Scikit learn grid search report

- ***Submitted by Shivam Gupta (SXG190040)***

Google Colab public link for the grid search algorithms:

https://colab.research.google.com/drive/1jnEM-DwJuPeGwGqde636bjkbdRdqLXpN

The UCI dataset which I used for Grid search is ***Breast Cancer Wisconsin (diagnostic)***:

Dataset link:
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Data Set Characteristics:

Number of Instances: **569**

Number of attributes: **10**

- Classification Diagnosis (M = malignant (**class label 2**), B = benign(**class label 4**))

Ten real-valued 11 features which are computed for each cell nucleus in this dataset:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

The table for grid search results is shown below:

Algorithm	Best Parameters	Average Precision	Average Recall	Average F1	Accuracy Score
Decision Tree	*{'max_depth': 6, 'max_features': 'log2', 'max_leaf_nodes': None, 'min_samples_leaf': 3, 'min_samples_split': 4}*	*0.96*	*0.96*	*0.96*	*0.96*
Neural Net	*{'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': 200, 'learning_rate': 'adaptive', 'max_iter': 500}*	*0.73*	*0.69*	*0.60*	*0.67*
Support Vector Machine	*{'C': 1000, 'gamma': 1e-05, 'kernel': 'rbf', 'max_iter': 200, 'random_state': 2}*	*0.62*	*0.66*	*0.62*	*0.66*
Gaussian Naive Bayes	*{'priors': None}*	*0.83*	*0.80*	*0.81*	*0.80*
Logistic Regression	*{'C': 1, 'max_iter': 100, 'penalty': 'l1', 'tol': 1e-05}*	*0.97*	*0.97*	*0.97*	*0.97*
k-Nearest Neighbors	*{'algorithm': 'auto', 'n_neighbors': 5, 'p': 1, 'weights': 'distance'}*	*0.63*	*0.63*	*0.63*	*0.63*
Bagging	*{'max_features': 2, 'max_samples': 2, 'n_estimators': 10, 'random_state': None}*	*0.89*	*0.87*	*0.86*	*0.87*
Random Forest	*{'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 10}*	*0.92*	*0.92*	*0.92*	*0.92*
AdaBoost Classifier	*{'algorithm': 'SAMME', 'learning_rate': 1.1, 'n_estimators': 50, 'random_state': None}*	*0.92*	*0.92*	*0.92*	*0.92*

Gradient Boosting Classifier	***{'learning_rate': 0.2, 'loss': 'exponential', 'max_depth': 5, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 80}***	***0.93***	***0.93***	***0.93***	***0.93***
XGBoost	***{'booster ': 'gbtree', 'learning_rate': 0.2, 'max_delta_step': 0, 'min_child_weight': 2, 'n_estimators': 100, 'seed': 0}***	***0.96***	***0.96***	***0.96***	***0.96***

Conclusion and results:

- In the end I would like to conclude that out of all the 11 Algorithms, grid search performed with ***Logistic regression*** gave the best results with ***97%*** accuracy score which pretty good and average precision and recall are also close to 1 i.e. 0.97, 0.97, 0.97 respectively.
- This result was expected because it is binary classification (M = malignant (class label 2), B = benign (class label 4)). All the feature values are easily able to fit into the logit function within that specific range and is easily able to classify well.
- The results could be improved more by providing more number of values to each parameter in the algorithm which is going to consider all the scenarios and will give the best accuracy by fetching the best parameters.
- Neural network is performing the worst on this datasets for activation "ReLu" some of the neurons are getting zero values so not getting considered and there are only 11 attributes which are very less.