MLBA_Assignment_2_Readme
Created by Group 66
Members : Akhil P Dominic, Rajith Ramachandran, Sarvani Gupta

Run the code:
Usage: python <filename.py> --files <full_testdata_filename.csv> <full_traindata_filename>

Example : python mlba_cancer_pred.py --files cancer_dataset/kaggle_train.csv cancer_dataset/kaggle_test.csv

Aim: Classify high and low risk cancer patients.
Provided data: test, train datasets, and sample submission file

Methodology: We have tried using various machine learning techniques like random forest, gradient boost etc, on the given train dataset.
The maximum performance was obtained for the random forest classifier, which has given us an accuracy of around ~81.6 percent. We have also used grid searches to find the best hyperparameters and avoid overfitting.

Steps:
● Data importing and preprocessing
We had the kaggle_train.csv. We used the pandas library to read the CSV file.

```
8
9  #Downloading the given packages
10 '''import subprocess
11 packages_to_install = ['pandas', 'numpy','sklearn','imbalanced-learn']
12 for package in packages_to_install:
13   subprocess.check_call(['pip', 'install', package])'''
14
15 #importing the necessary packages
16 import argparse
17 import pandas as pd
18 import numpy as np
19 from sklearn.model_selection import train_test_split
20 from sklearn.ensemble import RandomForestClassifier
21 from sklearn.metrics import roc_auc_score
22 from sklearn.preprocessing import StandardScaler
23 from sklearn.feature_selection import RFE
24 import csv
25 #Parsing the data from command line
26 parser=argparse.ArgumentParser()
27 parser.add_argument("--files",nargs='+',help="Train and test files")
28 args = parser.parse_args()
29 args=args.files
30 train_csv=args[0]
31 test_csv=args[1]
32 #reading the training csv file
33 train_data = pd.read_csv(train_csv)
```

● Test Train Split

We then applied the test train split function to split the dataset into a test and train set. We have taken the value of the hyperparameter as 100 and random state as 42. We are splitting the dataset into the ratio 80:20

```
30
31  #Taking in the labels from the train_data
32  labels = train_data["Labels"]
33
34
35  scaler = StandardScaler()
36  #Applying scaling to the training dataset
37  X_train = scaler.fit_transform(X_train)
38
```

● Creating the model

Recursive feature elimination(RFE):

RFE is popular because it is easy to configure and use and because it is effective at selecting those features (columns) in a training dataset that are more or most relevant in predicting the target variable. The RFE technique would start with the original number of features and would select features by recursively eliminating less important features.

```
39  #Using recursive feature elimination technique to find the features which are less important and remove them so that w
40  recursive_fe = RFE(estimator=RandomForestClassifier(n_estimators=100, random_state=42), n_features_to_select=25)
41  X_train = recursive_fe.fit_transform(X_train, labels)
42
43  #Splitting the dataset into training and testing dataset in the ratio 80:20
44  X_train, X_val, y_train, y_val = train_test_split(X_train, labels, test_size=0.2, random_state=42)
45
46  #declaring the random forest classifier with 100 estimators
47  rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
48  rf_classifier.fit(X_train, y_train)
49
50  #predicting the values from the testing dataset inorder to test AUC accuracy
51  y_pred = rf_classifier.predict_proba(X_val)[:, 1]
52
53  auc_score = roc_auc_score(y_val, y_pred)
54  print(f"Accuracy: {auc_score}")
55
56  #Importing the testing dataset
57  test_data = pd.read_csv("cancer_dataset/kaggle_test.csv")
58
59  #Taking in and removing the ID column from testing dataset
60  test_ID = test_data['ID']
61  test_data = test_data.drop("ID", axis=1)
62  test_data = scaler.transform(test_data)
63
64  test_data = recursive_fe.transform(test_data)
```

● Final Output

We applied the rf classifier model to the test dataset and have predicted the probability of the person having cancer . We saved the file in the output folder in the attached folder. We consider the AUC ROC score and hence we have found out the probability of a particular row belonging to cancer or not.

```
65
66  #Predicting the final output probability
67  final_pred =rf_classifier.predict_proba(test_data)[:, 1]
68
69  #Writing the output predictions onto the file ./output_rf.csv
70  with open('./output_rf.csv', mode='w', newline='') as file:
71    writer = csv.writer(file)
72    writer.writerow(['ID', 'Labels'])
73    for i in range(0,len(test_ID)):
74      writer.writerow([test_ID[i],final_pred[i]])
75
```

```
~/Desktop/MLBA/Assignment2 python mlba_cancer_pred.py --files cancer_dataset/kaggle_train.csv cancer_dataset/kaggle_test.csv
Accuracy: 0.8169354838709678
```

Directory structure: