Name: Jonas de Deus Guterres
Class: IF-42-INT
Subject: Artificial Intelligence Observation Parallel Assignment 3

Report

This assignment is about the k-nearest neighbor (KNN) which is built by me to estimate the accuracy of Pima India Diabetes Dataset (PIDD) in file diabetes.csv. There are containing 768 data in total. The language that I used to build is python in google colab. Firstly, we should upload the csv file for loading the data. I would like to explain the strategy and the process of KNN to find the best k value with its accuracy in the following steps:

1. **Selection of the Distance Function**
   I select the Euclidean Distance to calculate the test and training data because using equation along with scaled data in order to avoid the effect of units. Firstly, I split the label data for both train set and test set before the calculation. Afterwards, I start to calculate the distance of each data of test set to all train data until the iteration of the length of test set is terminate. The formula of Euclidean distance that used is:

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

   Description: p and q = two points in Euclidean n-space, qi and p_i = Euclidean vectors, starting from the origin of the space (initial point), n = n-space.

2. **Data Preprocessing Techniques**
   The technique that I used is split the data of testing set and training set. The testing is 20% to calculate with the training. Beforehand, I load the data form the database.csv file and the total of the column that going to do the calculation is 8 column of feature and the last column named outcome is for labelling data in the function preprocessed.

3. **Feature Engineering Techniques**
   In my strategy I used standard scalar to standardize the feature column by subtracting the mean and the scaling to unit variance. So the half value of the feature data will be in between -1 and 1. Thus, in my function standard_scaler using the library from the sklearn.preprocessing to calculate the mean and standard deviation in standard scalar.

$$Standardization: z = \frac{x-\mu}{\sigma} \qquad Mean: \mu = \frac{1}{N}\sum_{i=1}^{N}(x_i) \qquad STD: \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

   For the normal calculation of standard scalar will use the formula above to get the standardization data

4. **Strategies for using the KNN algorithm**
   The function that I build for KNN algorithm is classification function. The first technique is concatenating the calculated data from Euclidean distance with the label data that we have split before. Then, do the ascending sort for every calculated data to find the nearest neighbor regarding to the K value that we used such as k=1,k=2…k=30. For the best to find the nearest is odd number because for even number of k is difficult to choose if the label of 1 and 0 is equal.
   Let's assume the k = 3. The first step is we opt the three smallest data that we have sorted. Then we make two new variable to count the total of 1s and 0s to decide which label is

containing many number of diabetes (1) and or non-diabetes (0). If the result is diabetes (1) or non-diabetes (0), then we compare with the label data of the test. If they have same value then we sum it then stored in the new variable of sum. Do it continuously until the final iteration, then we add total diabetes and total non-diabetes divided by total data of test. Then we got the result of accuracy by multiply to 100.

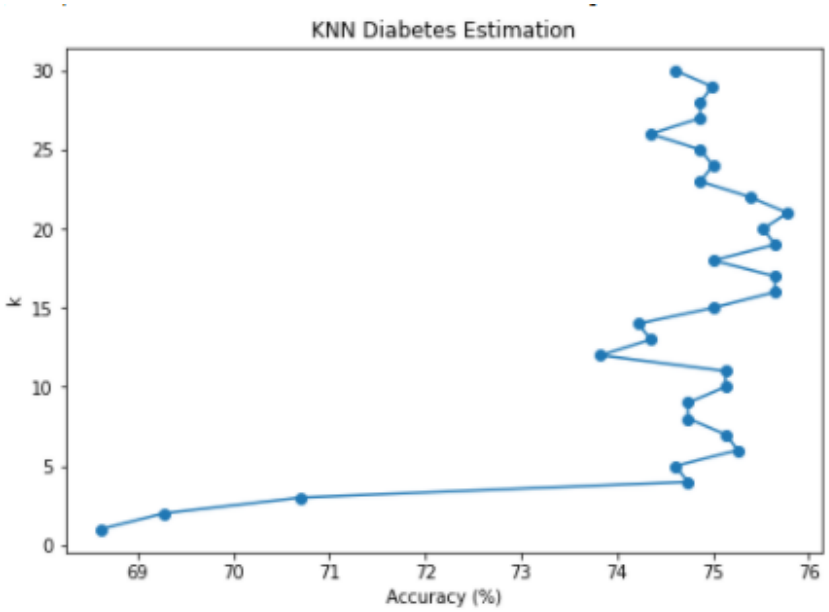
$$accuracy = \frac{TD + TN}{TD + TN + FD + FN}$$

Description:
TD=True Diabetes, TN=True Non-Diabetes, FD=False Diabetes, FN= False Non-Diabetes

5. **Selection of the Best *k* value for the Selection and Estimation Processes of the KNN Model**

For finding the estimation and selection of best k, first we use 5 fold cross validation to calculate every fold and every k to get all accuracy. After we do the classification process, we take the data which has already split in the question to do the cross validation. In my function of cross validation I make 5 new variables to save every fold percentages. Then, calculated the average of the 5 folds.

Afterwards we do the estimation of the highest average among the k that given. In my case I got k =21 as the best k with accuracy 76%. The following is the graphic of estimation.



The parameter that I used are k = 30. As we can see in the graph, the highest point is 76% in the k of 21.

Name: Jonas de Deus Guterres
Class: IF-42-INT
Subject: Artificial Intelligence Observation Parallel Assignment 3

6. Output

```
k: 1 accuracy:  68.61386979034037
k: 2 accuracy:  69.27170868347339
k: 3 accuracy:  70.69858246328835
k: 4 accuracy:  74.73983532807063
k: 5 accuracy:  74.60656990068755
k: 6 accuracy:  75.25931584755114
k: 7 accuracy:  75.13284101519395
k: 8 accuracy:  74.74068415244886
k: 9 accuracy:  74.7415329768271
k: 10 accuracy:  75.13114336643748
k: 11 accuracy:  75.13199219081572
k: 12 accuracy:  73.8307444189797
k: 13 accuracy:  74.35022493846023
k: 14 accuracy:  74.22205245734658
k: 15 accuracy:  74.99957558781088
k: 16 accuracy:  75.64892623716153
k: 17 accuracy:  75.65062388591801
k: 18 accuracy:  74.99957558781088
k: 19 accuracy:  75.65147271029625
k: 20 accuracy:  75.5216025804261
k: 21 accuracy:  75.78049401578814
k: 22 accuracy:  75.39173245055598
k: 23 accuracy:  74.86970545794075
k: 24 accuracy:  75.00212206094558
k: 25 accuracy:  74.87140310669722
k: 26 accuracy:  74.35107376283847
k: 27 accuracy:  74.86885663356252
k: 28 accuracy:  74.86800780918428
k: 29 accuracy:  74.99702911467617
k: 30 accuracy:  74.60741872506578
Best k value of kNN learning is: 21 with accuracy: 76 %
```