CS 6375
SciKit Lab 2

PSD170000
Paril Doshi

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.

Output:

Algorithm	Best Parameter	Avg Precision	Avg Recall	Avg F1	Accuracy Score
Decision Tree	'max_depth': 5, 'max_features': 3, 'max_leaf_nodes': 1600, 'min_impurity_decrease':0.0001, 'min_samples_leaf': 4, 'min_samples_split': 9, 'min_weight_fraction_leaf': 0	0.9483	0.9606	0.9831	0.9730
Neural Net **(Found best among all)**	'activation': 'tanh', 'alpha': 0.1, 'early_stopping': False, 'hidden_layer_sizes': (5, 5, 5), 'learning_rate': 'constant', 'max_iter': 1000, 'momentum': 0.9, 'tol': 0.0001	1.0	1.0	1.0	1.0
Support Vector Machine	'C': 10, 'gamma': 0.001, 'kernel': 'linear'	0.9755	1.0	0.9881	0.9902
Gaussian Naïve Bayes	'priors': [0.5, 0.5]	0.7786	0.9528	0.8443	0.8398
Logistic Regression	'C': 1.0, 'fit_intercept': True, 'intercept_scaling': 1, 'max_iter': 100, 'multi_class': 'ovr', 'penalty': 'l2', 'solver': 'newton-cg', 'tol': 0.0001	0.9619	1.0	0.9729	0.9757
K-Nearest Neighbors	'algorithm': 'auto', 'n_neighbors': 10, 'p': 1, 'weights': 'uniform'	1.0	1.0	1.0	1.0
Bagging	'max_features': 3, 'max_samples': 10, 'n_estimators': 17, 'random_state': 8	0.8586	0.9943	0.8901	0.9077

Random Forest	'criterion': 'gini', 'max_depth': 20, 'max_features': 1, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 10	0.9862	0.9673	0.9863	0.9927
Ada Boost Classifier	'algorithm': 'SAMME', 'base_estimator': DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort=False, random_state=None, splitter='best'), 'learning_rate': 1, 'n_estimators': 10, 'random_state': 4	0.9575	0.9787	0.9735	0.9757
Gradient Boosting Classifier	'learning_rate': 0.4, 'loss': 'deviance', 'max_features': 1, 'n_estimators': 125	0.9809	1.0	0.9919	0.9902
XGBoost	'learning_rate': 1, 'max_delta_step': 1, 'min_child_weight': 1, 'n_estimators': 10, 'seed': 1	1.0	1.0	1.0	1.0

I have submitted the code in file(Learn SciKit 1.py). I have also uploaded the code on google colab.

Colab Link:-

https://colab.research.google.com/drive/13Hx5ySQfUMnQrBWcYuTjN_2fRaE9l7wH

There is a need to input the index number of classifier while running the code to select the algorithm needed to be run on the dataset. The number ranges from (1 to 11).

```
1. Decision Tree
2. Neural Net
3. Support Vector Machine
4. Gaussuian Naive Bayes
5. Logistic Regression
6. K-Nearest Neighbors
7. Bagging
8. Random Forest
9. AdaBoost Classifier
10. Gradient Boosting Classifier
11. XGBoost
Please enter the respective number of classifier you want to use
```

The dataset link is:- https://archive.ics.uci.edu/ml/machine-learning-databases/00267/data_banknote_authentication.txt

Report on results:

- Having used the best combination of parameters for all Algorithms, the accuracy score for all the algorithms have reached above 90% except Naïve Bayes.

- The best classifier for bank note authentication dataset found is Neural Net, K-Nearest Neighbors, XGBoost.

Why is Neural Net best algorithm:
Firstly, tanh is the best activation function. It is because of there are only two classes(0 and 1) the data needs to be classified in.

The second property of tanh is that it maps the negative inputs strongly negative and zero inputs maps near zero. As the current dataset is having negative values tanh outperformed other activation functions.

The confusion matrix of this displayed that there were 0 points misclassified.

The runner up algorithm found is K-Nearest Neighbors Algorithm.

- There are already three classifier(Neural Net, Gradient Boosting Classifier, K-Nearest Neighbours) who predicts the test data to 99% not much scope of further increase in accuracy.