**DATA MINING**

**PROJECT 1**

**REPORT**

Name: Rehana Devani

UTA ID 1001100807

NET ID: rxd0807

Two python scripts are submitted.

Knn\_norm.py script implements KNN algorithm of normalized datasets.

KnnComplete.py script implements KNN algorithm of un-normalized datasets.

Report contains evaluation of both the scripts for different values of K-Fold and KNN.

K-Fold value is hard coded in both the scripts in the ‘hardk’ variable on line 155.

The following steps should be followed to run the code:

1. Give the path of dataset you want to run based on its location on your computer on lines 128 and 138 in main method.
2. The CSV file label (class) path should be given on line 128 and the CSV file dataset path should be given on line 138.
3. Both the files are merged in the program.
4. To run the third dataset(seeds), its CSV files for label and dataset are attached with this folder.
5. Then go to the terminal and give the path where the python script is stored.
6. Use the following command to run the python script

**python knn.py**

The output shows the accuracy of each fold and the overall accuracy for the value entered bu user for KNN.

* **What is the classification problem you are solving?**

The classification problem being solved is K- Nearest Neighbors.

KNN Algorithm divides

**KNN**

* **What data do you use? What is the dimension? What are the attributes? Where is the data from? Cite any reference.**
* We are using 3 datasets:
* 1. ATT dataset
* 2. Iris dataset
* 3. Seeds dataset
* ATT and Iris dataset was provided and the link of Seeds dataset is:
* https://archive.ics.uci.edu/ml/machine-learning-databases/00236/
* Seeds dataset was downloaded from one of the two links provided.
* Seeds dataset was in .txt format. It was later saved as csv files which are also
* submitted for ease.
* References:
* http://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/
* **What algorithms do you use? Describe the algorithm briefly.**

The algorithm used is K- Nearest Neighbor.

I have merged two CSV files containing data and labels. I divided the dataset based on the class. I then divided each class data into k folds where k value is hard coded as 5 and 10 respectively.

One part of the data is used for test and remaining part is used for training set.

Then I calculate the euclidean distance of first value of testing set with all values of training set, and then second value of testing set with all values of training set.

I am sorting the list based on euclidean distance in ascending order.

Now majority voting takes place based on the k value which user initially provided. Accuracy is calculated based on the actual and predicted values.

KNN algorithm works as follows:

K Nearest Algorithm divides the dataset into training and testing set by taking one part as testing and remaining parts as training. Euclidean distance is calculated from each point of testing to every point in training and so on. Then for each value of testing data, all the values are sorted in ascending order and then majority voting takes place. Accuracy can be found from the actual and predicted values.

* **How will you evaluate your result?**

Result was evaluated by taking k-fold values as 5, 10 and then doing k-fold of k =5, k=10

with each k value of KNN. Average accuracy of each fold of KNN is observed and this step

is repeated for different k values of KNN.

The results are evaluated and shown in the following tables

IRIS DATA SET EVALUATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KNN | KFOLD = 5 | | KFOLD = 10 | |
|  | NORMALIZED | UNNORMALIZED | NORMALIZED | UNNORMALIZED |
| K=7 | 94.6 | 94.0% | 94.6 | 94.6% |
| K=8 | 94.6 | 94.0% | 94.6 | 94.6% |
| K=9 | 95.3 | 95.3% | 95.3 | 94.6% |
| K=10 | 95.9 | 95.3% | 95.3 | 95.3% |
| K=11 | 95.9 | 95.3% | 96.0 | 95.3% |
| K=12 | 95.9 | 94.6% | 95.3 | 94.6% |
| K=13 | 95.3 | 94.6% | 96.0 | 95.3% |
| K=14 | 95.3 | 94.0% | 96.6 | 94.6% |
| K=15 | 95.9 | 94.6% | 96.0 | 95.3% |
| K=16 | 94.0 | 94.6% | 96.0 | 94.0% |
| K=17 | 95.9 | 94.6% | 95.3 | 94.0% |
| K=18 | 95.9 | 94.0% | 96.0 | 93.3% |
| K=19 | 95.3 | 94.0% | 96.0 | 94.0% |
| K=20 | 95.3 | 94.0% | 96.0 | 94.6% |
| K=21 | 94.6 | 94.6% | 96.0 | 94.0% |
| K=22 | 95.3 | 94.0% | 96.0 | 94.6% |
| K=23 | 94.6 | 93.3% | 95.3 | 94.0% |
| K=24 | 94.6 | 94.0% | 96.0 | 94.0% |
| K=25 | 93.3 | 94.0% | 94.6 | 93.3% |
| K=30 | 93.3 | 92.6% | 93.3 | 93.3% |
| K=40 | 92.0 | 90.6% | 92.0 | 92.6% |
| K=50 | 90.0 | 86.6% | 90.0 | 89.3% |
| K=60 | 88.6 | 86.0% | 88.6 | 86.6% |
|  |  |  |  |  |

ATT DATA SET EVALUATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KNN | KFOLD = 5 | | KFOLD = 10 | |
|  | NORMALIZED | UNNORMALIZED | NORMALIZED | UNNORMALIZED |
| K=1 | 98.0% | 97.75% | 98.25 | 98.0% |
| K=2 | 96.5% | 94.5% | 97.25 | 96.25% |
| K=3 | 95.25% | 94.5% | 95.75 | 95.5% |
| K=4 | 92.0% | 91.25% | 94.75 | 94.0% |
| K=5 | 91.25% | 90.5% | 94.25 | 91.25% |
| K=6 | 89.5% | 87.5% | 92.75 | 90.5% |
| K=7 | 87.25% | 86.5% | 90.5 | 90.0% |
| K=8 | 86.25 | 85.25% | 88.75 | 88.5% |
| K=9 | 83.5 | 82.0 | 85.5 | 85.5% |
| K=10 | 81.25 | 81.25 | 83.5 | 83.5% |
| K=11 | 79.0 | 79.25 | 82.25 | 83.0% |
| K=12 | 77.0 | 79.25 | 79.5 | 81.5% |
| K=13 | 77.0 | 76.25 | 79.0 | 79.25% |
| K=14 | 74.75 | 74.0 | 77.25 | 77.5% |
| K=15 | 73.25 | 72.75 | 77.25 | 76.75% |
| K=16 | 73.25 | 73.0 | 75.0 | 75.25% |
| K=17 | 73.0 | 71.75 | 74.75 | 73.0% |
| K=18 | 73.25 | 71.0 | 74.5 | 73.75% |
| K=19 | 73.5 | 70.25 | 73.5 | 72.5% |
| K=20 | 72.25 | 68.75 | 72.5 | 71.75% |
| K=21 | 70.0 | 68.25 | 72.0 | 70.75 |
| K=22 | 68.25 | 66.75 | 71.75 | 70.75 |
| K=23 | 67.0 | 65.0 | 70.75 | 69.75 |
| K=24 | 66.0 | 63.25 | 69.0 | 67.5 |
| K=25 | 64.5 | 62.0 | 67.25 | 66.5 |
| K=30 | 62.25 | 61.0 | 64.75 | 63.5 |
| K=40 | 55.75 | 53.25 | 58.25 | 56.5 |
| K=50 | 47.25 | 50.0 | 54.0 | 51.25 |
| K=60 | 40.5 | 43.75 | 46.0 | 50.25 |
|  |  |  |  |  |

The accuracy is increased when the data is normalized. The above tables shows that the accuracy is changed with respect to the parameters and this is

SEED DATA SET EVALUATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KNN | KFOLD = 5 | | KFOLD = 10 | |
|  | NORMALIZED | UNNORMALIZED | NORMALIZED | UNNORMALIZED |
| K=7 | 89.5 | 89.0 | 89.5 | 88.0 |
| K=8 | 88.0 | 89.0 | 89.5 | 88.0 |
| K=9 | 90.0 | 89.5 | 90.0 | 89.0 |
| K=10 | 88.5 | 89.0 | 89.5 | 88.5 |
| K=11 | 89.5 | 89.5 | 89.5 | 89.5 |
| K=12 | 91.4 | 89.5 | 89.5 | 88.0 |
| K=13 | 90.0 | 88.5 | 89.9 | 88.0 |
| K=14 | 90.0 | 89.0 | 90.4 | 88.0 |
| K=15 | 89.0 | 89.5 | 89.5 | 89.9 |
| K=16 | 89.5 | 90.0 | 89.5 | 89.0 |
| K=17 | 89.5 | 90.0 | 89.5 | 89.5 |
| K=18 | 89.5 | 90.0 | 89.5 | 89.5 |
| K=19 | 90.0 | 90.0 | 89.0 | 89.5 |
| K=20 | 89.0 | 89.5 | 89.5 | 89.5 |
| K=21 | 88.5 | 89.5 | 89.0 | 89.5 |
| K=22 | 89.0 | 90.0 | 89.0 | 88.5 |
| K=23 | 89.5 | 90.0 | 89.0 | 90.0 |
| K=24 | 89.5 | 90.0 | 89.0 | 90.0 |
| K=25 | 89.0 | 90.0 | 88.5 | 90.0 |
| K=30 | 88.0 | 89.5 | 89.5 | 89.9 |
| K=40 | 88.5 | 89.5 | 89.5 | 88.5 |
| K=50 | 84.2 | 89.9 | 86.6 | 89.0 |
| K=60 | 84.7 | 89.9 | 84.2 | 89.5 |
|  |  |  |  |  |

The accuracy increases for normalized data. The above table shows the change of accuracy for different parameters and this is done for all the parameters. KNN algorithm was implemented and K fold code was written. Python was used for KNN implementation.