­­ CSE-6363-002 Machine Learning

Spring 2024

Programming Assignment 1

A blue and black logo

Description automatically generated

Submitted By:

Keerthana Reddy Gudi

1002160234

Instructed By:

Jesus A. Gonzalez

Teaching Assistant:

Priyanka Pradeep Desai

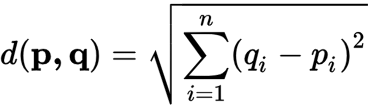
In this assignment, I am building the K Nearest Neighbors Algorithm from Scratch with K-Fold cross validation and tested this onto 3 datasets given i.e., Hayes Roth, Car Dataset and Breast Cancer Dataset. I used 4 different types of distance metrics Euclidean, Manhattan, Hamming and Cosine Similarity in KNN and compared the accuracies. I also built a scikit-learn model to test these 3 datasets. And performed hypothesis testing for the above accuracies.

Apart from the 3-distance metrics mentioned as extensions in reference link, I used another distance metric i.e., cosine similarity. I studied cosine similarity in Data Mining Course and thought that it would fit here. So, I extended KNN using cosine similarity analyzing the accuracies.

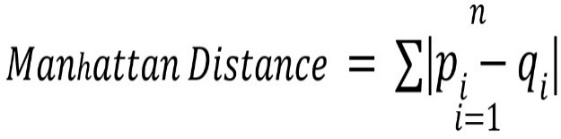
The code for this assignment is originally written in the Google Collab notebook. For execution we need to upload datasets before running.

**Distance Functions:**

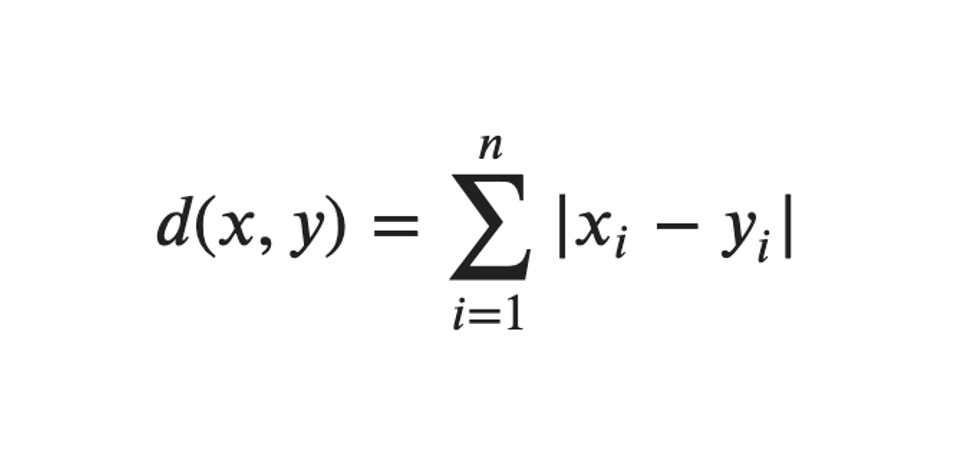
**Euclidean Distance:**



**Manhattan Distance:**



**Hamming Distance:**



**Cosine Similarity:**

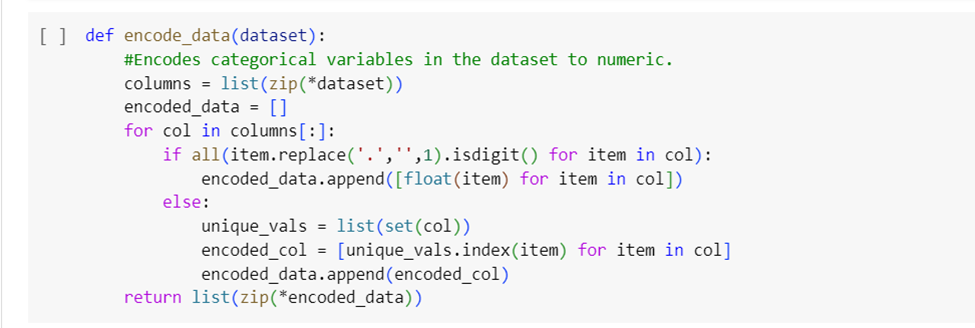
A black and white math equation

Description automatically generated with medium confidence

**Data Pre-processing:**

**Car Data & Breast-Cancer Data:**

I converted all the categorical features to numerical, so that we can train the algorithm in a better way. The below function is used to perform encoding.

****

**Model Analysis**

1. **Hayes-Roth Dataset:**

* **From Scratch:**

Developing KNN from scratch using K-fold cross validation. I calculated the accuracies for 4 different distance metrics to see which one performs better.

Euclidean Distance: **43.84%**

Manhattan Distance: **43.07%**

Hamming Distance: **45.38%**

Cosine Similarity: **20.00%**

The below code snippet is used for finding k value that gives us best accuracy for Euclidean distance and depicts it onto a graph using matplotlib. Did not use encode function as all the values in the dataset are numeric by default.

A screenshot of a computer program

Description automatically generated

A graph with red dots and blue lines

Description automatically generated

A screenshot of a computer

Description automatically generated

Hrscores1, hrscores2, hrscores3, hrscores4 are the variables that store accuracies obtained for each fold.

* **Using scikit learn:**

By using inbuilt libraries for K Nearest Neighbors & K-Fold, the below implementation of Hayes Roth dataset is by considering all the given features.

The accuracies obtained are:

Euclidean Distance: **40.98%**

Manhattan Distance: **44.06%**

Hamming Distance: **54.45%**

Cosine Similarity: **43.90%**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A white screen with black text

Description automatically generated

Cv\_scores is the k fold accuracies obtained from scikit learn implementation.

Therefore, we can conclude that Hamming and Euclidean Distance gave better accuracy than any other distance metric.

1. **Car Dataset:**

* **From Scratch:**

I converted all the categorical features to numerical so that we can train the algorithm in a better way.

I calculated the accuracies for 4 different distance metrics to see which one performs better.

Euclidean Distance: **69.59%**

Manhattan Distance: **69.06%**

Hamming Distance: **69.30%**

Cosine Similarity: **69.88%**

Encode function is used to convert all the categorical values to numerical values.

A screenshot of a computer

Description automatically generated

Cscores1, cscores2, cscores3, cscores4 are the variables that store accuracies obtained for each fold.

* **Using scikit learn:**

By using inbuilt libraries for K Nearest Neighbors & K-Fold, the below implementation of Car dataset is by considering all the given features.

The accuracies obtained are:

Euclidean Distance: **75.75%**

Manhattan Distance: **77.43%**

Hamming Distance: **77.72%**

Cosine Similarity: **77.545%**

A screenshot of a computer program

Description automatically generated

A computer code with text

Description automatically generated with medium confidence

A screenshot of a computer code

Description automatically generated

Therefore, we can conclude that Euclidean Distance gave better accuracy than any other distance metric.

1. **Breast Cancer Dataset:**

* **From Scratch:**

I converted all the categorical features to numerical so that we can train the algorithm in a better way.

I calculated the accuracies for 4 different distance metrics to see which one performs better.

Euclidean Distance: **75.35%**

Manhattan Distance: **77.5%**

Hamming Distance: **77.14%**

Cosine Similarity: **70.71%**

The below code snippet is used for finding k value that gives us best accuracy for Euclidean distance and depicts it onto a graph using matplotlib. Encode function is used to convert categorical values to numeric values.

A screenshot of a computer program

Description automatically generated

A graph with red dots and numbers

Description automatically generated

A screenshot of a computer

Description automatically generated

bcscores1, bcscores2, bcscores3, bcscores4 are the variables that store accuracies obtained for each fold.

* **Using Scikit learn:**

By using inbuilt libraries for K Nearest Neighbors & K-Fold, the below implementation of Breast-Cancer dataset is by considering all the given features.

The accuracies obtained are:

Euclidean Distance: **72.74%**

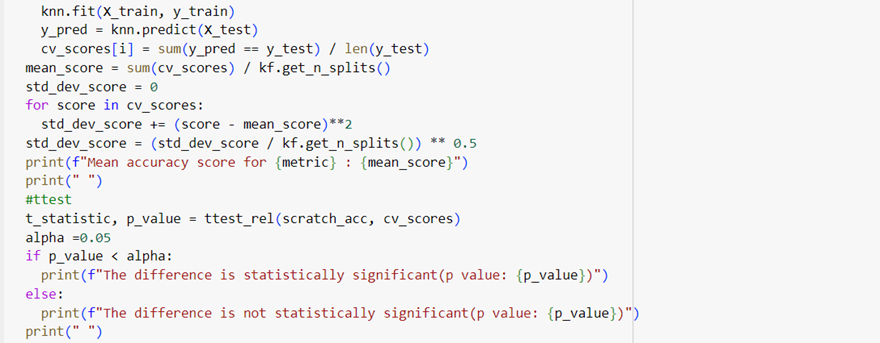
Manhattan Distance: **71.69%**

Hamming Distance: **74.12%**

Cosine Similarity: **71.26%**

A screenshot of a computer

Description automatically generated



A screenshot of a computer code

Description automatically generated

Therefore, we can conclude that Hamming Distance & Manhattan Distance gave better accuracy than any other distance metric.

**Hypothesis Testing:**

By performing Hypothesis Testing, null hypothesis and alternative hypothesis can be determined.

If the accuracy of the KNN implementation by scratch and the accuracy of sci-kit implementation has no significant difference, or difference is not statistically significant, then it can be concluded to be Null Hypothesis (H0).

Conversely, if the accuracy of the KNN implementation by scratch and the accuracy of sci-kit implementation has significant difference, or difference is statistically significant, then it can be concluded to be Alternative Hypothesis (H1).

**Statistical Significance test utilizing a Paired T-Test:**

I performed Paired T-Test for all 4-distance metrics that I considered, while performing scikit learn implementation. The mean accuracy mentioned in the below snippets is of scikit learn implementation.

**Paired t-test on Hayes-Roth Dataset:**

For Euclidean, Manhattan and Hamming distances the paired t-test difference came out to be not statistically significant.

For Cosine distance the difference is statistically significant.

A screenshot of a computer

Description automatically generated

**Paired t-test on Car Dataset:**

For all the distance metrics i.e., Euclidean, Manhattan, hamming and cosine the difference is statistically significant.

**A screenshot of a computer code

Description automatically generated**

**Paired t-test on Breast-Cancer Dataset:**

For all the distance metrics i.e., Euclidean, Manhattan, hamming, cosine the difference came out to be not statistically significant.**A white screen with numbers and letters

Description automatically generated**

**Overall Conclusion:**

Based on the t-statistics and p-values (considering majority), in case of Hayes Roth and Breast-Cancer data, the p-values are greater than the chosen significance level of 0.05 (alpha), the difference is not statistically significant, indicating that we should accept null hypothesis (H0).

In the case of Car dataset, the p-values are less than the chosen significance level of 0.05 (alpha), the difference is statistically significant, indicating that it is sufficient, and therefore rejecting null hypothesis (H1).

**REFERENCES:**

1. <https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/>
2. <https://machinelearningmastery.com/k-fold-cross-validation/>
3. <https://www.kaggle.com/code/burhanykiyakoglu/k-nn-logistic-regression-k-fold-cv-from-scratch>
4. <https://towardsdatascience.com/create-your-own-k-nearest-neighbors-algorithm-in-python-eb7093fc6339>
5. <https://medium.com/@avijit.bhattacharjee1996/implementing-k-fold-cross-validation-from-scratch-in-python-ae413b41c80d>