Develop and evaluate a logistic model to predict the quality (such as high quality and low quality) of red wines according to the several features and calculating test scores. Write your model in Python.

Features that affect wine quality : *'fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol' .*

Please split 80% data for training and 20% data for testing.

Calculate the accuracy of the prediction.

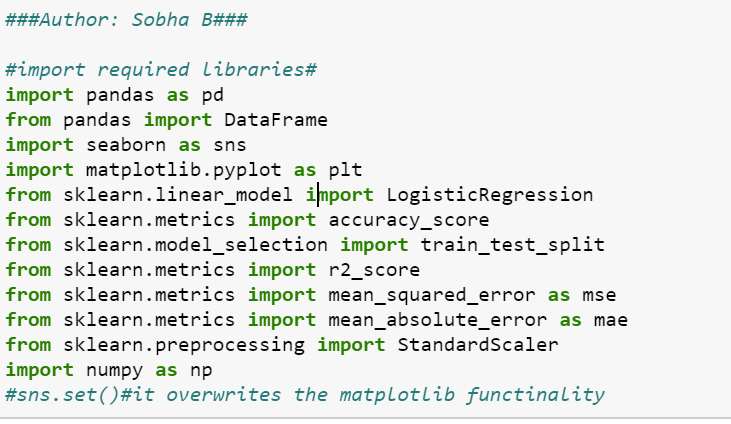
**SOLUTION:**

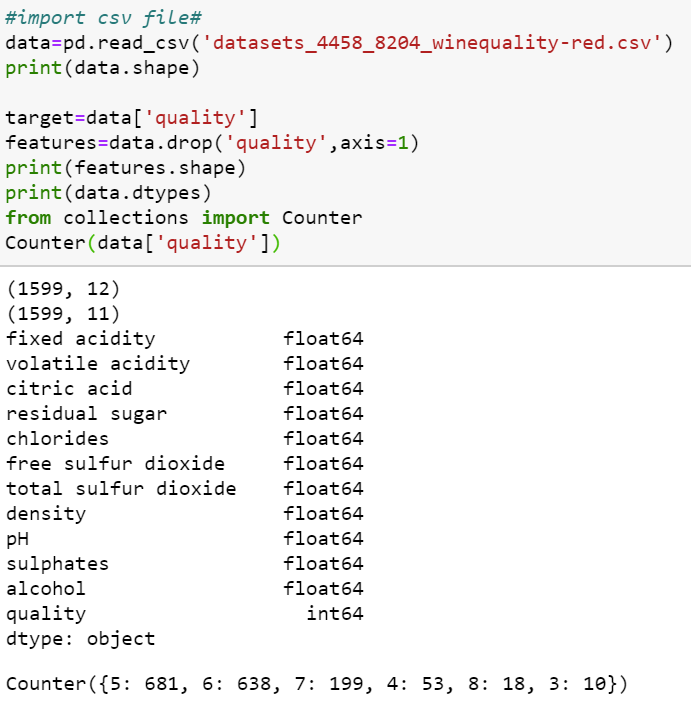
Attached is the Python code generated in Jupyter NB.

**LOGISTIC REGRESSION (LogReg.ipynb):**

**Logistic regression is the one which we use to predict the classification variables.**

* As per the task requirement, I have imported the required libraries and read the datasets\_4458\_8204\_winequality-red.CSV file. From which, I have read all the variables.
* I have created a “target” and “features” variables, considering target as the Dependent and features are the independent variables.
* As per our requirement, I have considered QUALITY variable as the target variable and all others are the feature variable.
* QUALITY is a categorical variable and the values with >=7 are considered as the **high quality** and others as the **low quality** as per the description on source.

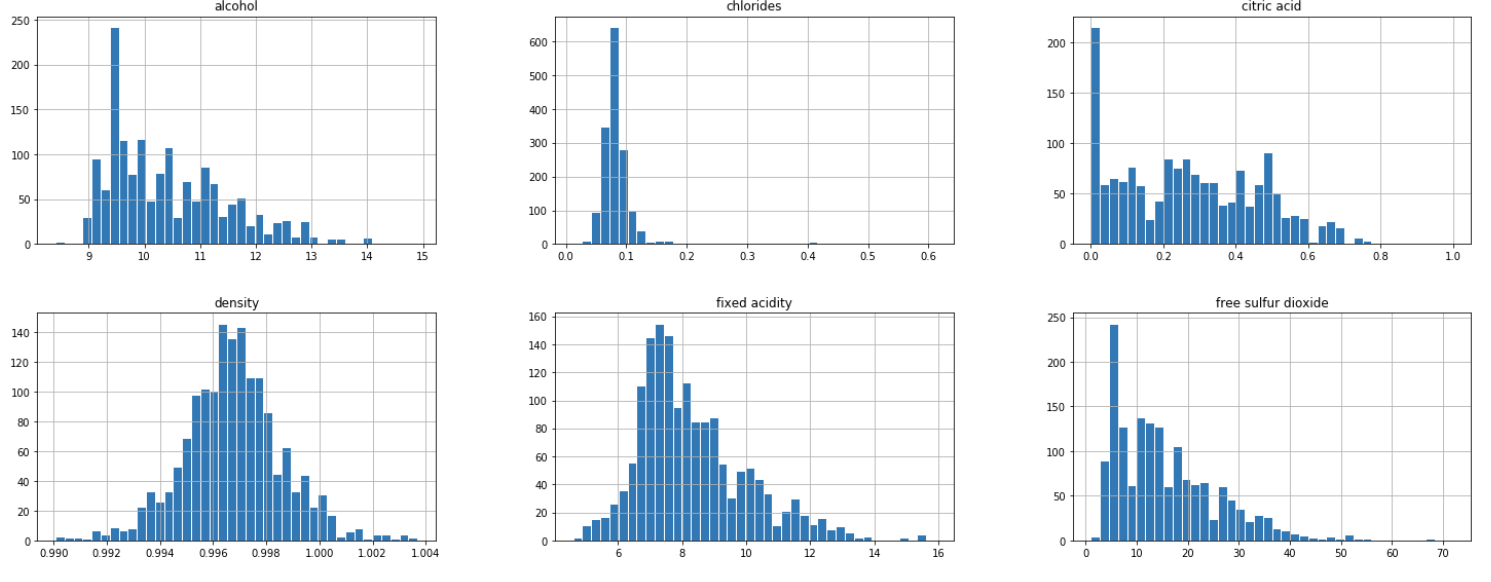


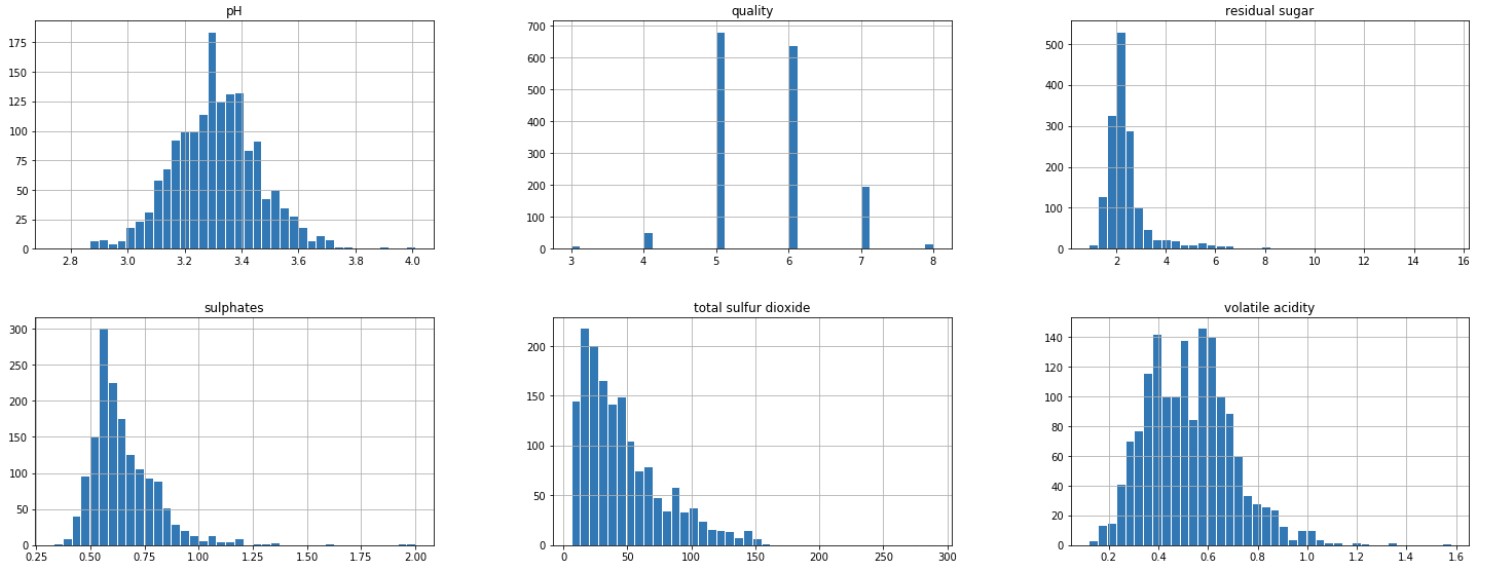


**Histograms:**

* To explore how the data has been distributed, I have created a scatter plots for “HEIGHT”, “WIDTH”, “PICTURE” using seaborn and “SIZE” variables.

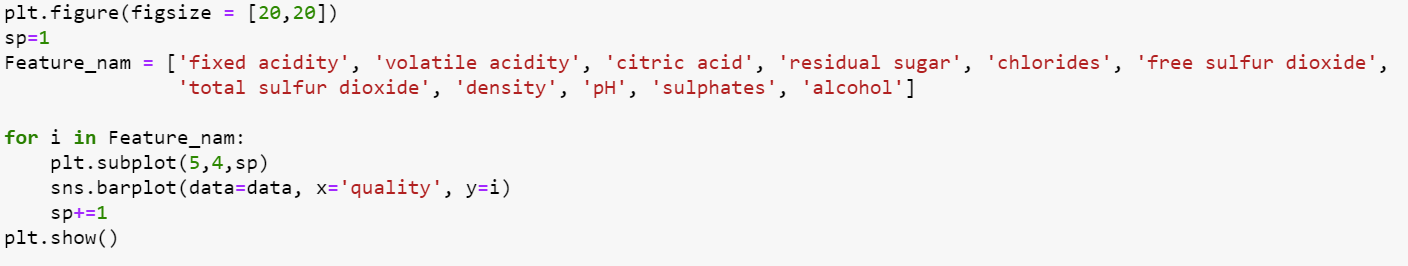


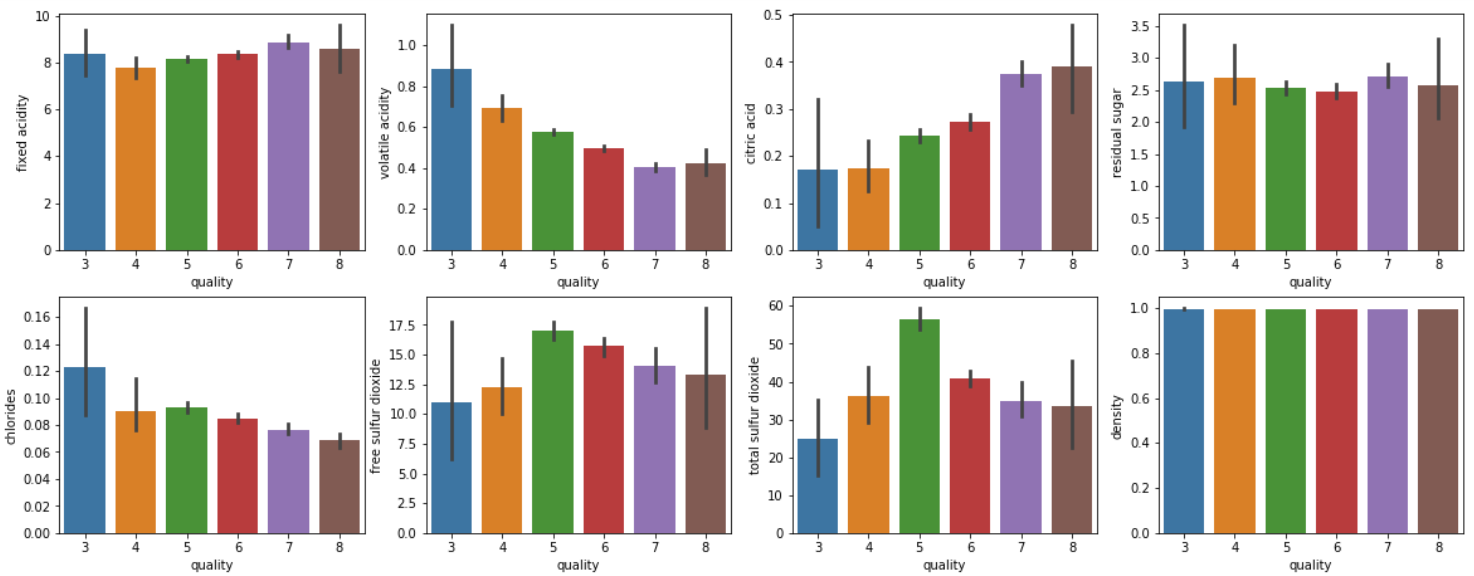


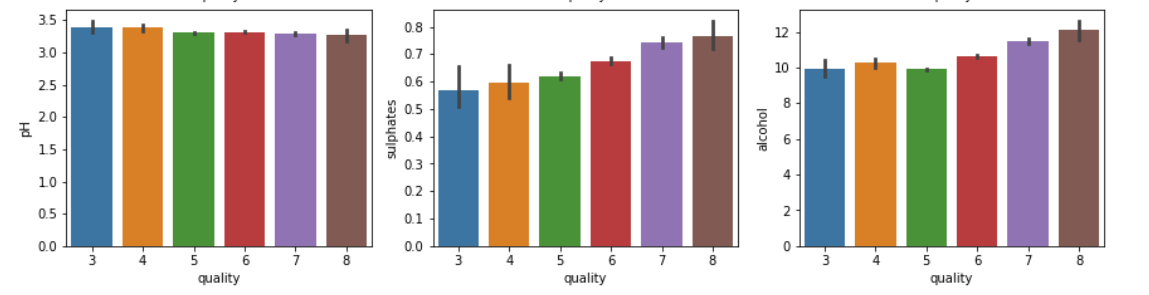


* By looking at the above plots, we can say, whether the data is normally distributed or not and the ranges of specific variables.

**Bar charts:**

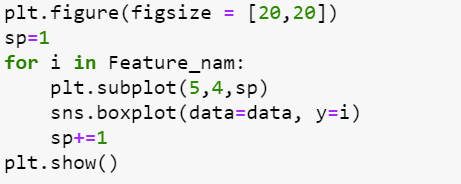


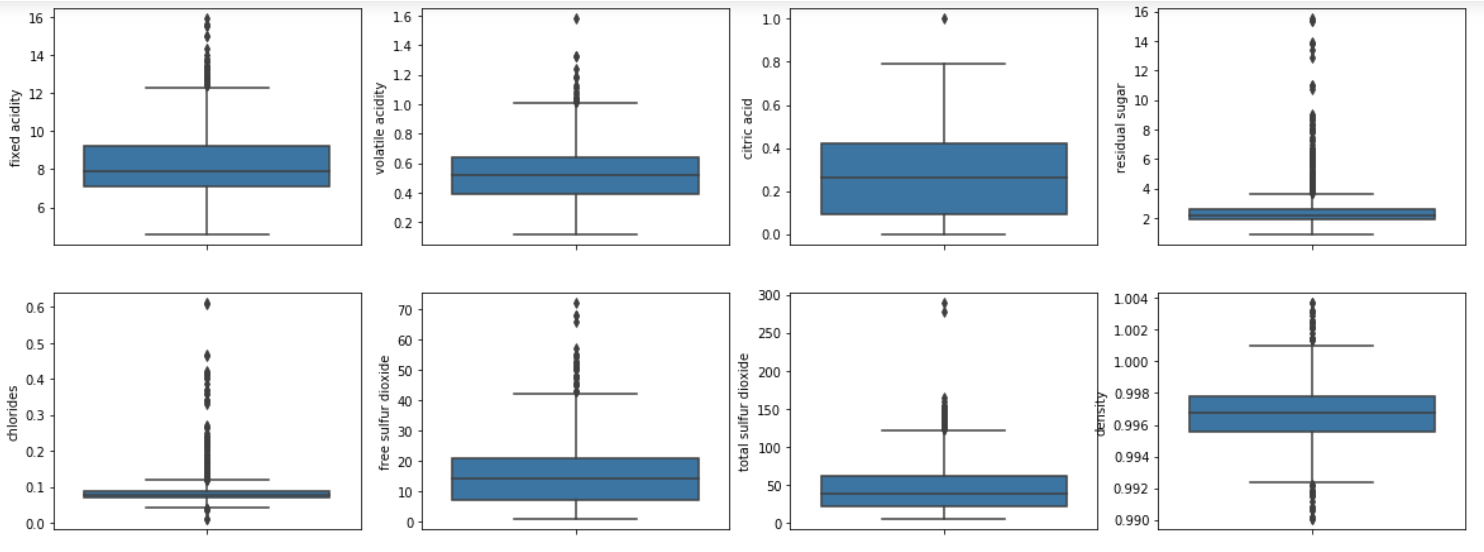


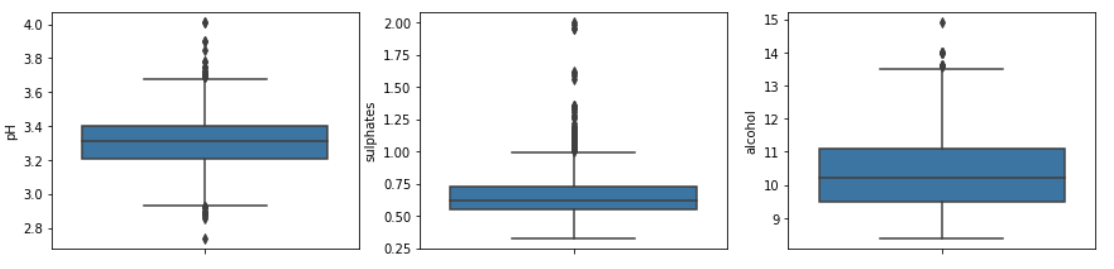


* By looking at the above plots, we can say that the increase of sulphates, citric acid, alcohol increases the quality and the decrease of pH, volatile acidity, chlorides decreases the quality.

**Box plots:**







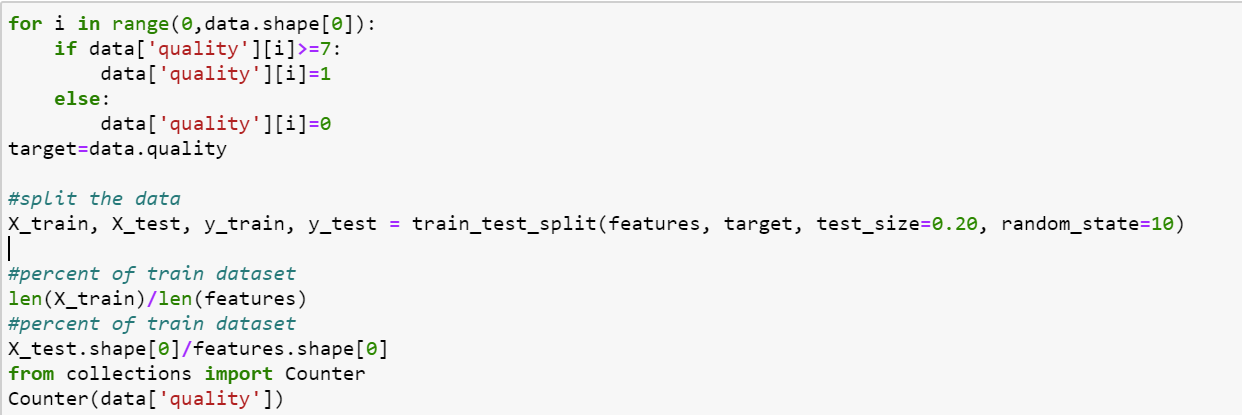
* I have also created box plots, to see if there are any major outliers that affect our data. However, I didn’t find any major outliers.

**Create a classification variable and develop a model:**

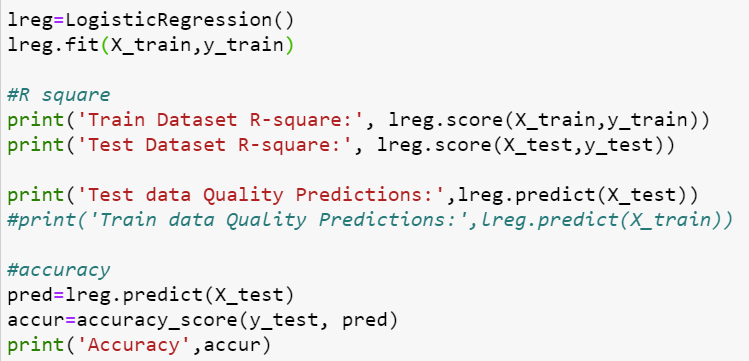
* As per the model, we should create a classification variable, considering the values with >=7 are considered as the **high quality** and others as the **low quality** as per the description on source.
* A classification variable has been created with 0 and 1 values, so this variable has 1382 observations with 0 value (low quality) and 217 observations with 1 value(high quality).
* Once the dependent and independent variables are identified and EDA was performed. The next step is to develop a model.
* I have used Sklearn to develop, train and evaluate logistic regression method.

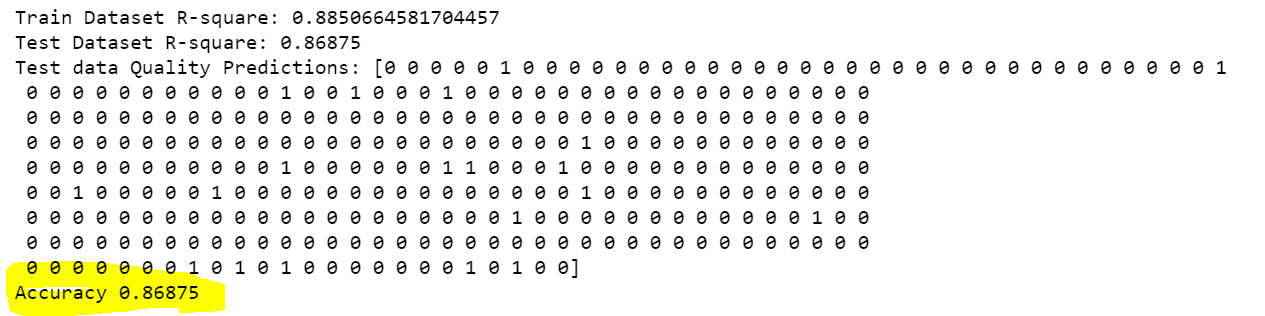
**Then train the data:**

* The data has been trained, and splitted in the ratio of 80:20 for train and test respectively.
* Now the co-efficient and intercept were calculated based on the split data.
* It is straight forward to train regression and make predictions with scikit learn.
* As a first step, I have used the “LogisticRegression()” function from linear\_model module.
* Once the model has selected, I have used fit() function to fit the model on the train set and subsequently used predict() function to perform prediction on the test data.







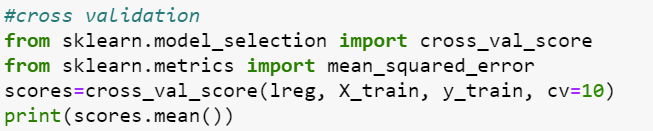


**Calculation of R-square, Predictions and Accuracy:**

* I have calculated the **R-square** to know the goodness of fit and finally **predicted** the results.
* I have checked the R^2 score using the “lreg.score” function on the test data.
* R^2 value should be 0 to 1, which indicates that the data is nearly linear and the independent variable (like size) is correlated to the price.
* The accuracy has been calculated using “accuracy\_score” function and the **accuracy was found as 86.8 which is good.**

**Cross-validation using k-fold cross validation method:**

* In this step, I am going to create a subset of training set, called validation set using a cross\_val\_score() function.
* K-fold cross validation involves randomly dividing the training set into k different folds/ groups of approximately equal size. The first fold is treated as a validation set and the method is fit on the remaining k-1 folds.
* I have used a fold of 10 (cv=10). Hence, 10 folds will be created for cross\_val\_score function.
* For each fold, one value will be generated, however we should take the mean of 10 folds. The obtained score value is almost near to the R^2 score and hence it is validated.





**Loss Functions/ Errors of Prediction:**

* In regression functions, where quantity is predicted, we use mean squared error (MSE) loss function.
* Few basic functions are commonly used, out of which MSE is the popular regression problems.
* Mean squared error (MSE), measures the [average](https://en.wikipedia.org/wiki/Expected_value) of the squares of the [errors](https://en.wikipedia.org/wiki/Error_(statistics)), i.e., that is, the average squared difference between the estimated values and the actual values.
* RMSE, MAE can also be considered.
* Mean absolute error (MAE) is a measure of [errors](https://en.wikipedia.org/wiki/Error_(statistics)) between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed values.
* Whereas RMSE is the square root of MSE
* These can be calculated by importing modules like mean\_absolute\_error, mean\_squared\_error from metrics.
* I ideal values of loss functions are 0, lower the loss, the good is the data. However, there are no hard definitions to define the ranges.

