Importing Library Pandas DataFrame

import pandas as pd

Using Machine Learning 'Random Forest' for Score Classification

def byRandomForest(dataset):  
  
 from sklearn.model\_selection import train\_test\_split  
   
 # Used as the basis/FICO accurate data   
 df = pd.read\_csv('data/FICODATA\_modified.csv')  
 # USed as the current data to predict  
 testingFrame = dataset  
   
 # Using Generated Credit Score as class indicator  
 header = ['payhis','amtowed','lenofcrhis','newcred']  
   
 X = df[header]  
 y = df["FICO Score"]  
  
 # Processing Model  
 X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=14)  
 from sklearn.ensemble import RandomForestClassifier  
 rf\_model = RandomForestClassifier(n\_estimators=50, random\_state=14)  
 rf\_model.fit(X\_train, y\_train)  
  
 from sklearn import metrics  
  
 # Dataset Class Score Prediction  
 preddata = testingFrame[header]  
 result = rf\_model.predict(preddata)  
   
 #Saving to CSV  
 testingFrame["Class Score"] = result  
  
 return testingFrame

Using the'CSPA scoring' ranges for Score Classification

def byCSPA(dataset):  
   
 #######-----------SCORE CATEGORIZATION---------------########  
 #Create new score column  
 dataset['CSPA Score'] = ''  
  
 #Filter Credit Scores Greater than 0 and less than 850.(FICO score range)  
 filter\_score = dataset[(dataset["Score"] > 0)]  
 filter\_score = filter\_score[(filter\_score["Score"] > 0) & (filter\_score["Score"] <= 990)]   
  
 #5. Exceptional Credit Scores  
 filter\_score['CSPA Score'] = filter\_score['CSPA Score'].mask(filter\_score["Score"] >= 831, filter\_score['CSPA Score'].replace('',"Exceptional"))  
  
 #4. VERY GOOD Credit Scores  
 filter\_score['CSPA Score'] = filter\_score['CSPA Score'].mask(filter\_score["Score"] >= 731, filter\_score['CSPA Score'].replace('', "Very Good"))  
  
  
 #3. Good Credit Scores  
 filter\_score['CSPA Score'] = filter\_score['CSPA Score'].mask(filter\_score["Score"] >= 571, filter\_score['CSPA Score'].replace('',"Good"))  
  
  
 #2. Fair Credit Scores  
 filter\_score['CSPA Score'] = filter\_score['CSPA Score'].mask(filter\_score["Score"] >= 400, filter\_score['CSPA Score'].replace('',"Fair"))  
  
  
 #1. Poor Credit Scores  
 filter\_score['CSPA Score'] = filter\_score['CSPA Score'].mask(filter\_score["Score"] > 0, filter\_score['CSPA Score'].replace('',"Poor"))  
  
 return filter\_score

Count each class amount. Used only for Graphing

def obtain(dataset):  
  
 #1. Poor Credit Scores  
 poor = dataset[(dataset["Class Score"] == "Poor")]   
  
 #2. Fair Credit Scores   
 fair = dataset[(dataset["Class Score"] == "Fair")]   
  
 #3. Good Credit Scores   
 good = dataset[(dataset["Class Score"] == "Good")]  
  
 #4. VERY GOOD Credit Scores   
 verygood = dataset[(dataset["Class Score"] == "Very Good")]   
  
 #5. Exceptional Credit Scores]   
 exceptional = dataset[(dataset["Class Score"] == "Exceptional")]   
  
 Total = [len(poor), len(fair), len(good), len(verygood),len(exceptional)]  
   
 return Total