ARTIFICIAL INTELLIGENCE

FINAL SEMESTER PROJECT

PHASE 3

TRAINING, CLASSIFICATION AND MODEL EVALUATION

GROUP IV (4)

**GROUP MEMBERS**

EMMANUEL SAAKE

VIKRAM VENKATESH GOPALAN

SUNIL SRINIVAS

HAMZA SHAHID

SECTION ONE

PROJECT OBJECTIVE, CONCEPT AND PROJECT DESIGN

* 1. Project Objective

Practice design of classification model, training of models using samples and evaluation of classification model

* 1. Concept of Classification

Machine learning is a technique of applying artificial intelligence to impact knowledge into a system, to give it the ability to automatically learn and make inferences or solve problems base on experience without being explicitly programmed to do so.

Under Machine learning we have the following categories:

* Supervised machine learning
* Unsupervised machine learning
* Semi-supervised machine learning and
* Reinforced machine learning algorithms

There are two types of Supervised learning, Classification and regression. The focus of this assignment is classification. Supervised learning involves training a machine learning model on a data with labels and secondly making predictions or inferences on another set of data.

Assuming we have multiple categories of items and there is a new observation (item) whose category must be determined by a procedure(algorithm). Identifying the category to which the new observation belongs is classification.

* 1. Classification Models

In this project three (3) major classification models were selected, applied and evaluated

* K Nearest Neighbor (KNN)
* Random Forest
* Support Vector Classifier (SVC)
  1. Process Flow in Codes
* Start
* Data Normalization using MinMaxScaler
* Read data as X values
* Read target [1] as Y values
* Definition of classifier models
  + KNN /Random forest/SVC definition
* Splitting of data using train\_test\_split
* Call of classifiers on train and test data set
* Drawing of ROC curve and AUC computation
* Balancing data with SMOTE
* Evaluating the 3 classifier models with 10\_Fold cross validation
* End

SECTION TWO

IMPLEMENTATION AND OUTPUT

2.1. Importation and Data Loading



2.2. Normalization of Data



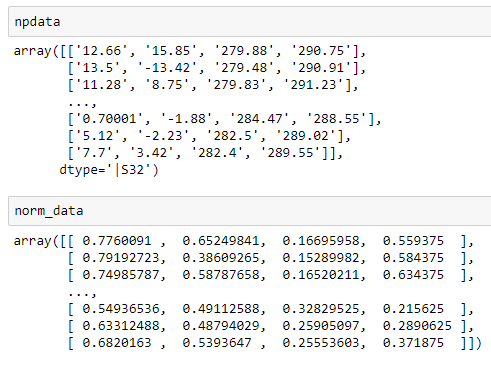


Figure 1: unnormalized data(above) versus normalized data [range 0,1] (below)

2.3. Initializing X and Y Data





Figure 2: X and Y shape print out

2.4 Defining Classifier models





2.5. Calling Classifiers

2.6 Outputs from calling of classifiers

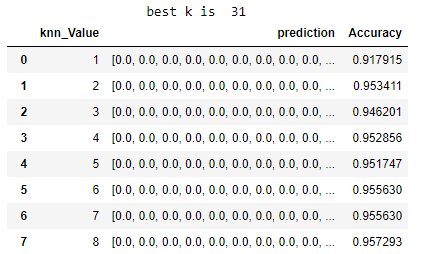


Figure 3: KNN values, predictions and accuracy scores (continues to 100)

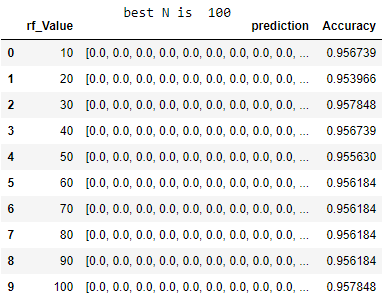


Figure 4: various number of decision trees, predictions and accuracy scores

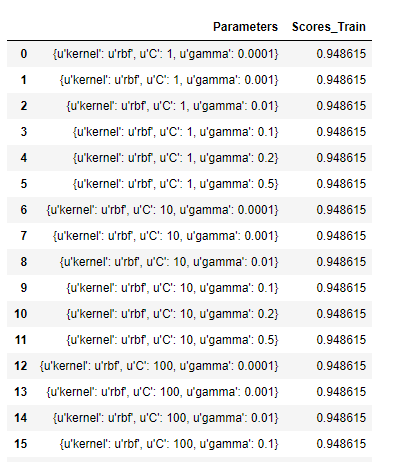


Figure 5: Training scores and parameters from the GridSearchCV parameter tuning

2.7. ROC and Area Under Curve (AUC ) FUNCTION [Using Prediction Probability]



2.8. Calling ROC and AUC function for each Classifier and Outputs



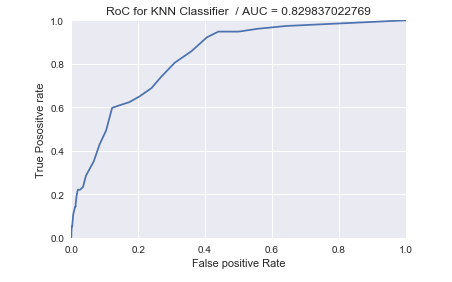


Figure 6: KNN ROC and AUC scores

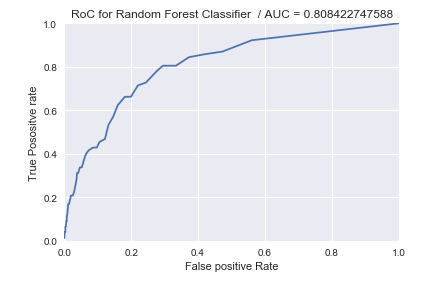


Figure 7: Random Forest RoC and AUC scores

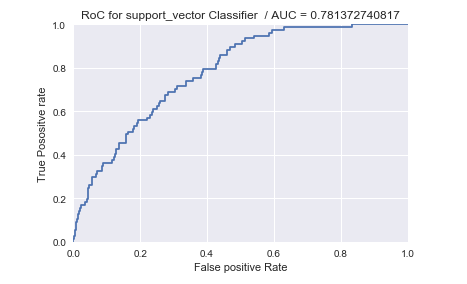


Figure 8: Support Vector Classifier RoC and AUC

3.9 Setting up Best Parameters for Each Model



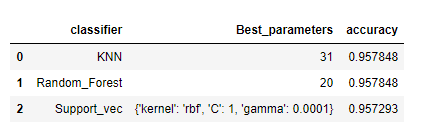


Figure 9: Best parameters for each classifier to be used to run the 10-Fold CV

2.10. Function to Compute report on Classifiers Before With Unbalanced Data for Evaluation

SECTION 3

BALANCING DATA AND EVALUATION

3.1. SMOTE(Synthetic Minority Oversampling Technique).

In balancing the data, we used the imbalanced learn library and applied Over\_Sampling with SMOTE(Synthetic Minority Oversampling Technique). This technique generates more of the minority class synthetically, to even up the difference in number of samples in terms of Minority and Majority. It was learned that applying SMOTE before splitting into train and test set , results in leakage of data; so we first split data before applying the SMOTE on the training set and then predicting on the actual data. Kfold cross validation was then applied.





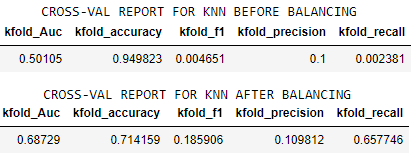


Figure 10: Evaluation report on KNN Classifier Before and After Balancing Data

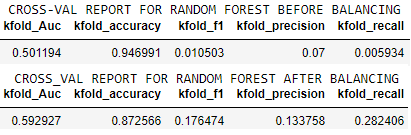


Figure 11: Evaluation report on random Forest Classifier Before and After Balancing Data

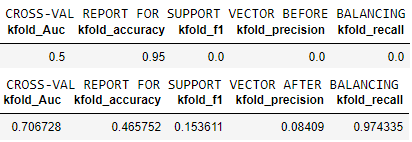


Figure 12: Evaluation report on SVC Classifier Before and After Balancing Data

3.2 REDRAWING ROC WITH PREDICTION AFTER DATA BALANCING



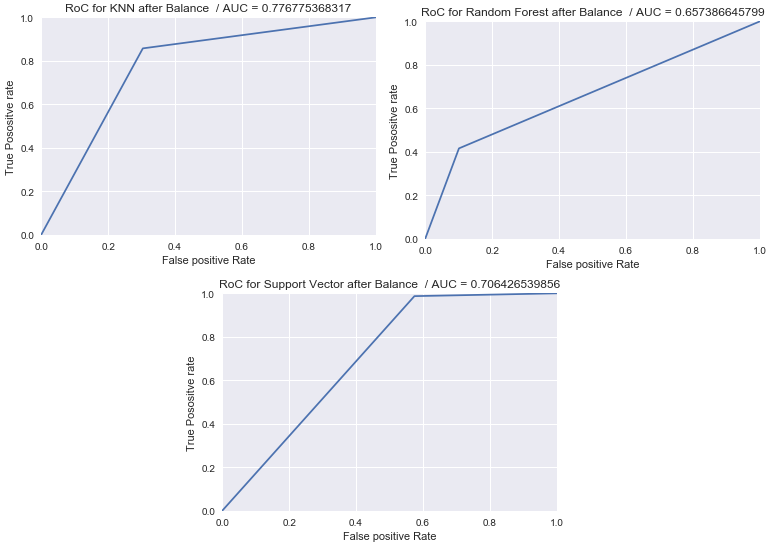


Figure 13: Redrawn ROC of classifiers for Balanced data and using actual predictions

CONCLUSION

During the execution of program, some few observations were made between sample selection with ‘train\_test\_split’ and ‘kfold\_Cross\_Validation. Although ‘train\_test\_split’ is quite simple, Kfold\_Cross\_Validation, run turns with data for the purpose of training and testing. It was observed that for ‘train\_test\_split’, performance depends extensively on the data being used for training and testing. However, ‘K-fold’ reduces this impact by running turns for data from it k-splits for training and testing, thus the eventual score is the mean of the total score from each of the k runs.

With unbalanced data, the accuracy of the model was very high. This was because the samples were biased towards the 0-classes, so even if the model predicts all zeroes, it will end up with very high score. During most instances all predictions were Zeroes.

ROC and AUC was drawn and calculated respectively in two different ways. Before Balancing, the **test\_labels** with **‘prediction\_probability’** of the model was used to draw ROC and compute AUC. Then After Balancing the data, ROC and AUC was redrawn and recomputed respectively. Thus, two set of drawings were shown in the report. Moreover, AUC from K-fold is different from AUC from ‘train\_test\_split’; thus, few variances across.

For performance**, Before Balancing**, using ‘train\_test\_split’ AUC score indicated **KNN outperformed** the other Models. However, Scores from the10-fold\_Cross validation puts Support Vector Classifier as the best.

**After Balancing**, in terms of Accuracy for 10-fold CV, Random Forest was the best nevertheless, looking at other metrics like ‘recall’ (Best was Support Vector), precision (best -Random Forest ) and ‘F1’ (Best was KNN) shows varying outputs. These outcomes are shown in Figure 11, Figure 12 and Figure 13.