SAT5114 – BREAST CANCER DIAGNOSIS USING ML CLASSIFIERS

GROUP MEMBERS – VISHWAJEETH BALAJI, NAGENDRA MADI REDDY

Team Contribution:

Vishwajeeth Balaji:

- Responsible for loading and cleaning the dataset.
- Conducted initial data exploration and visualization using Matplotlib and Seaborn.
- Contributed to the implementation and testing of classifiers.

Nagendra Madi Reddy:

- Focused on data preprocessing, including data splitting and scaling.
- Involved in the implementation and testing of classifiers.
- Led the hyperparameter tuning using GridSearchCV.
- Collaborated in performance evaluation, reporting key metrics for all classifiers.

Introduction

The main aim of the project is to analyse the data set for predicting the breast cancer using different classifiers. To achieve this the data set was screened for null values, scaled, split the data into test and train and further processed to choose the best hyperparameters to check the performance of the evaluation metrics of the classifiers.

Methodology

The project is initiated using a data set from UCI(mentioned in the canvas assignment) and this data set is pulled into python programming with all the necessary packages.

Numpy – Performs numerical operations.

Pandas – Performs data processing.

Matplotlib and Seaborn – Performs data visualization

Train_Test_split performs – Splits the data into training and testing for model evaluation for the unseen data.

Cross_val_score – Splits the data into multiple subsets for checking the training data subsets to its complementary subsets.

GridSearchCV- Performs a grid search for the optimal parameters in a cross validation.

Standard_Scalar – This library helps to remove the mean values and helps in scaling to unit variance.

Label_Encoder – Coverts the categorical values into numerical values.

Random_Forest_Classifier – This classifier fits the data into number of decision tree classifiers upon combines into sub samples of the datasets and helps in finding accuracy and also helps in finding the over-fitting issues.

Logistic-Regression – This classifier performs a linear regression and predicts the binary outcome.

Decision_Tree_Classifier – This classifier comes under supervised learning of a model and mostly used for classification.

Accuracy_Score – This function provides the accuracy of a model by combining both predicted and true labels.

Classification_Report – This function provides a detailed information of the model performance using different attributes such as F1 score, recall, accuracy, and precision for each class labels.

Code description

Step 1:

The csv file that contains the dataset is loaded into the environment and using **df.head** the initial rows are displayed.

Step 2:

Using **df.isna().sum()** all the null values from the columns are pulled out. After this **df.shape** is used to get return the reshaped data and drop the null values using **df=df.dropna(axis=1).**

Step 3:

Checking the data shape with **df.shape** after removing the null values. And then data is described using **df.describe**.

Step 4:

Now we reached at point where the dataset is clear with no null values, we use **diagnosis_count – df.['diagnosis].value_counts()** and **print diagnosis_counts** for both malignant M & Benign B cells.

Step 5:

Using **sns.countplot** a bar plot is plotted for both type of cells.

Step 6:

Using Label_Encoder the categorical values B and M are converted into numerical values as 0 & 1.

Step 7:

sns.pairplot is used to provide an output of scatter plot between the datasets.

Step 8:

Now a correlation of the data is shown as an output using **df.iloc** function, and further correlation visualization is performed to plot heatmap / correlation matrix using **sns.heatmap**.

Step 9:

Now the data is split into dependent (X) & independent (Y) using **df.iloc** function. Further the data is split into test and train using **train_test_split** and scaled using StandardScalar function.

Implementation & testing of datasets using classifiers

Step 10:

The data is then used to perform the model algorithms of different classifiers such as Random Forest, Decision Tree and Logistic Regression to get the accuracy. K-NN cross validation is also performed with test data for checking the accuracy difference.

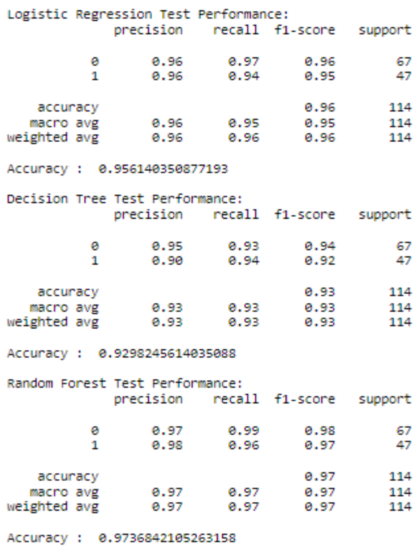
Step 11:

Then the grid search hyperparameter tuning is done for the classifiers to optimize the performance and choose the best hyperparameters.

Step 12:

Then finally test data performance is reported for all the classifiers with different attributes such as f1 score , accuracy , precision & recall.

Results:



```
Logistic Regression Test Performance:
              precision    recall  f1-score   support

           0       0.96      0.97      0.96        67
           1       0.96      0.94      0.95        47

    accuracy                           0.96       114
   macro avg       0.96      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114

Accuracy :  0.956140350877193

Decision Tree Test Performance:
              precision    recall  f1-score   support

           0       0.95      0.93      0.94        67
           1       0.90      0.94      0.92        47

    accuracy                           0.93       114
   macro avg       0.93      0.93      0.93       114
weighted avg       0.93      0.93      0.93       114

Accuracy :  0.9298245614035088

Random Forest Test Performance:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98        67
           1       0.98      0.96      0.97        47

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114

Accuracy :  0.9736842105263158
```

Conclusion

The random forest outperforms all other classifiers with a better accuracy of 0.97. So as a conclusion the random forest classifier is best suited for the diagnosis of the breast cancer.

GITHUB LINK:

https://github.com/Vishwajeeth2000/Breast-Cancer-Diagnosis-using-Machine-Learning-Classifiers