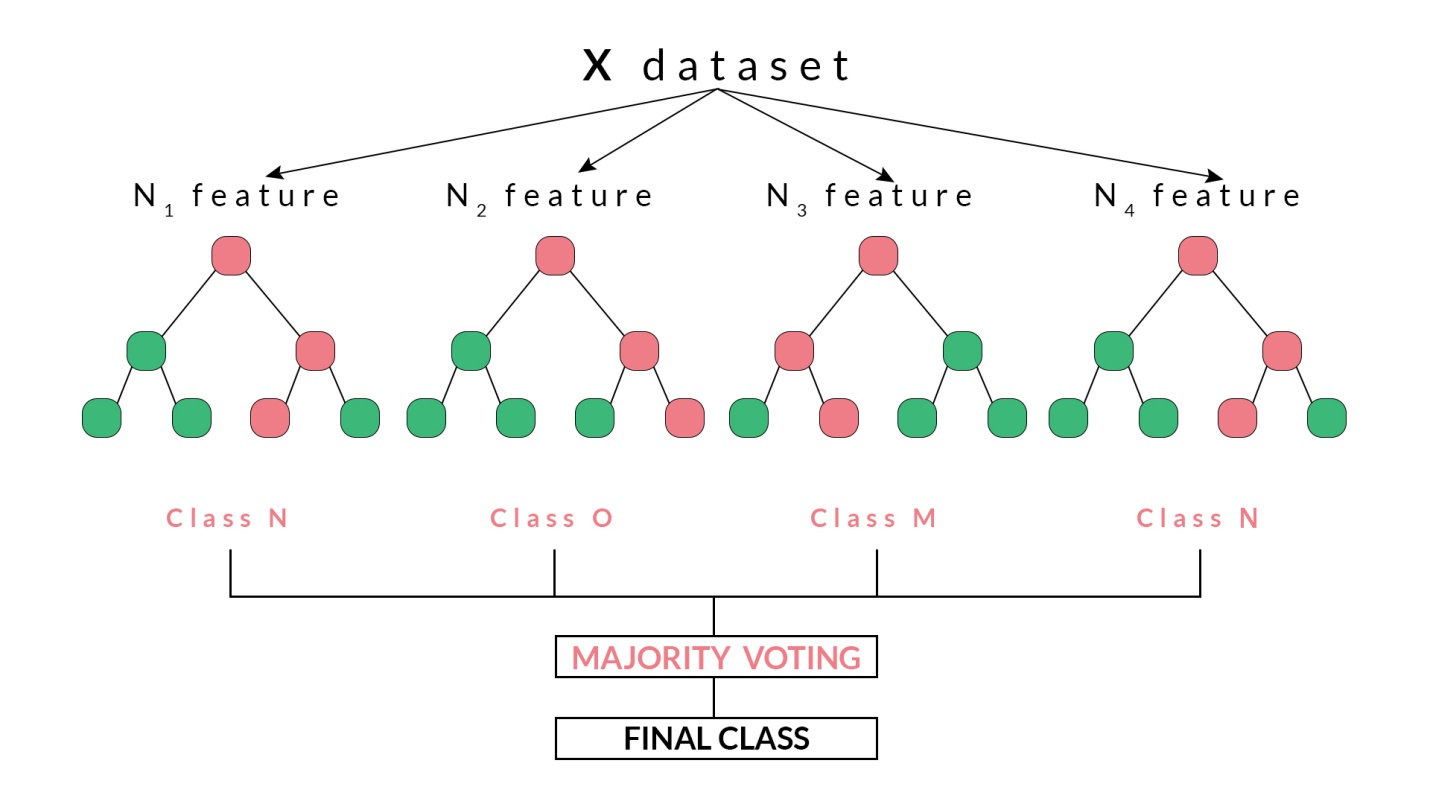
This model uses 2 key concepts that named it as Random, rather than just averaging the prediction of trees or majority of votes. They are:

* Random sampling of training data points while building trees
* Random subsets of features considered while splitting nodes

Describing the former one, when in training, each tree in the Random Forest will learn from a random sample of all the data points. The samples are drawn with replacement (called as bootstrapping), which leads to the usage of some samples multiple times in a single tree. By training all trees with different samples, even though every tree might have high variance with respect to particular set of training data, overall it will have lower variance but not at cost of increasing bias.

When in test, the final decision is based on the average of the predictions made by each trees. This method of training each decision tree by different bootstrapped subset of samples, and then predicting the results by averaging the predictions made by all trees is known as bagging (short term for bootstrap aggregating).

The latter important concept of Random Forest is, for splitting each node only a subset of all features are considered in each decision tree. For classification meaning, this model is set to “sqrt(n\_features)”, which indicates that if there are 16 features in at each node in a single tree, only 4 will be considered for splitting the node.



*“Random forest combines hundreds/ thousands of decision trees, training each of them on a slightly different set of datapoints, considering a limited number of the features by splitting nodes in each tree. The final predictions are made by averaging the predictions of each individual tree.”*

To understand how Random Forest model is better than Decision Tree model.

**Scenarios where Random Forest worked pretty well**

Let’s imagine the following scenario:

It is you who will take the decision, whether Tesla stock will go up & you have access to a dozen of analysts who do not have any prior knowledge about this company. Each analyst has low bias because they did not come in with any assumptions, & is allowed to learn only from a dataset of news reports.

It does seem like an ideal situation, but the problem here is that the reports are likely to contain noise in addition to real signals. Because the analysts’ predictions are based completely on the data: 🡪 They do have high flexibility. 🡪 They can be swayed by irrelevant information.

The analysts may come up with different predictions from same dataset. Moreover, each individual analyst has high variance and would come up with drastically different predictions if given a different training set of reports.

The solution of this problem is to not rely on any one individual, but pool/average the votes of each analyst. Furthermore, like in a random forest, allowing access to each analyst to a section of reports only and hoping that the effects of the noisy information will be cancelled out by the sampling. In real life, we rely on multiple sources (never trust a solitary Amazon review), and therefore, not only is a decision tree intuitive, but so is the idea of combining them in a random forest.

## Feature Importance

The feature importance in a random forest implies the sum of the reduction in Gini Impurity (a measure that decision tree tries to minimize) over all the nodes that are split on that feature. We can use these features to figure out what predictor variables the random forest considers most important. These can be extracted from a trained random forest model.

Next comes Optimization, that can be done through random search using the [RandomizedSearchCV in Scikit-Learn.](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html) Optimization (also known as model tuning) indicates finding the best hyper parameters for a model. These hyper parameters will vary between datasets, so it needs to be done separately on each datasets. What can we might optimize in random forest are the no. of decision trees, the maximum depth of each decision tree, the maximum no. of features considered for splitting each node, and the maximum no. of datapoints needed in a leaf node.

**CONCLUSION:**

Rather than building machine Learning models in Python without having any understanding of them, I say it’s better to have some prior Knowledge, about what is going on behind the screen. In this article, we not only created and used the model but did have some understanding on the basics of Random Forest.

The key concepts to be learned from this model are:

* Decision tree
* Gini Impurity (Represents the probability that a randomly selected sample from a node will be incorrectly classified according to the distribution of samples in the node.)
* Bootstrapping
* Bagging
* Random subsets of features
* Random Forests
* The larger the number of trees, more accurate is the model

Hopefully this article might have given you the confidence and understanding required to start using random forest on your projects. The more we know about a model, the better equipped we will be to use it effectively. Let’s have a look at an example through an audit dataset

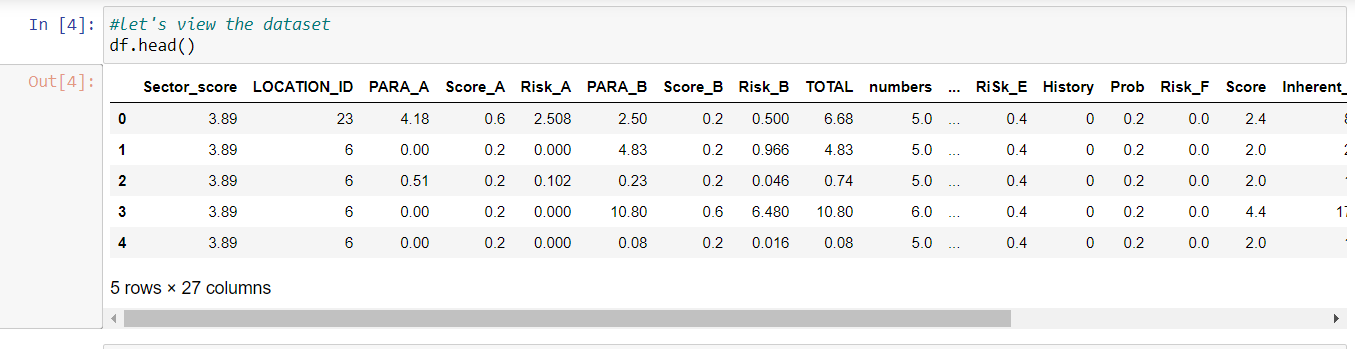
## Business Requirement

## The goal of the dataset is to help the auditors by building a classification model that can predict the fraudulent firm on the basis the present and historical risk factors.

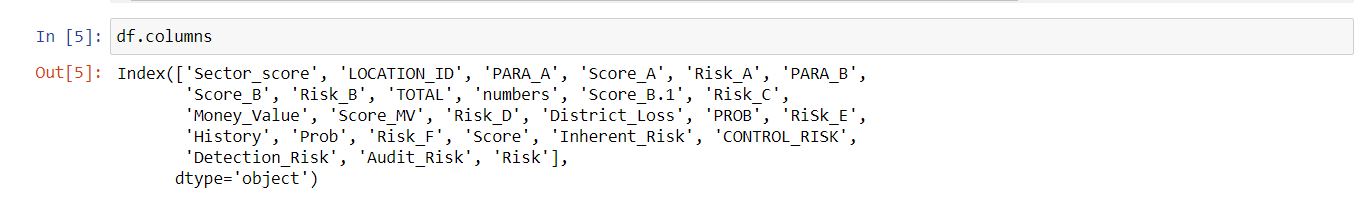
Let’s see how to build a Random Forest model in Python along with some visualization.

# Required Packages are imported initially and the dataset is loaded using read\_csv.

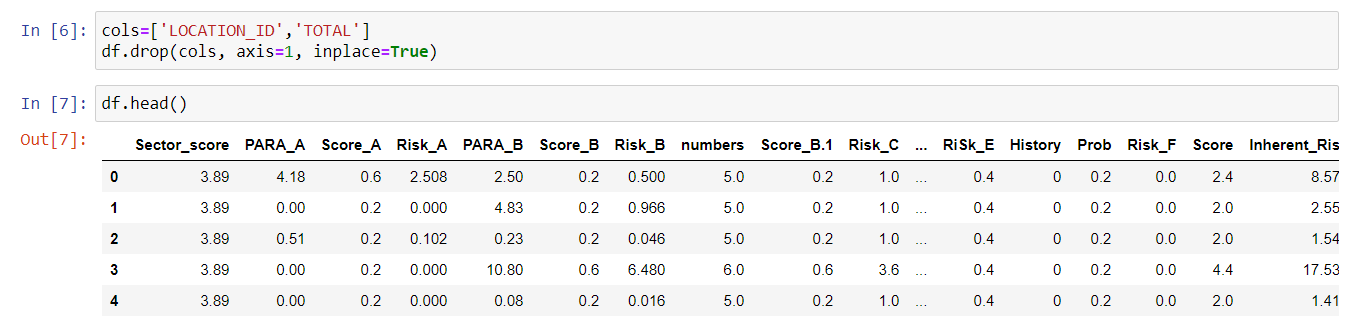


# Let’s have a look on the top 5 rows of the dataset. 

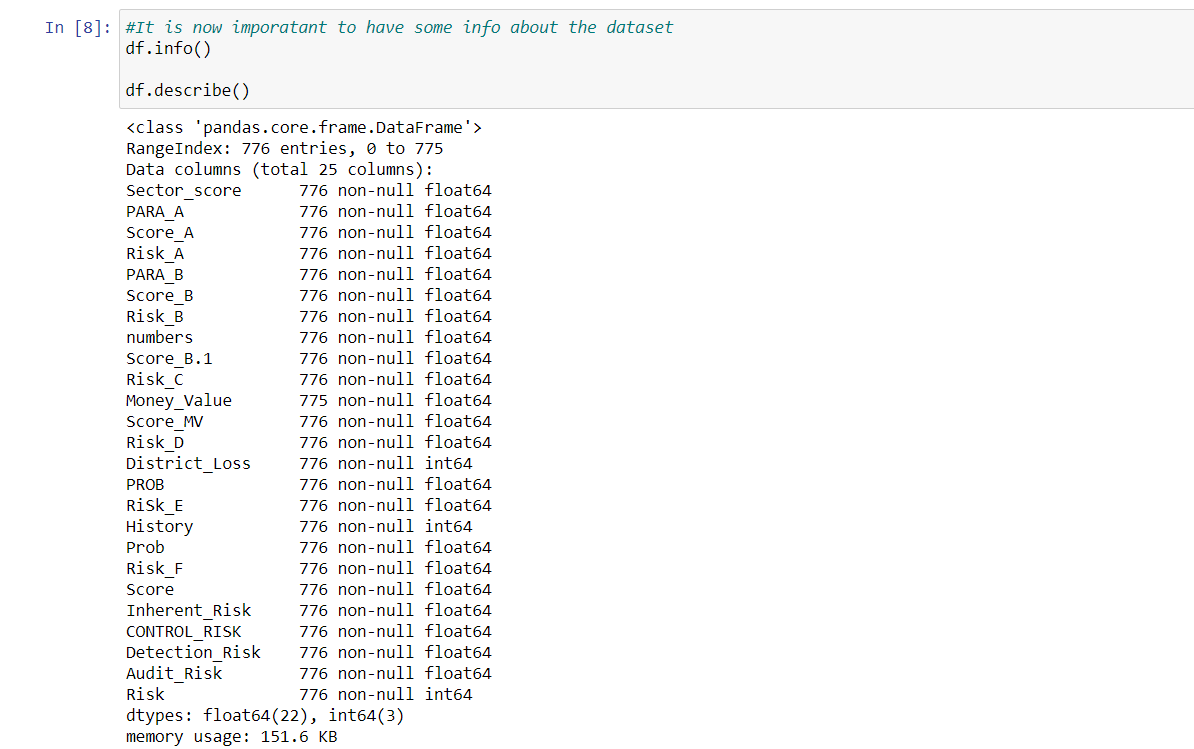
# df.columns to get the name of the columns in our data.



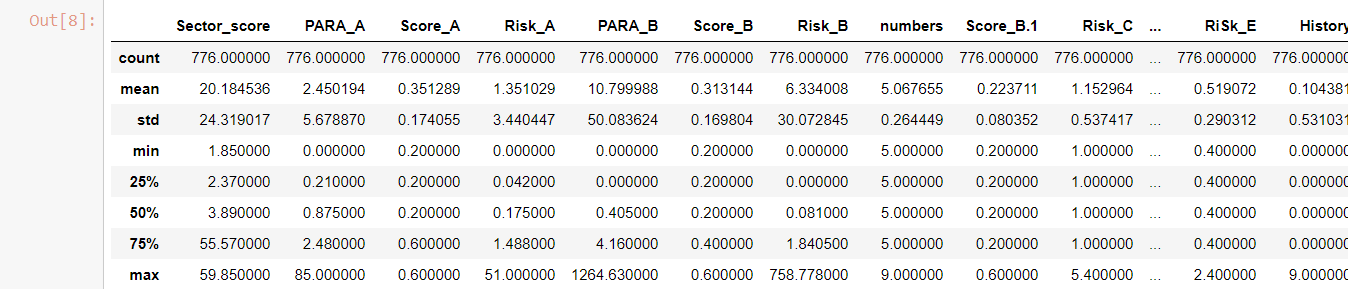
# looking at the columns, df.drop() is used to drop the columns that are not required for model building or prediction.



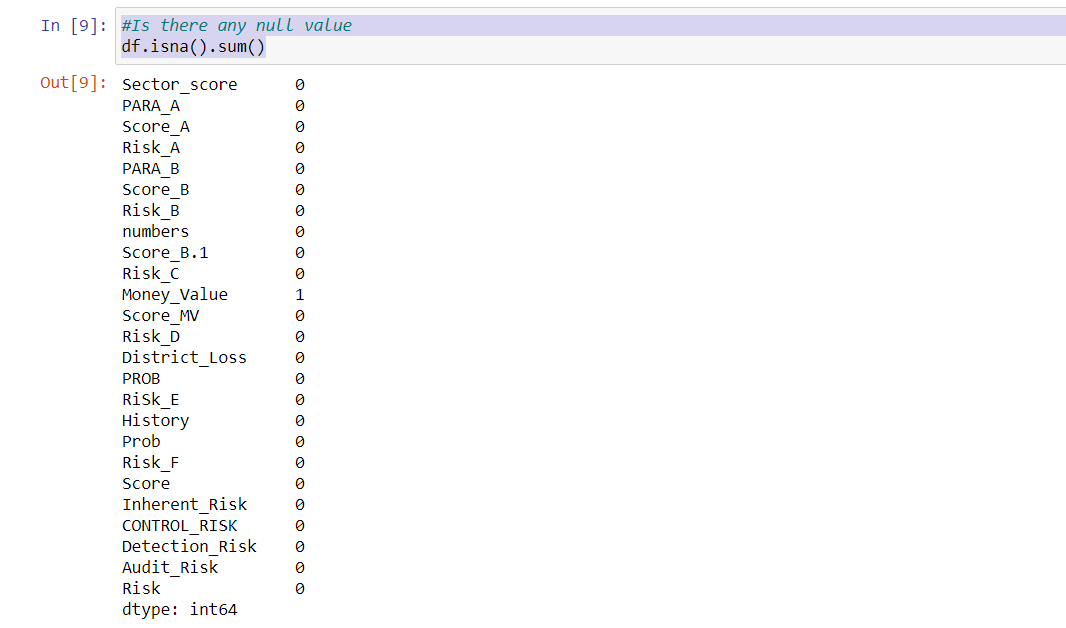
# Now that we know we have only relevant variables, let’s have a look at the data through info() that provides class of data, along with the no. of rows and columns, their data types



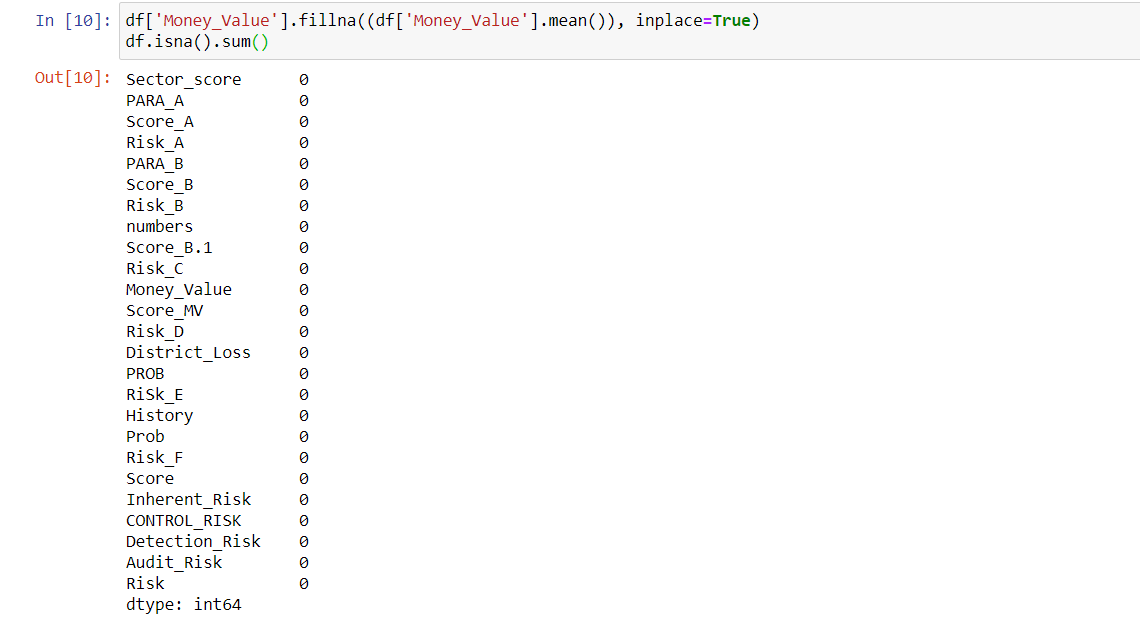
# describe() to provide percentiles, mean, standard deviation, first quartile, third quartile etc.



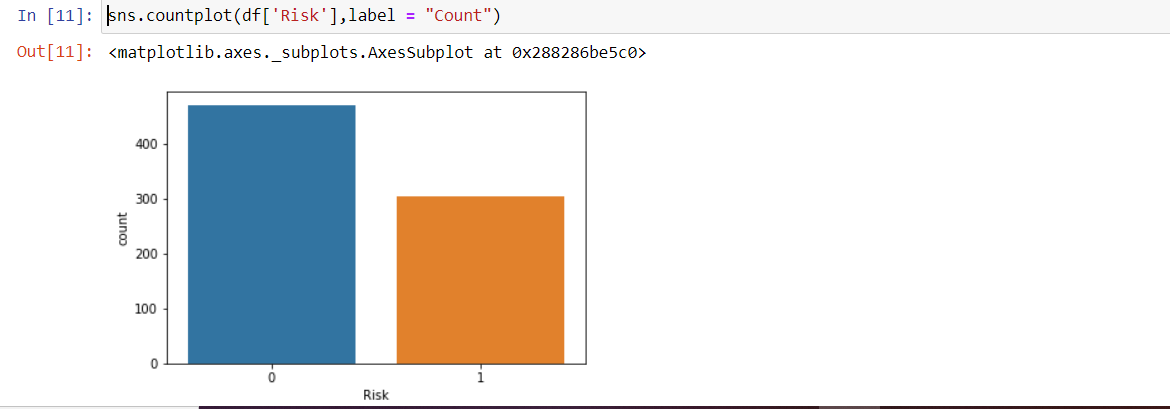
#Next, an important step to be taken is to find null value in the data if any.



#If found, replace them with logical value such as mean, here Money\_Value column has one null value which is being replaced with the mean of its values. After replacing, isna().sum() to check if any null value is still there.



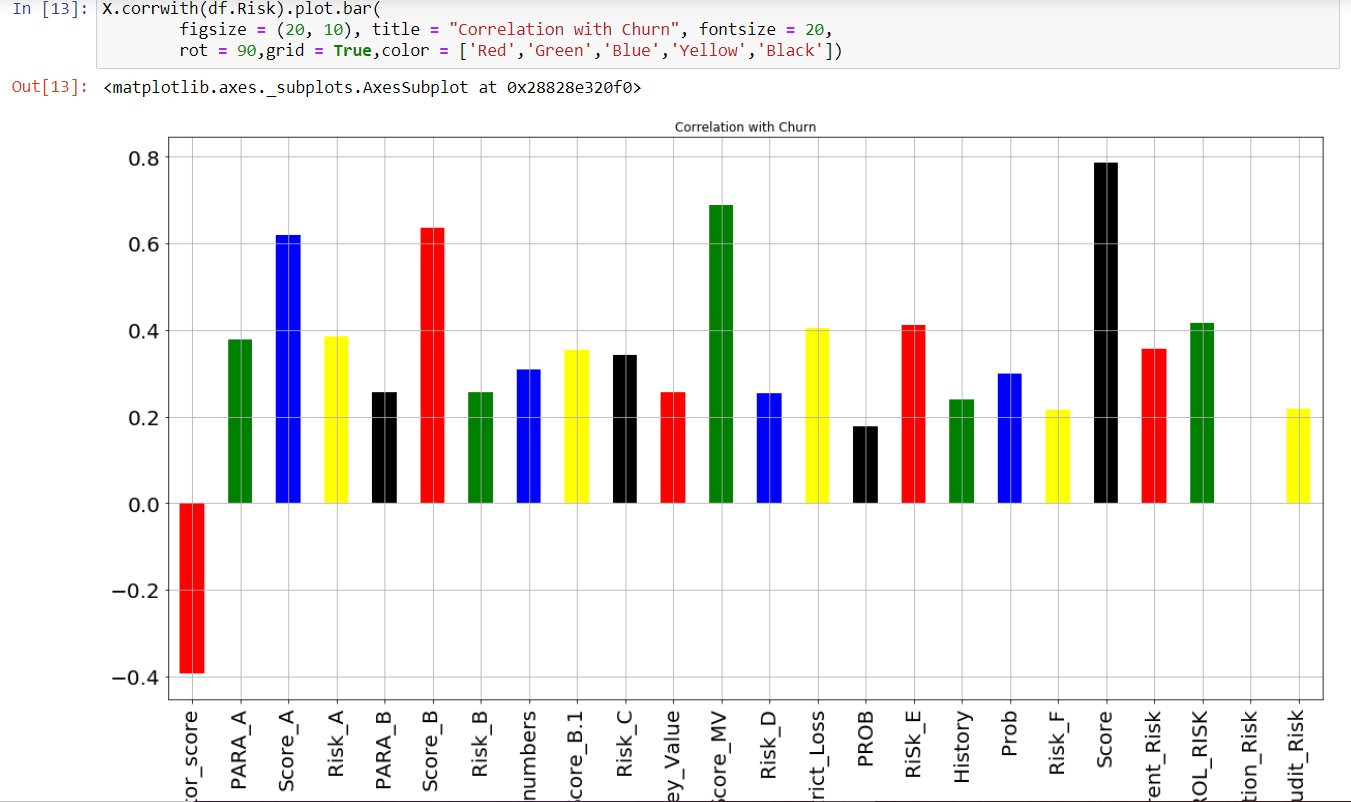
# sns is used for advanced visualizations, here we are plotting the Risk variable with count on Y-axis



# Here we will drop the Risk column from the data

Audit10.PNG

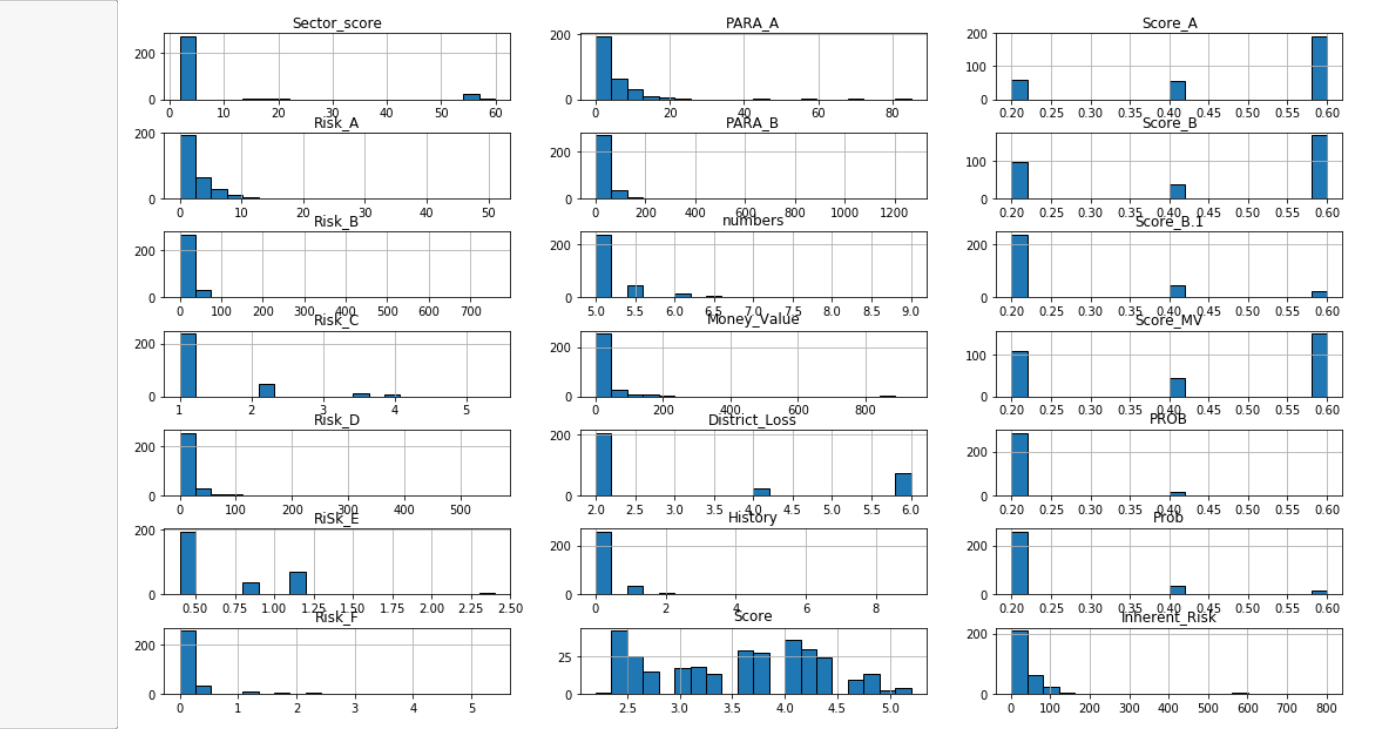
# Plot the correlation of all the input variables with respect to Risk as an Output Variable in barplot form



#Here, the same correlation plotting but with Pearson’s method

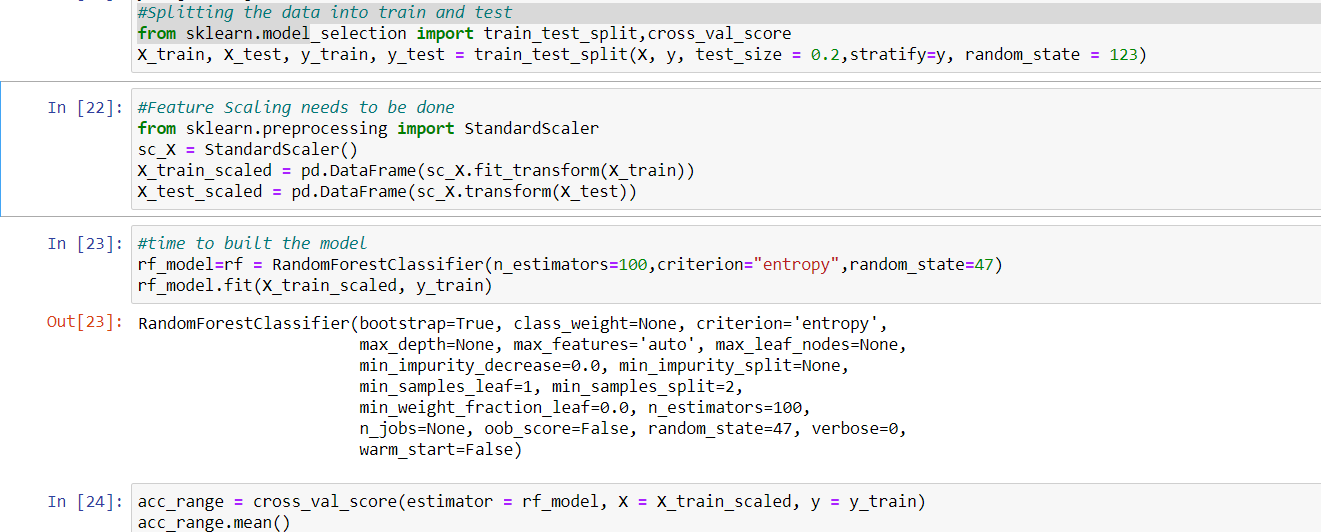


# Here we plotted all the input variables, to see the data distribution in each column

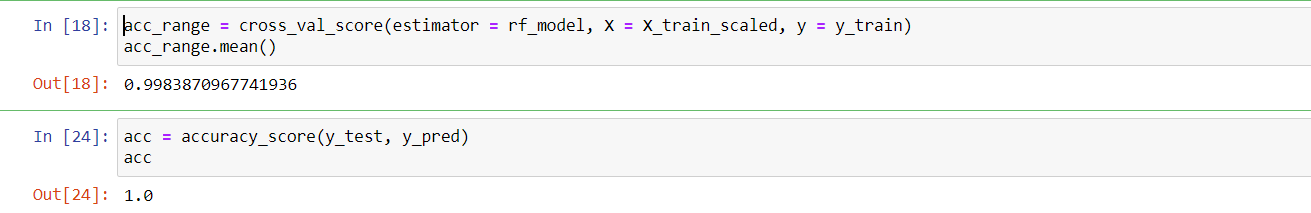


# After visualizing the correlations, replacing null values and other steps of data pre-processing , we proceed further with partitioning the data. Here, after partitioning, we performed scaling on both test and train data to get standardized values.

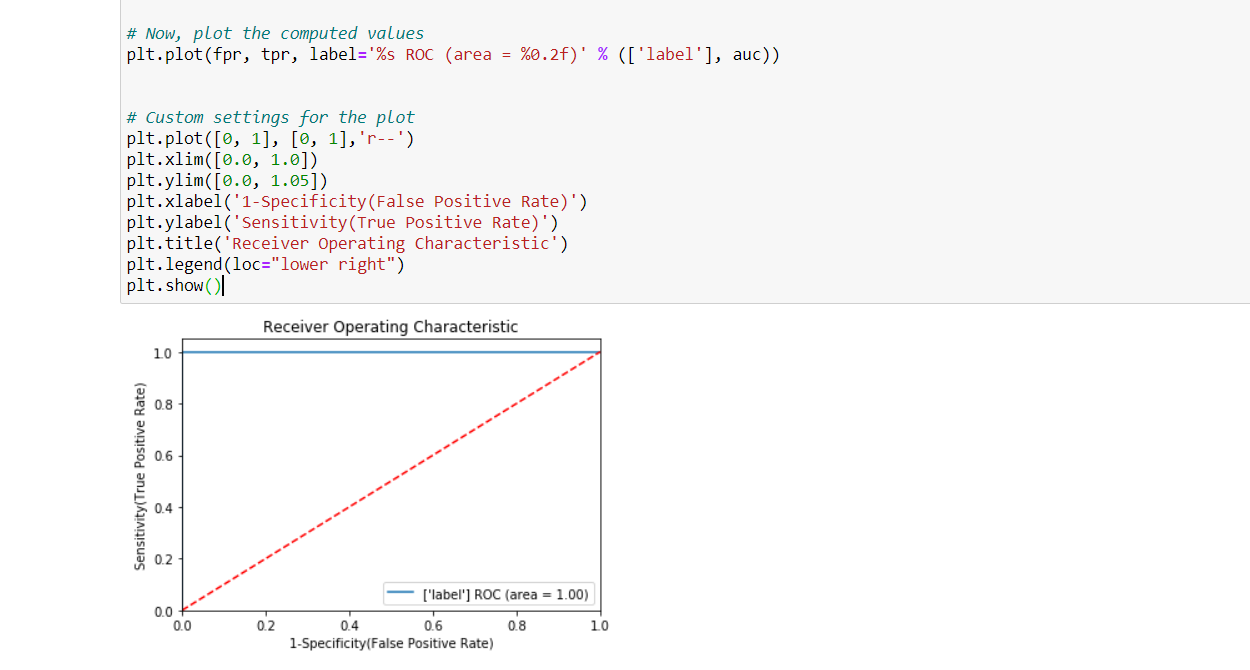
#Next we built a Random Forest Model using RandomForestClassifier() in Python, fitting the model in the train data and checking the accuracy



#This step is to evaluate the accuracy of train and test data



#Till here, we have finished model building, accuracy evaluation, next we will try to plot the ROC curve to get the exact idea of fpr (False Positive Rate), tpr (True Positive Rate)

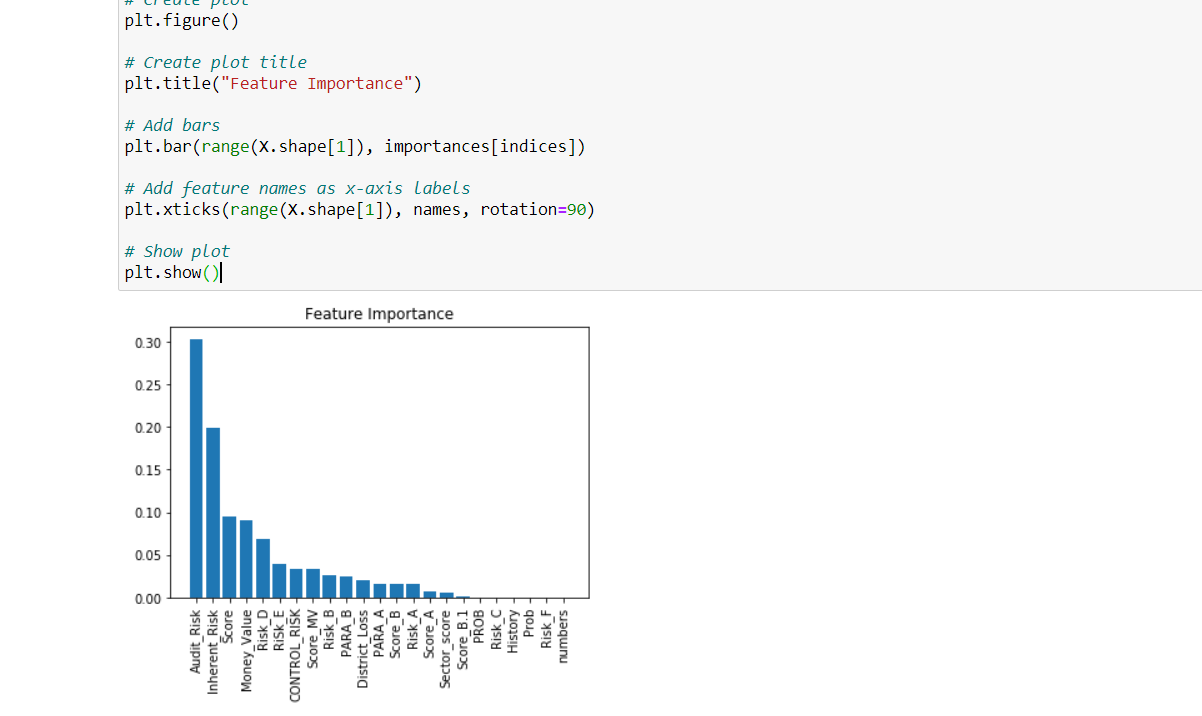


# Building a confusion matrix to see the TP, FP,TN & FN Values in the heat map form



#Let’s have a look at feature importance to figure out what predictor variables the random forest considers most important, that has been extracted from trained model.

This plot shows Audit\_risk is the most influential factor in this dataset, followed by Inherent\_risk



Now that we have some idea on how Random Forest works, next time will come up another very interesting topic. Hope, now you will be able to work in Random Forest algorithm, thereby I leave to come back soon.

