Machine Learning with Python

Mini Project

Prepared by Gurveer Singh Dhillon

Problem Statement

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

Libraries and IDE used

The program was written and performed on Google Colab Notebook

The Libraries used are as follows

- numpy
- pandas
- matplotlib.pyplot
- seaborn
- sklearn
- train_test_split from sklearn.model_selection
- StandardScaler from sklearn.preprocessing
- LogisticRegression from sklearn.linear_model
- confusion_matrix, classification_report, accuracy_score from sklearn.metrics
- KNeighborsClassifier from sklearn.neighbors

Methodology

First of all we study the data given in the problem statement, after which we open our Google Colab Notebook to proceed with the problem and solve it.

In our Notebook, we first import the aforementioned libraries which will be used in the program. Then we load our given dataset on our notebook. The file is then read and we print the first 5 rows of the Dataframe. describe() function is then used to calculate statistical data of the DataFrame. info() function is used to give a summary of the DataFrame. isnull().sum() is used to get the count of null values. The zeroes values in the columns are replaced to NaN values and then impute accordingly with median. These steps comprise the Exploratory Data Analysis.

Then we begin the Preparation of data for Models. For the preparation we begin by dividing the data into 'Train' and 'Test'. The data which is put under the section 'Train' is used for training of our models while the data under 'Test' is used to perform tests on our model. The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. We have divided 'Train' and 'Test' in percentages of 70 and 30. random_state=42 is used as the parameter to produce the same results across a different run.

After preparing the data, we start with Classification Models. In this project Linear Regression and K Nearest Neighbor (KNN) models are used. The other necessary packages are imported from their respective libraries. Then we create the model for Linear Regression and train it using the data in 'Train'. Then we make a confusion matrix between the predicted values from our model and the data from 'Test' and print out the results. Then we try out the KNN method, where we first use 5 as the value for n_neighbors or K and train the model using 'Train' data. The next step involves the creation of confusion matrix between the predicted values from our model and the data from 'Test' and print out the results. After getting the results from this method we change the previous K value from 5 to 7 and then

repeat the same process of training and testing our model and then get the fresh results.

Model Codes

The confusion matrix of Linear Regression is here as follows

```
Confusion Matrix:
 [[127  24]
 [ 32  48]]


Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.84      0.82       151
           1       0.67      0.60      0.63        80

    accuracy                           0.76       231
   macro avg       0.73      0.72      0.73       231
weighted avg       0.75      0.76      0.75       231
```

The confusion matrix of KNN with K=5 is here as follows

```
Confusion Matrix:
 [[121  30]
 [ 21  59]]


Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.80      0.83       151
           1       0.66      0.74      0.70        80

    accuracy                           0.78       231
   macro avg       0.76      0.77      0.76       231
weighted avg       0.79      0.78      0.78       231
```

The confusion matrix of KNN with K=7 is here as follows

```
Confusion Matrix:
 [[126  25]
 [ 25  55]]


Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.83      0.83       151
           1       0.69      0.69      0.69        80

    accuracy                           0.78       231
   macro avg       0.76      0.76      0.76       231
weighted avg       0.78      0.78      0.78       231
```

Conclusion

Here I conclude my findings that the Linear regression method gives us the accuracy of 75.75% while the KNN method with K=7 gave us an accuracy of 78.35%. Hence the KNN method with K=7 is more accurate than Linear Regression method.