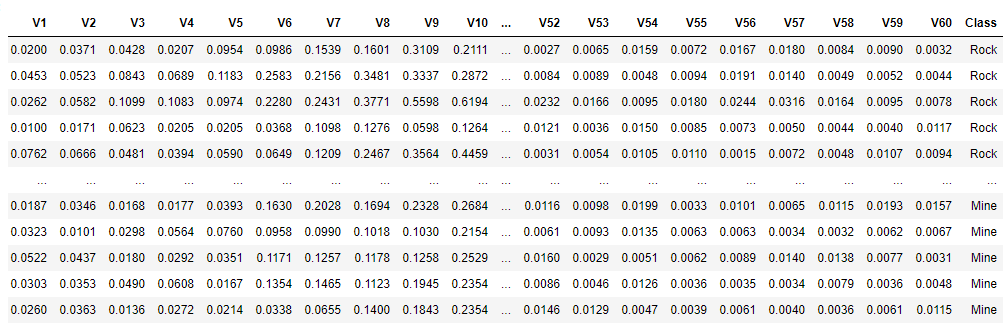
**Problem Definition**  
 This is sonar dataset that contains the data that were obtained by the bouncing sonar signals off a metal cylinder at various angles under various conditions. The sonar signal is a frequency modulated chirp, rising in frequency. From aspect of these angels, spanning 90 degrees is for the cylinder and 180 degrees for the rock. I have done this study to make a predicting model that predicts weather the object is rock or mine, by testing and training the model by the given data that is present in the sonar dataset. This model will help to defence to identify the mines and to decrease the dangerous condition in their path.

**Data Analysis**

I have collected the dataset of the sonar from the UCI repository. The dataset contains 208 instances and 61 attributes. The dataset is in the ‘.csv’ format. For my project I choose jupyter note book. After loading the data into the jupyter notebook checked the information about the columns, all the columns are of float datatype except the last column which is of object datatype. There is no missig values in the data.

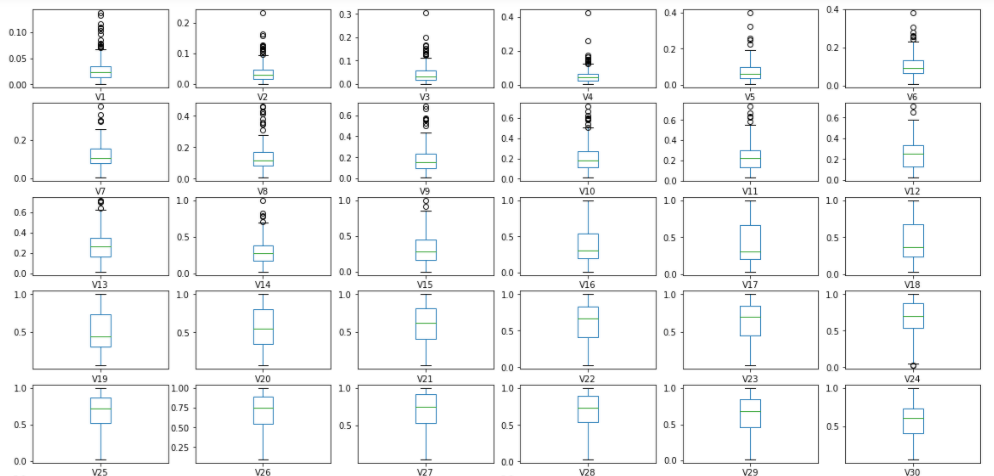


*This is the data imported for the for the prediction, the only the last column ‘class’ have the object datatype rest all of them are of float datatype*

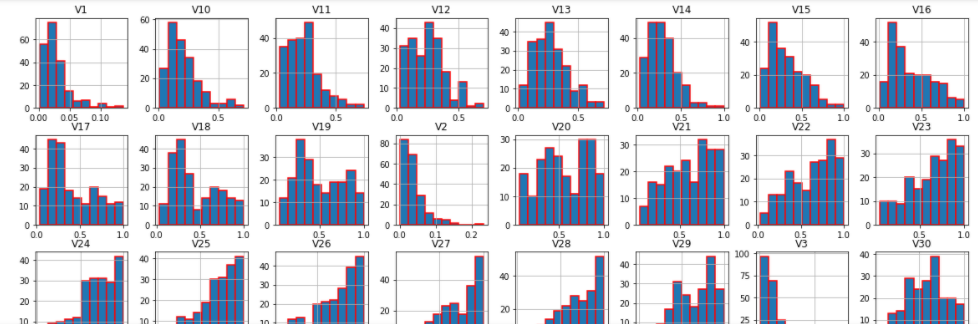
**EDA Concluding Remarks**

This contains the bouncing sonar signals readings from various angles. This is recorded to distinguish the object is weather rock or mine. That is the reason the first 60 columns are the angles which is rising in frequency when going to the right. At last in the ‘class’ column it is distinguished whether it is rock or mine.

From the describe method it can find that there are outliers present in the dataset. When we look the 75th percentile and the max column we can see substantial difference in the value of some columns. When plotted into boxplot by using pandas method (subplots, layout, figsize as parameters) it shows that there are 40 columns in the data have the outliers. After that the data is checked for skewness. Normally +/- 0.5 is the default skewness values. There are some columns with skewness. It is then plotted by using hist function to understand the distribution of the data of each columns. From the graphs it is clear that most of the columns where right skewed. And a few columns are left skewed.

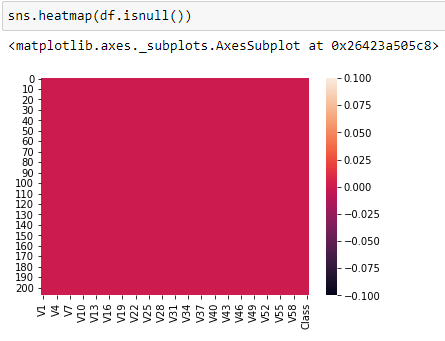


*Above picture shows the box plot of 30 columns to find the outliers in the dataset.*

**

*The above pic shows the distribution of data of each column. From this pic itself it is clear that there are many right skewed data as well as left skewed data are present in the given dataset*

Before removing the skewness I have created new column named ‘class2’ to save the encoded data of the column ‘class’. For encoding reshape function has been used. After the encoding the ‘class’ column was droped. Because there is no use of that column again. To check weather any nan values are present , I have used isnull() function and plot that in heatmap for understanding quickly.

*From the heatmap given below it is clear that there is no missing value in the dataset*

To remove the outliers, I have used 2 methods one is zscore method from the scipy.stats library and the other is IQR(Inter Quartile Range) method. By using zscore method it is showing the less percentage loss of data than the IQR method. So zscore method is used to eliminate the outliers. Then ‘threshold’ was set to 3 and new dataset was made by removing all the outliers that is by using the zscore value which is lesser than 3. The data is then split into x and y in which x consists of independent features other than the target ,which is ‘class2’ column and y consist of ‘class2’ column.

Now all the outliers are eliminated, next from sklearn.preprocessing , power\_transform method was imported. Then

**Building Machine Learning Models**

Before building a model the data should be split for training and testing. This is done by train test split from sklearn.model\_selection.In that x, y , test size and random state is given as the parameters. The testing size was set to 30 percent of the new data and remaining 70 percent is set for the training purpose. Hence it is a classification type dataset LogisticRegression,GaussianNB, SVC, RandomForestClassifer and GradiantBosstingClassifier are used to measure the accuracy\_score.

The first model is LogisticRegression from sklearn.linear\_model. Along with that accuracy score, confusion\_matrix, classification\_report are imported from sklearn.metrics to check the accuracy and error of all the matrix. Then I make instance for LogisticRegression() as lr after that fit the x\_train and y\_train into it then predict the model by using instance lr and print the accuracy\_score(with y\_test and lr\_pred as parameters), confusion\_matrix, classification \_report

After that the second model is GaussianNB which is from sklearn.naive\_bayes. As above I have make make instance for GaussianNB() as nb after that fit the x\_train and y\_train into it then predict the model by using instance nb and print the accuracy\_score(with y\_test and nb\_pred as parameters), confusion\_matrix, classification \_report. In the similar way I have created instances for SVC() as svc, RandomForestClassifier() as rf and GradientBoostingClassifier() as gbc. After that fit all the models with their corresponding insatances and find the accuracy\_score, confusion\_matrix and classification\_report for each of them separately.

**Concluding Remarks**

I choose GradientBoostingClassifier as the best model as it is giving high accuracy score of 82 percent. RandomForestClassifier was giving only 74 percent accuracy. SVC was giving a little more than the RandomForestClassifier, ie 76 percent. GaussianNB and LogisticRegression are giving 66 and 68 percent respectively. Atlast the best model is saved in ‘sonar\_tested.obj’ file.