**KNN and distance metrics project**

**Prepared by:**

Omar Sami Mousa

Data mining course

# **Content**

[**Content 2**](#_Toc123922511)

[**Abstract 3**](#_Toc123922512)

[**Introduction 3**](#_Toc123922513)

[**Details 4**](#_Toc123922514)

[**Results and discussion 9**](#_Toc123922515)

[**Conclusion 12**](#_Toc123922516)

[**References 13**](#_Toc123922517)

# **Abstract**

There are many other distance metrics that can be used with the KNN algorithm, and the choice of distance metric can impact the algorithm's performance. It is often a clever idea to try out different distance metrics and see which one gives the best results on your dataset.

# **Introduction**

In the k-nearest neighbors (KNN) algorithm, the distance metric is used to determine which points in the training dataset are "nearest" to the new data point that we are trying to classify. The distance metric is used to compute the distance between the new data point and the existing points in the training dataset. The KNN algorithm then selects the K points in the training dataset that are closest to the new data point and classifies the new data point based on the most common class among these K points.

Some common distance metrics that can be used with the KNN algorithm include:

* Sorensen distance (SD)
* Additive distance (ASCSD)
* Divergence distance (DivD)
* Jaccard distance (JacD)

# **Details**

**Analytics of the datasets we chosen:**

1. **exams dataset :** This dataset consists of the marks of the students in various subjects, it evaluates students’ performance in exams.

The original dataset contains 8 columns (attributes)-- **[gender, race/ethnicity, parental level of education, lunch, test preparation course, math score, reading score, writing score]** and 1001 rows (observations).

We prepare and clean our dataset exams(UP).csv, there are a lot of numbers and strings, but there were no values to classification the KNN, so we made a simple equation to make males with (1) instead of (‘Male’), and females with (0) instead of (‘Female’).

We removed all columns from type string and kept the columns that have numbers (numeric) because they are basis of the work.

1. **diabetes dataset :** 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 a patient has diabetes, based on certain diagnostic measurements included in the dataset.

It contains 8 attributes and the class -- **[Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome(class)]** and 769 observations. It is particularly good and there are no problems or columns that should be removed. Most values in all columns are consistent.

1. **covid-19 dataset :** Contains a vast amount of anonymized patient-related information including pre-conditions. The raw dataset consists of 21 unique features and 1,048,576 unique patients. In the Boolean features, 1 means "yes" and 2 means "no". values as 97 and 99 are missing data.

In dataset Covid\_Data(UP).csv, this data is good, because it contains a lot of columns with values that we will classification the KNN. There are some columns that contain a string type, such as the dates column, so we removed it. Although data contains about half a million data records, we take the first 1500 data records.

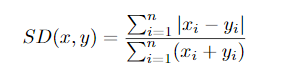
**Short explain for distances we chosen to work on it:**

1. **Sorensen distance (SD):**

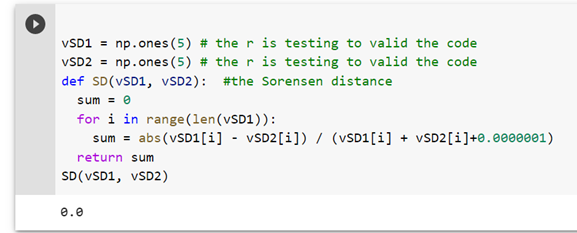
The Sorensen distance is a measure of dissimilarity between two sets. It is defined as the sum of the absolute differences between the elements of the two sets, divided by the sum of their elements.

The Sorensen distance is often used in fields like ecology and data analysis to measure dissimilarity between sample sets. It is often used as a substitute for the Jaccard index, which measures the similarity between sets, when the magnitude of the differences between the elements is more important than whether they are present or absent.

Sorensen’s original formula was intended to be applied to discrete data. Given two sets, X and Y, it is defined as:



**Code:**



1. **Additive distance (ASCSD)**

The Additive distance is a measure of dissimilarity between two sets. It is defined as the sum of the squared differences between the elements of the two sets, multiplied by the sum of their elements, divided by the product of their elements.



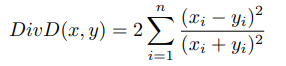
**Code:**



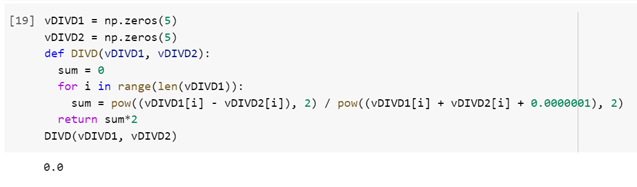
1. **Divergence distance (DivD)**

The Divergence distance, also known as the Kullback-Leibler divergence or relative entropy, is a measure of the dissimilarity between two probability distributions. It is defined as the difference between the entropy of one distribution and the cross-entropy of the two distributions, which is a measure of the average number of bits needed to represent the elements of one distribution using the other distribution as a reference.

Calculates the Divergence distance between two vectors. The Divergence distance is a measure of dissimilarity between two sets. It is defined as the sum of the squared differences between the elements of the two sets, divided by the squared sum of their elements.



**Code:**



1. **Jaccard distance (JacD)**

The Jaccard distance is a measure of dissimilarity between two sets. It is defined as the sum of the squared differences between the elements of the two sets, divided by the sum of the squares of the elements of the first set and the difference between the squares of the elements of the second set and the product of the elements of the two sets.



**Code:**



**Analytics of the code we have done:**

Graphical user interface, text

Description automatically generated

Here we assign the distances, datasets, and empty matrix to calculate the accuracy. We used a for-loop to execute all distances in all datasets and used pd.read\_csv function to read the data, then we split the data into training and testing model, chosen 30% for testing.

We make the object of the KNN classifier. After that we fit the model using fit () function, then we assign predicted values from testing model using model.predict() function. Furthermore, we create a variable(I), to show the result, we use (“”.join) to convert a path into string, (split) to split between strings by (/), (-1) to take last word in the path, [:-4] to remove last 4 strings in path (.csv) , then we define the performance measure(accuracy score, confusion matrix, precision score and recall score) to see how the classifier is work.

Then we use append function to add the result to the matrix we assign it before (accumat) and printing it to have output that we will discuss it in the next point.

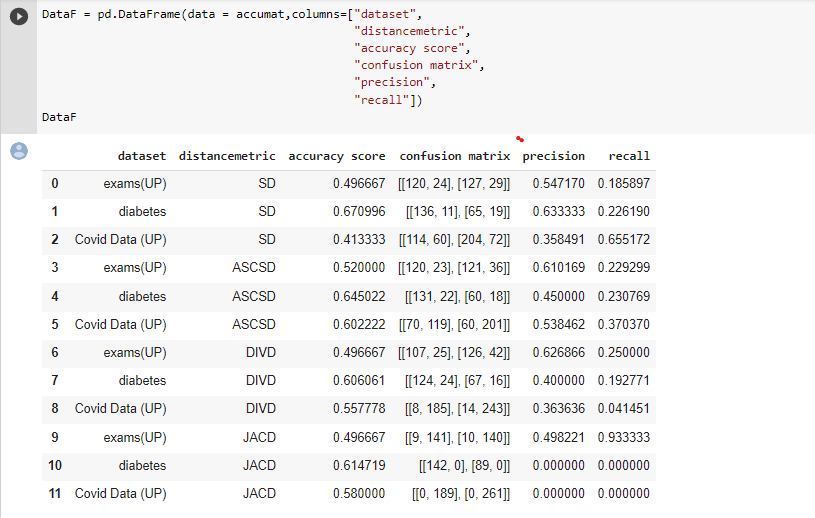
Finally, we draw a histogram for each measurement performance to show results, we will explain it in the next point also.

# **Results and discussion**

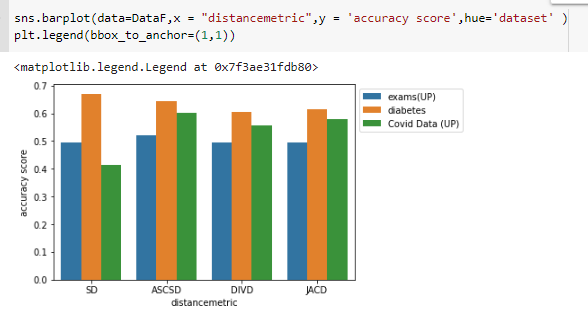
Text

Description automatically generated with medium confidence

The output shows the name of datasets then the distance we use, and the performance measure, but not clear enough to read it, so we use DataFrame function that we get it from pandas library and name the columns. To have this obvious output:



**Graphical representation (Histogram):**

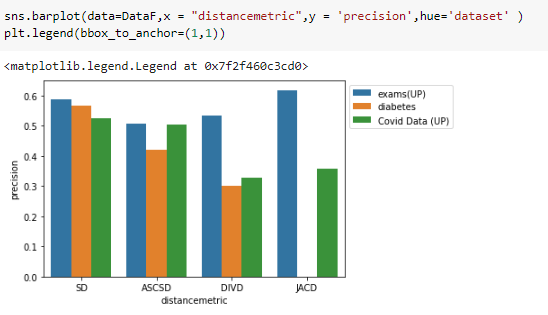
****

**Figure(1): Histogram for accuracy score**

The figure shows that exams dataset possesses the same accuracy score which is approximately 50%, but ASCSD showed a slightly greater percentage of 53%.

It also shows that diabetes dataset works good in all distances we used; yet, SD had the most accuracy rate of 68%.

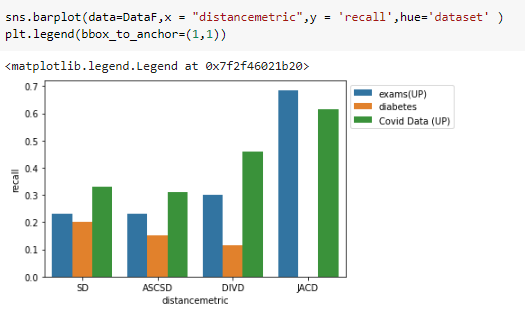
Covid data revealed that the most accurate scores were that of ASCSD, nearly

****

**Figure(2): Histogram for precision**

The figure above highlights that exams dataset had the greatest accuracy, 63%, with JACD. Variable results were seen with the Diabetes dataset within which the precision with JACD did not relay a satisfying outcome. On the other hand, SD illustrated the most precise results with a percentage of approximately 57%.

Covid Data depicted near precision results with SD and ASCSD upon which the percentages approximated 53%.



**Figure(3): Histogram for recall**

This figure shown that the exams dataset had the highest recall which is 68% with JACD, but diabetes dataset had a bad and not satisfying percentages with all distances, JACD was the worst. Covid Data had a high recall with JACD 65%.

# **Conclusion**

According to the statistics that appeared with us in the **Figure(1): Histogram for accuracy score**, we conclude that the values are close, but the distance ASCSD is better than SD because all datasets in it are of high accuracy. As for the distance DIVD and JACD, they are similar.

While the Jaccard distance yielded a good accuracy it did so because it misclassified most of the data and the data was mostly negative class, however it performed the worst because it was intended for Boolean vector space rather than real number victor space.

# **References**

* <https://www.kaggle.com/datasets/whenamancodes/predict-diabities>
* <https://www.kaggle.com/datasets/whenamancodes/students-performance-in-exams>
* <https://www.kaggle.com/datasets/meirnizri/covid19-dataset>
* <https://arxiv.org/pdf/1708.04321.pdf>