assignment_6

May 26, 2018

1 Team Members

1.1 RaviKiran Bhat

1.2 Rubanraj Ravichandran

1.3 Mohammad Wasil

1.4 Ramesh Kumar

```
In [1]: from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
        from IPython.display import display
        from sklearn import metrics
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import accuracy_score
        from sklearn.metrics import confusion_matrix
        import pandas as pd
```

```
In [2]: df = pd.read_csv('parkinsons.data')
        print(df.head())
```

```
  parkinsons.dataname  MDVP:Fo(Hz)  MDVP:Fhi(Hz)  MDVP:Flo(Hz)  \
0      phon_R01_S01_1      119.992       157.302        74.997
1      phon_R01_S01_2      122.400       148.650       113.819
2      phon_R01_S01_3      116.682       131.111       111.555
3      phon_R01_S01_4      116.676       137.871       111.366
4      phon_R01_S01_5      116.014       141.781       110.655

   MDVP:Jitter(%)  MDVP:Jitter(Abs)  MDVP:RAP  MDVP:PPQ  Jitter:DDP  \
0         0.00784           0.00007   0.00370   0.00554     0.01109
1         0.00968           0.00008   0.00465   0.00696     0.01394
2         0.01050           0.00009   0.00544   0.00781     0.01633
3         0.00997           0.00009   0.00502   0.00698     0.01505
4         0.01284           0.00011   0.00655   0.00908     0.01966

   MDVP:Shimmer    ...     Shimmer:DDA      NHR     HNR  status      RPDE  \
0       0.04374    ...         0.06545  0.02211  21.033       1  0.414783
1       0.06134    ...         0.09403  0.01929  19.085       1  0.458359
2       0.05233    ...         0.08270  0.01309  20.651       1  0.429895
```

```
3       0.05492    ...         0.08771  0.01353  20.644       1  0.434969
4       0.06425    ...         0.10470  0.01767  19.649       1  0.417356

        DFA   spread1   spread2        D2       PPE
0  0.815285 -4.813031  0.266482  2.301442  0.284654
1  0.819521 -4.075192  0.335590  2.486855  0.368674
2  0.825288 -4.443179  0.311173  2.342259  0.332634
3  0.819235 -4.117501  0.334147  2.405554  0.368975
4  0.823484 -3.747787  0.234513  2.332180  0.410335

[5 rows x 24 columns]
```

```
In [3]: X = df.drop('status', axis=1)
        X = X.drop('parkinsons.dataname', axis=1)
        y = df['status']
```

```
In [7]: X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)

        # The number of estimators (n_estimators) determines how dense our
        #decision forest is and the random_state is given for reproducibility.

        random_forest = RandomForestClassifier(n_estimators=30, max_depth=10, random_state=1)
        random_forest.fit(X_train, y_train)

        y_predict = random_forest.predict(X_test)
        accuracy_score(y_test, y_predict)
        pd.DataFrame( confusion_matrix(y_test, y_predict),
                     columns=['Predicted Healthy', 'Predicted Parkinsons'],
                     index=['True Healthy', 'True Parkinsons'] )
```

```
Out[7]:                   Predicted Healthy  Predicted Parkinsons
        True Healthy                     11                     1
        True Parkinsons                   2                    35
```

It shows that the model can classify the given dataset with less number of misclasification (e.g. the model incorrectly predicts 1 positive class (false positive) and 2 negative class(false negative)). Overall the number of true positive and true negative (correctly classified) are considerably high.