**ENGR 421 HW#7**

**ANIL KUL**

**60337**

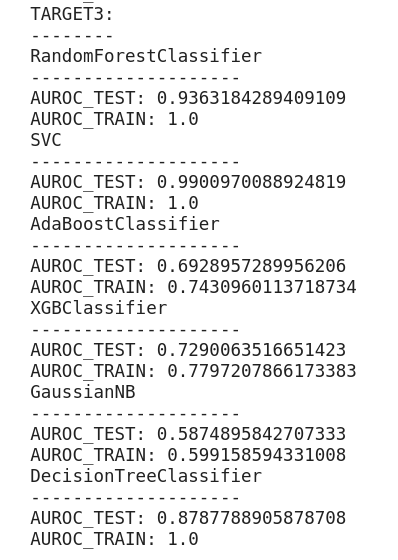
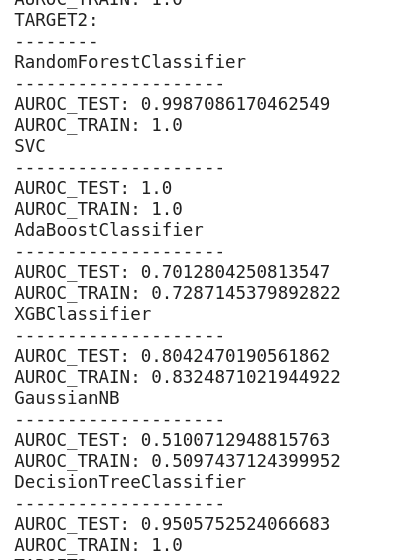
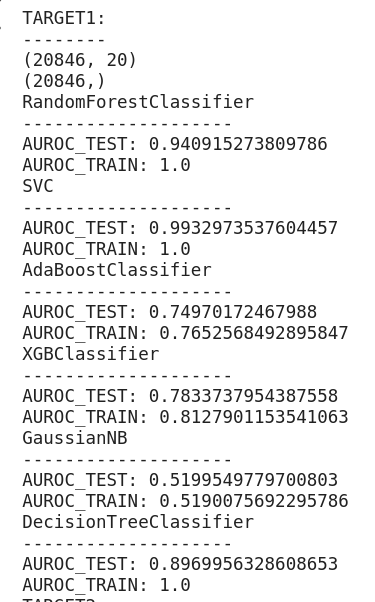
In that homework, we will develop a machine learning algorithm for 3 real life classification problem.

*MODEL:*

For this classifier problem I choose random forest classifier because:

* It is one of the most common algorithm in industry because of its high accuracy.
* It is powerful when working with large datas and high dimensionality.

At the same time to verify the best solution is coming from Random Forest Classifier, I tried some other algorithm such as : support vektor machine, AdaBoost Classifier, XGBoost Classifier, Naive Bayes and Decision Trees. You can see each algorithm performans for each Target problem below.



*PREPROCESSING OF DATA:*

* Dimensionality reduction with PCA
  + I apply dimensionality reduction with PCA because we have to many features which leads to waste of computational power.
* Handling with non-numerical Datas
  + I changed NAN values with mean if column has numeric data otherwise I change them with mode of column which means the most common label in column.
* CROSS VALIDATION
  + Since we haven’t got output in test data I divede train data with 0.2 proportion for test data and validate my model’s accuracy.

def preprocessData(x,x\_test,y):  
 ratio=round((y[y['TARGET']==0].shape[0]) / (y[y['TARGET']==1].shape[0]))  
 x["TARGET"] = y.TARGET  
 duplicate = x[x["TARGET"] == 1]  
 x = x.append([duplicate] \* ratio, ignore\_index=True)  
 y = x.TARGET  
 x = x.drop("TARGET", axis=1)  
 x.drop(x.columns[x.isna().sum() > len(x) \* 0.3], axis=1)  
 x = x.fillna(x.mean())  
 x = x.fillna(x.mode())  
 nonNumericCols = x.dtypes[(x.dtypes != int) & (x.dtypes != float)].index  
 x = pd.get\_dummies(x, nonNumericCols)  
  
 x\_test.drop(x\_test.columns[x\_test.isna().sum() > len(x\_test) \* 0.3], axis=1)  
 x\_test = x\_test.fillna(x\_test.mean())  
 x\_test = x\_test.fillna(x\_test.mode())  
 nonNumericCols = x\_test.dtypes[(x\_test.dtypes != int) & (x\_test.dtypes != float)].index  
 x\_test = pd.get\_dummies(x\_test, nonNumericCols)  
 missing\_cols = set(x.columns) - set(x\_test.columns)  
 for c in missing\_cols:  
 x\_test[c] = 0  
 missing\_cols\_test = set(x\_test.columns) - set(x.columns)  
 for c in missing\_cols\_test:  
 x[c] = 0  
 pca = PCA(n\_components=20, whiten=False, random\_state=2019)  
 x = pca.fit(x).transform(x)  
 x\_test = pca.transform(x\_test)  
 return x,x\_test,y