

Report KNN and holdout Assignment

July 2, 2022

1 Introduction

In this assignment I have implemented Knn with and without MinMax and Holdout of 20 percent for Test and 80 percent for Training, the focus of this assignment is the analysis of Performance, Mean Accuracy, and the code produced by this assignment.

The code of this assignment for the generation of results was made using Golang, a language well known to have better performance than Python in most cases and the analysis of Results used Python.

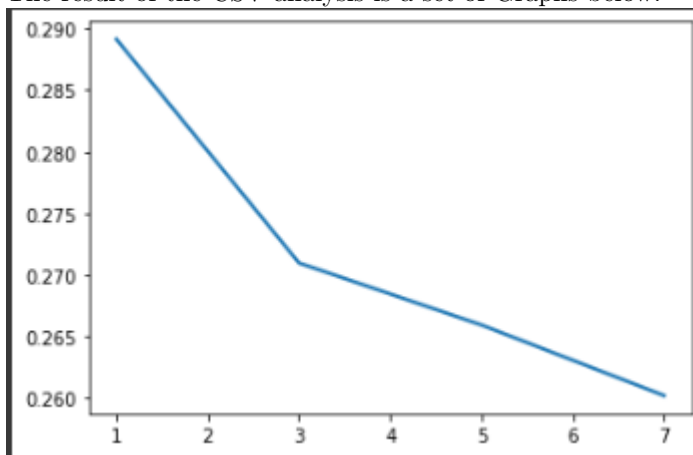
2 Result of Code Execution

| colx | coly | k | totalright | Performance | UseMinMax | Distance |
|------|------|---|------------|-------------|-----------|----------------|
| 0 | 1 | 2 | 1 | 27 | 31095691 | True euclidian |

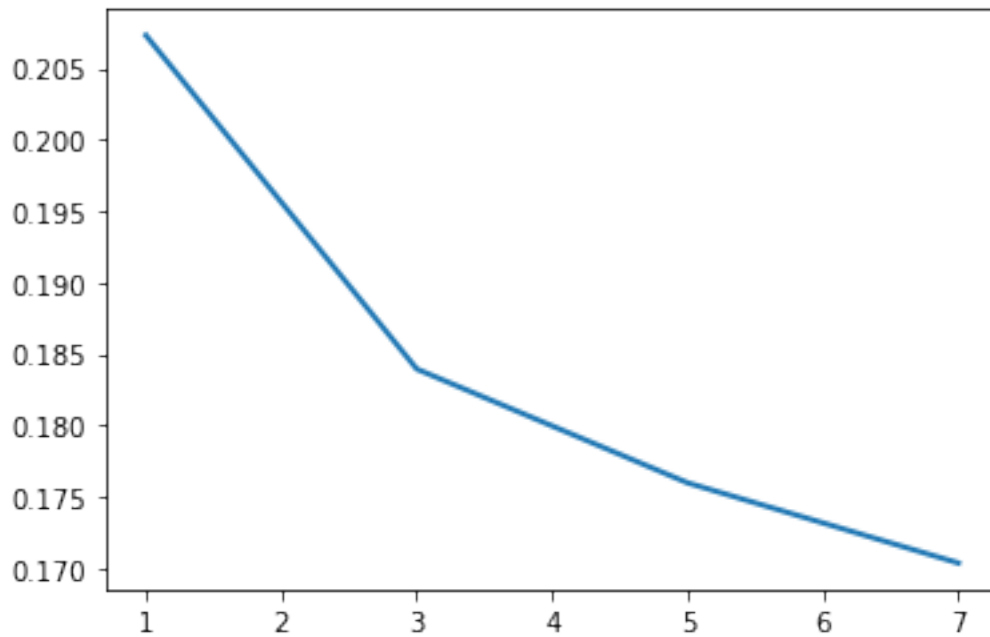
The Result of the Golang code execution was a Json that, after being transformed with python to a csv has the respective columns: colx and coly , the index of the attribute used as x and y respectively, k value for knn algorithm, Performance, Total Performance of Knn algorithm using every point of x and y on Test Data for colx and coly,Scorem, average Score of all points Score using colx and coly as Attribute, and UseMinMax and Distance, True or False for the case of UseMinMax and "Euclidian" and "Man" in the case of Distance.

3 Csv analysis

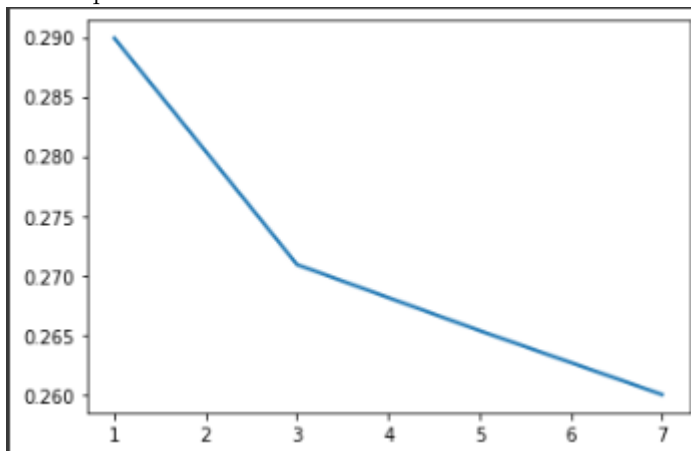
The result of the CSV analysis is a set of Graphs below.



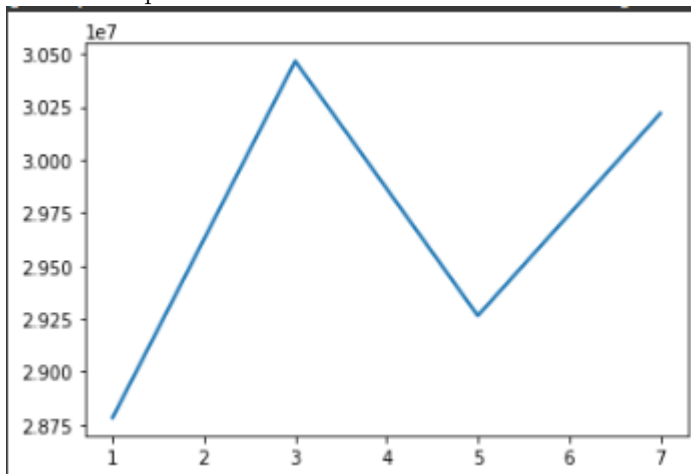
K on X and average accuracy of all combination of colx and coly with Euclidean distance on Y and MinMax equals True



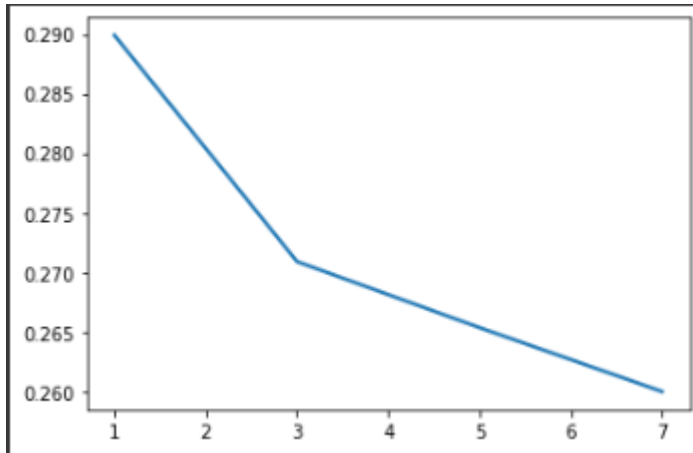
K on X and average accuracy of all combination of colx and coly with Euclidean distance on Y and MinMax equals False



K on X and average accuracy of all combination of colx and coly with Manhattan distance on Y and MinMax equals True



K on X and average Performance in nanoseconds of all combination of colx and coly with Euclidean distance on Y and MinMax equals True



K on X and average Performance in nanoseconds of all combination of colx and coly with Manhattan distance on Y and MinMax equals True

4 Conclusion of Csv analysis

The result was that a $k=1$ gave a better average accuracy if you calculate the average of every accuracy on the CVS with and without Euclidian Distance and with MinMax. There was a similar performance for every k used, the logic is that the algorithm uses a sort that independent of k gave similar performance, a better algorithm for KNN can be made but escape the scope of this assignment.

5 Code Analysis

The code was divided into three folders with one Golang file each and the Main.go File,knn for Knn Algorithm ,MinMax for MinMax algorithms and stratifiedSamplingTestTrainingSplit for stratified sampling algorithm.It has been noted that Golang was more challenging to develop than python, but it was worth it because of the gain in performance.

6 Conclusions

Performance with Manhattan distance was almost 2 times faster than Euclidean Distance, k equal 1 gave a better result in average and MinMax gave a better Accuracy.

References