CS 6375
ASSIGNMENT Scikit Lab

Names of students in your group:
Anshul Pardhi (ARP180012)

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.

Sample code given on e-learning

Dataset Used: Breast Cancer (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))
Google Colab Code Link: (https://colab.research.google.com/drive/1C2TdtTgWTAXhLr4taCrzihvFa4q-OLNr)
Scoring Parameter: precision_macro

ALGORITHM	BEST PARAMETERS	AVG PRECISION		AVG RECALL		AVG F1		ACCURACY SCORE
		MACRO	WEIGHTED	MACRO	WEIGHTED	MACRO	WEIGHTED	
Decision Tree	{'max_depth': 100, 'max_features': 15, 'min_samples_leaf': 3, 'min_samples_split': 2}	0.91	0.92	0.92	0.92	0.92	0.92	0.92
Neural Net	{'activation': 'identity', 'alpha': 0.1, 'learning_rate': 'invscaling', 'max_iter': 600}	0.93	0.94	0.94	0.94	0.94	0.94	0.94
Support Vector Machine	{'C': 10, 'kernel': 'linear', 'max_iter': -1, 'random_state': None}	0.96	0.97	0.97	0.97	0.97	0.97	0.97
Gaussian Naïve Bayes	{'priors': [0.3, 0.7]}	0.93	0.94	0.93	0.94	0.93	0.94	0.94
Logistic Regression	{'C': 2, 'fit_intercept': 'true', 'penalty': 'l1', 'tol': 0.01}	0.95	0.96	0.96	0.96	0.95	0.96	0.96
k-Nearest Neighbors	{'algorithm': 'auto', 'n_neighbors': 4, 'p': 1, 'weights': 'uniform'}	0.94	0.95	0.95	0.95	0.95	0.95	0.95
Bagging	{'max_features': 10, 'max_samples': 20, 'n_estimators': 5, 'random_state': 4}	0.93	0.94	0.94	0.94	0.93	0.94	0.94
Random Forest	{'criterion': 'entropy', 'max_depth': None, 'max_features': 10, 'n_estimators': 50}	0.96	0.95	0.94	0.95	0.95	0.95	0.95
AdaBoost Classifier	{'algorithm': 'SAMME', 'learning_rate': 1, 'n_estimators': 50, 'random_state': None}	0.98	0.98	0.97	0.98	0.98	0.98	0.98
Gradient Boosting Classifier	{'learning_rate': 1, 'loss': 'exponential', 'max_depth': 2, 'n_estimators': 5000}	0.96	0.96	0.95	0.96	0.96	0.96	0.96
XGBoost	{'booster': 'gbtree', 'learning_rate': 0.1, 'min_child_weight':	0.96	0.96	0.95	0.96	0.96	0.96	0.96

	1, 'n_estimators': 100}							

Scoring Parameter: recall_macro

ALGORITHM	BEST PARAMETERS	AVG PRECISION		AVG RECALL		AVG F1		ACCURACY SCORE
		MACRO	WEIGHTED	MACRO	WEIGHTED	MACRO	WEIGHTED	
Decision Tree	{'max_depth': 10000, 'max_features': 25, 'min_samples_leaf': 1, 'min_samples_split': 5}	0.90	0.91	0.90	0.91	0.90	0.91	0.91
Neural Net	{'activation': 'logistic', 'alpha': 0.01, 'learning_rate': 'invscaling', 'max_iter': 600}	0.92	0.93	0.93	0.93	0.92	0.93	0.93
Support Vector Machine	{'C': 10, 'kernel': 'linear', 'max_iter': -1, 'random_state': None}	0.96	0.97	0.97	0.97	0.97	0.97	0.97
Gaussian Naïve Bayes	{'priors': [0.3, 0.7]}	0.93	0.94	0.93	0.94	0.93	0.94	0.94
Logistic Regression	{'C': 2, 'fit_intercept': 'true', 'penalty': 'l1', 'tol': 0.01}	0.96	0.96	0.96	0.96	0.96	0.96	0.96
k-Nearest Neighbors	{'algorithm': 'auto', 'n_neighbors': 4, 'p': 1, 'weights': 'uniform'}	0.94	0.95	0.95	0.95	0.95	0.95	0.95
Bagging	{'max_features': 10, 'max_samples': 10, 'n_estimators': 100, 'random_state': None}	0.94	0.93	0.92	0.93	0.93	0.93	0.93
Random Forest	{'criterion': 'entropy', 'max_depth': None, 'max_features': 10, 'n_estimators': 20}	0.94	0.95	0.95	0.95	0.95	0.95	0.95
AdaBoost Classifier	{'algorithm': 'SAMME', 'learning_rate': 1, 'n_estimators': 50, 'random_state': None}	0.98	0.98	0.97	0.98	0.98	0.98	0.98
Gradient Boosting Classifier	{'learning_rate': 1, 'loss': 'exponential', 'max_depth': 2, 'n_estimators': 1000}	0.96	0.96	0.95	0.96	0.96	0.96	0.96
XGBoost	{'booster': 'gbtree', 'learning_rate': 1, 'min_child_weight':	0.97	0.97	0.96	0.96	0.96	0.96	0.96

	1, 'n_estimators': 50}							

AdaBoost Classifier performed the best, followed by Support Vector Machine, for the chosen parameters. It is so because in case of AdaBoost, many weak classifiers combine to form a strong classifier, stronger than other already stronger classifiers. For the case of SVMs, if a good decision boundary can be obtained, then the classifier correctly classifies the data items.

To improve the results, more parameter tuning is required, to try to get ALL the best possible parameters for a supervised learning algorithm. For this purpose, GridSearchCV comes in handy, but still, some part of parameter tuning is still required to obtain the best results.