CS 6375
ASSIGNMENT LAB ASSIGNMENT 2

Names of students in your group:

Tushar Gonawala (TXG170003)

Number of free late days used: 0

Note: You are allowed a **total** of 4 free late days for the **entire semester**. You can use at most 2 for each assignment. After that, there will be a penalty of 10% for each late day.

Please list clearly all the sources/references that you have used in this assignment.

Dataset:

Name: Wine recognition data
Source: https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data
Number of Instances: 3
Class 1 - 59
Class 2 - 71
Class 3 - 48
Number of Attributes: 13
All attributes are continuous.
None Missing Attribute Values.

Code:

The code is included in the zip file that is attached along with the word document.

In order to change the running algorithm user need to enter the value as per the menu list provided on the console at runtime. Here is the list:

1. DecisionTree
2. NeuralNet
3. SVM (Support Vector Machine)
4. GNB (Gaussian Naive Bayes)
5. LR (Logistic Regression)
6. knearest (k - Nearest Neighbor)
7. Bagging
8. RandomForest
9. AdaBoost
10. GBC (Gradient Boosting Classifier)
11. XGBoost

The user input should match the list names and these are case sensitive.

Output:

Algorithm	Best Parameters	Avg. Precision	Avg. Recall	Avg. F1	Accuracy Score
Decision Tree	{'min_weight_fraction_leaf': 0.1, 'max_depth': 5, 'max_features': 7, 'min_samples_leaf': 10, 'max_leaf_nodes': 70}	0.85	0.84	0.83	0.8333
Neural Net	{'hidden_layer_sizes': (100, 20), 'activation': 'logistic', 'max_iter': 500, 'learning_rate': 'constant', 'solver': 'adam'}	0.97	0.97	0.97	0.9722
Support Vector Machine	{'C': 1, 'gamma': 0.001, 'kernel': 'linear' ‘degree’: 3}	0.96	0.97	0.97	0.9722
Gaussian Naive Bayes	{'priors': [.3,.4,.3] }	1.0	1.0	1.0	1.0
Logistic Regression	{'multi_class': 'multinomial', 'C': 1, 'max_iter': 100, 'solver': 'newton-cg', 'tol': 0.0001, 'penalty': 'l2'}	0.94	0.94	0.94	0.9444
k - Nearest Neighbor	{'p': 1, 'weights': 'distance', 'algorithm': 'ball_tree', 'n_neighbors': 10}	0.83	0.83	0.83	0.8333
Bagging	{'n_estimators': 100, 'random_state': 10, 'max_features': 10, 'max_samples': 4}	0.93	0.92	0.92	0.9167
Random Forest	{'max_depth': 14, 'n_estimators': 20, 'max_features': 5, 'criterion': 'gini'}	0.97	0.97	0.97	0.9722
AdaBoost Classifier	{'algorithm': 'SAMME.R', 'random_state': 4, 'learning_rate': 1.5, 'n_estimators': 100}	0.97	0.96	0.97	0.9722
Gradient Boosting Classifier	{'max_depth': 1, 'learning_rate': 1.0, 'n_estimators': 20, 'loss': 'deviance'}	1.0	1.0	1.0	1.0
XGBoost	{'booster': 'gblinear', 'learning_rate': 1.0, 'max_delta_step': 1, 'n_estimators': 100, 'seed': None}	0.94	0.94	0.94	0.9444

Results:

Gaussian Bayes and Gradient Boosting Classifier provided the best accuracy on Test data with the parameters as per the above table. The reason for Gaussian Bayes to perform well as compared to other algorithm is because the attributes were independent from each other and this helps reduce correlation and redundancy in data, eventually leading to a much better accuracy. Gradient Boosting Classifier performed well because the algorithm uses weak classifiers and gradually increases accuracy by introducing new weak learners which compensate the weaknesses of existing weak learners.

To increase the efficiency of the model further pre-processing like normalization or mean can be applied on the dataset before training the model. Moreover, it is also effective to reduce all the dependencies on attributes to have a clean dataset which would help increase the accuracy.