# K-Nearest Neighbor Report

This report contains Information regarding the KNN algorithm implementation from scratch and the dataset description.

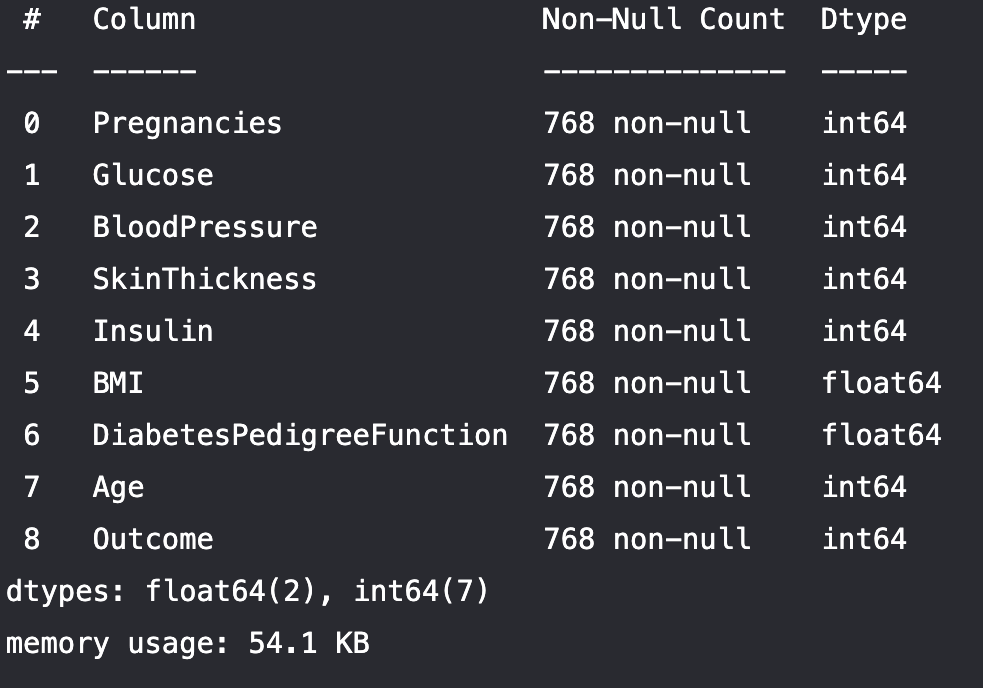
## Dataset Description

The data set used in the is project is the diabetes data set from hospital Frankfurt, Germany.

The data set was obtained and downloaded from [Kaggle](https://www.kaggle.com/c/diabetes/rules). User has to sign up for Kaggle and accept the competition to be able to download the data set.

Dataset or the problem statement is to use the remaining data set to determine if a person has diabetes (1) or not (0) with the outcome column.

The data set for diabetes has 9 features (columns) and 768 records (row). The dataset features and attribute types are shown in the image below



The data set attributes are mostly integers except for the BMI and DiabetesPedigreeFunction but technically the outcome variable is a categorical feature if the person had diabetes or not

### Description of Each Feature:

1. Pregnancies: Number of pregnancies a person has had (int)
2. Glucose: A person glucose level (int)
3. Blood Pressure: Person blood pressure (int)
4. SkinThickness: Usually caused by high glucose level as too much sugar in the blood prompts the body to pull fluid from cells in order to produce enough urine to remove the sugar. (int)
5. Insulin: Person insulin level(int)
6. BMI: A measure of a person weight relative to height (float)
7. DiabetesPedigreeFunction: A function that determines the risk of type 2 diabetes based on family history, the larger the function, the higher the risk of type 2 diabetes. (float)
8. Age: Person age (int)
9. Outcome: True or false 0,1 (int -> Categorical)

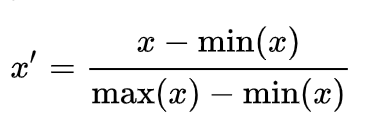
## Dataset Preprocessing

Data set contained no empty or null data point and no further data cleaning, wrangling was performed or feature engineering.  
However to improve KNN model performance, multiple data normalization and scaling was performed separately and performance was compared on each along with no normalization.

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.(*Feature Scaling - Wikipedia*, n.d.)

It is a good idea to rescale your data, such as using normalization, when using KNN(Brownlee, 2016)

1. Min- max normalization: Rescaling is the simplest method and consists in rescaling the range of features to scale the range in [0, 1] or [−1, 1] (*Feature Scaling - Wikipedia*, n.d.)



1. Standard Scalar: Using the default configuration, scale values by subtracting the mean to center them on 0.0 and divide by the standard deviation to give the standard deviation of 1.0. (Jason Brownlee, n.d.)

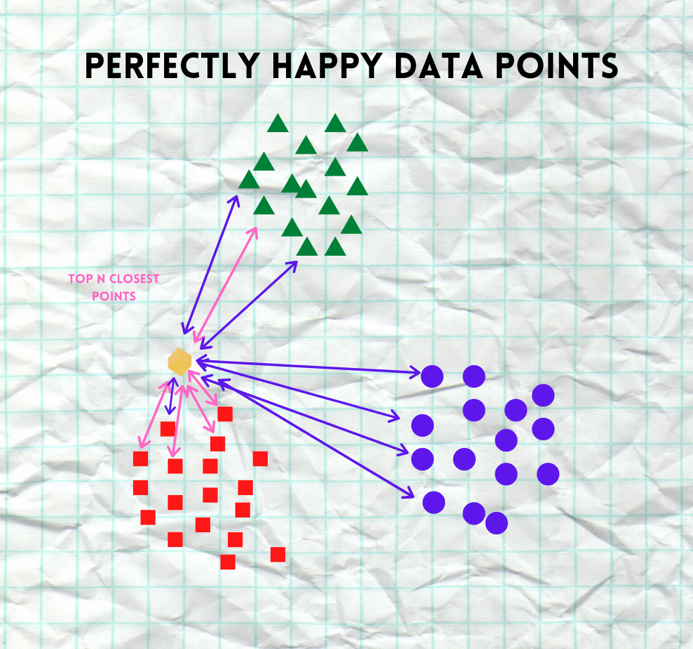
y = (x – mean) / standard deviation

## Algorithm Implementation

KNN (K Nearest Neighbor) is a supervised learning algorithm used for classification or regression. But unlike most other algorithms no learning occurs the model is simply a cached or stored dataset. Due to this fact no learning occurs data has to consistent and upgrade often and remove outliers.

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).(Onel, n.d.)

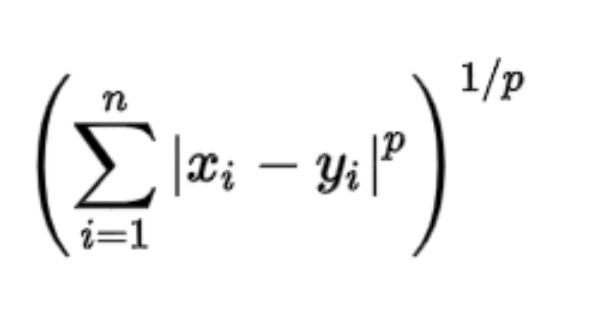
Prediction is performed by searching entire data set and compare for the k most similar instances hence the K in the algorithm name. For regression this can be done by averaging the output variables



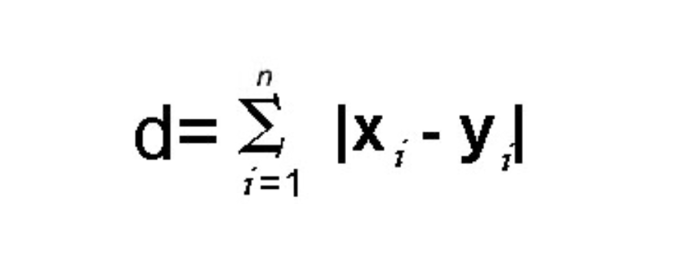
### Distance Metrics

Distance metrics are described below and their equations from(Sarang Anil, n.d.)

Minkowski Distance – It is a metric intended for real-valued vector spaces. We can calculate it only in a normed vector space, which means in a space where distances can be represented as a vector that has a length and the lengths cannot be negative. It is a generalization of Euclidean and Manhattan distance.

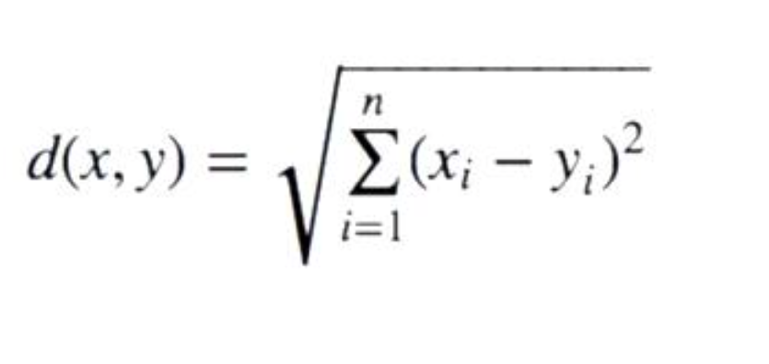


Manhattan Distance: The distance between two points is the sum of the absolute differences of their Cartesian coordinates.

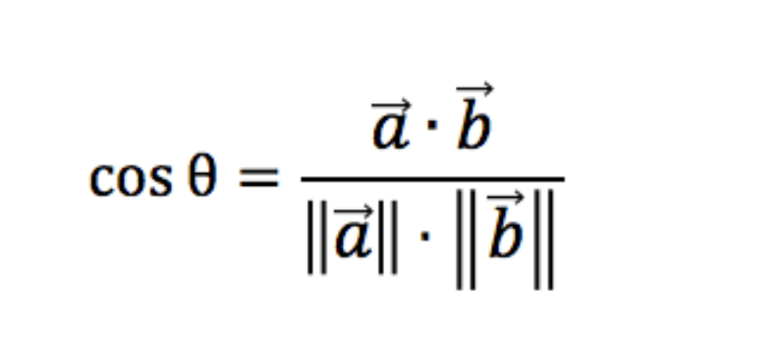


Euclidean Distance: distance is calculated as the square root of the sum of

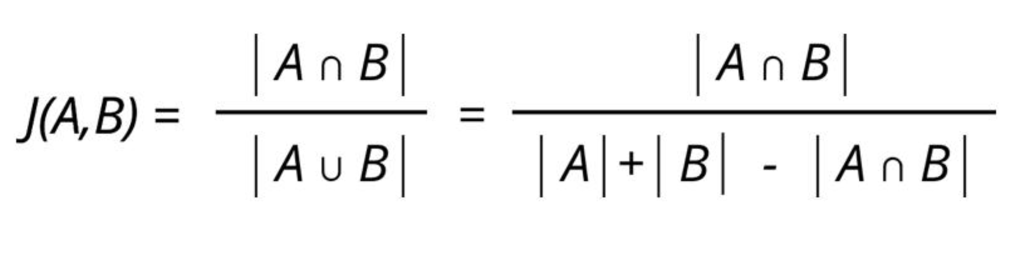
the squared differences between a point a and point b across all input attributes i. Which is the main the metric we would use



Cosine Distance – This distance metric is used mainly to calculate similarity between two vectors. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in the same direction.

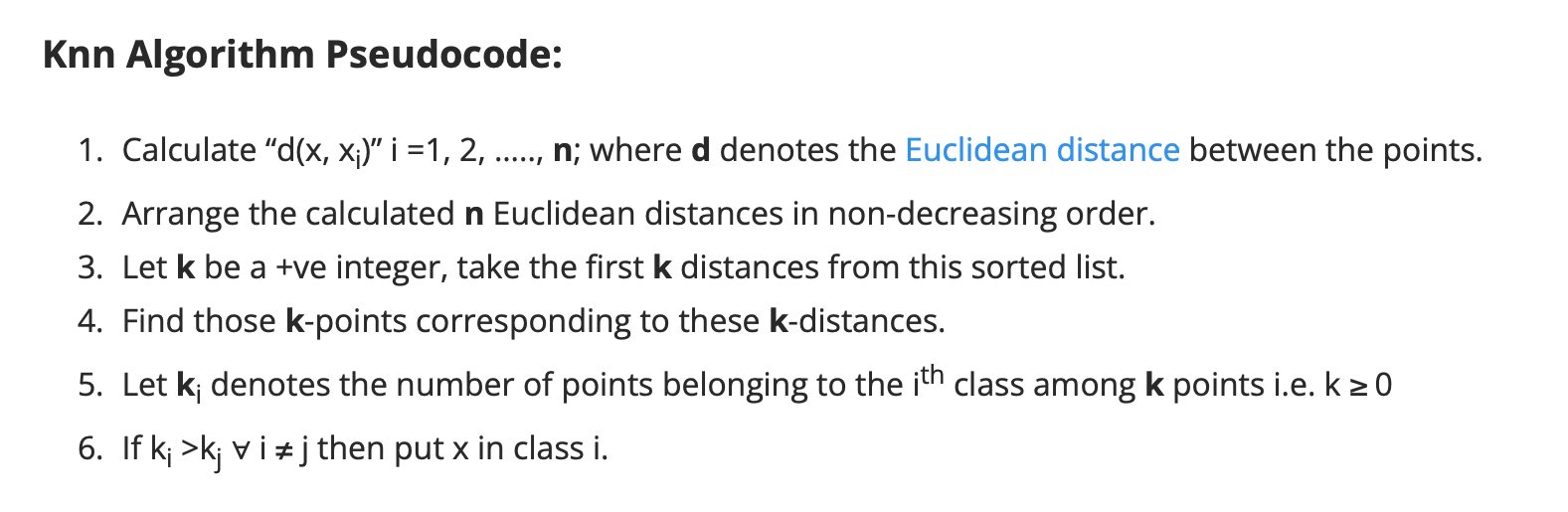


Jaccard Distance - The Jaccard coefficient is a similar method of comparison to the Cosine Similarity. The Jaccard approach looks at the two data sets and finds the incident where both values are equal to 1. So, the resulting value reflects how many 1 to 1 match occur in comparison to the total number of data points.

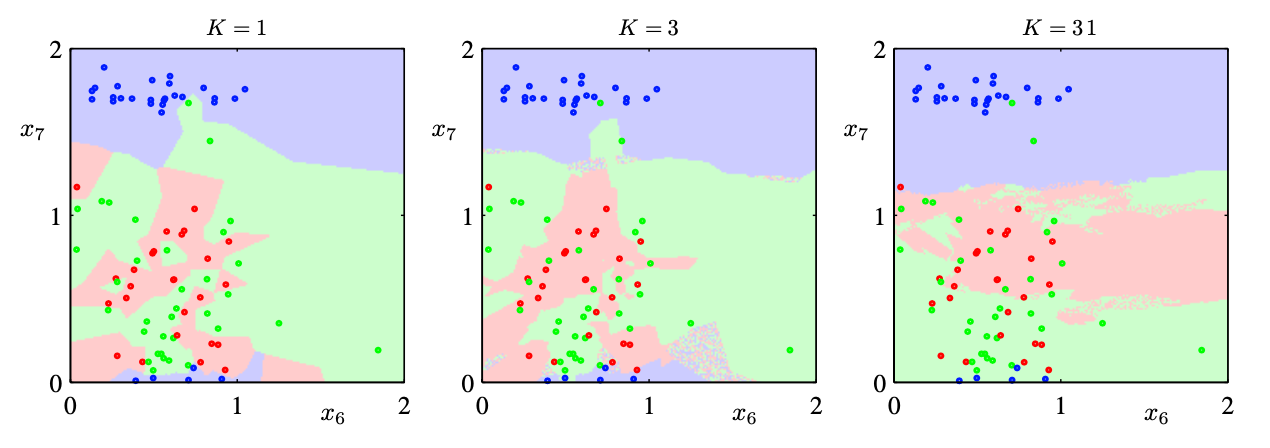


### Pseudocode and implementation

Pseudocode is presented here(Rahul, n.d.)

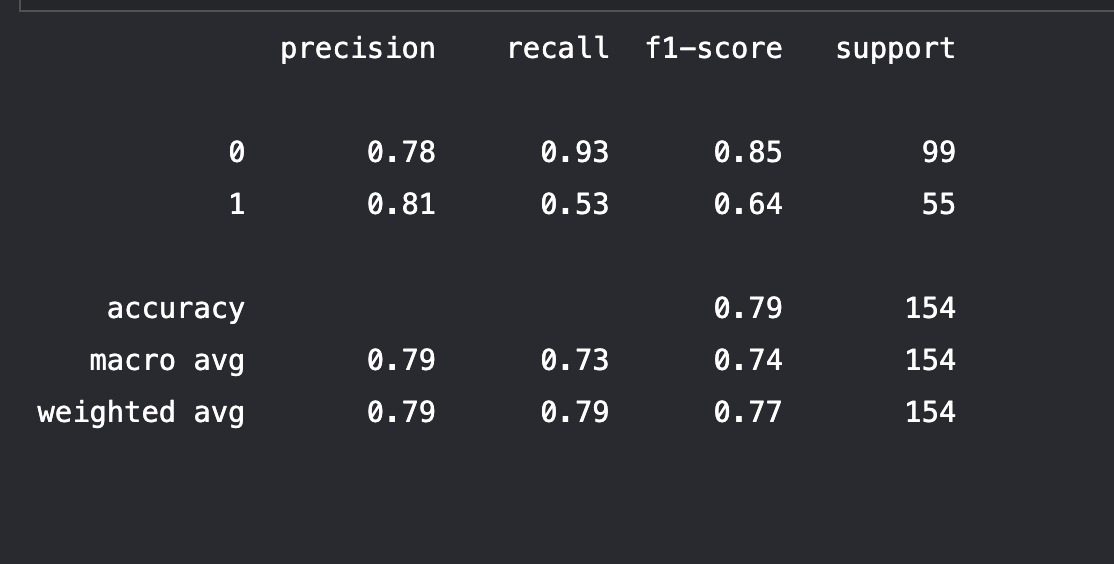


Choosing K value is done by a loop to determine the best value



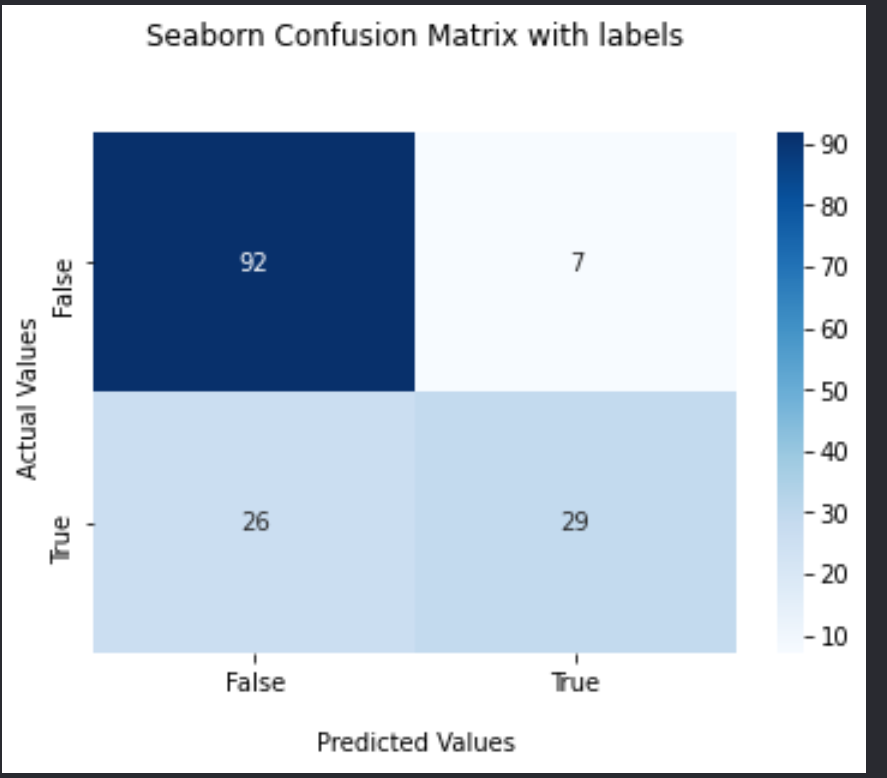
## Algorithm Performance

Using a k value of 21 and the standard scaler the performance metric is shown below this was obtained after trying multiple k values, distance metrics and Scaling methods



Shown below is the confusion matrix of the model and performance

(Vikram Aruchamy, 2021)



Further instructions on how to run and use the program are contained in the folder project folder read me

# References

Brownlee, J. (2016). *Master Machine Learning Algorithms: discover how they work and implement them from scratch*. Machine Learning Mastery.

*Feature scaling - Wikipedia*. (n.d.). Retrieved January 27, 2022, from https://en.wikipedia.org/wiki/Feature\_scaling

Jason Brownlee. (n.d.). *How to Use StandardScaler and MinMaxScaler Transforms in Python*. Retrieved January 27, 2022, from https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/

Onel, H. (n.d.). *Machine Learning Basics with the K-Nearest Neighbors Algorithm | by Onel Harrison | Towards Data Science*. Retrieved January 27, 2022, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Rahul, S. (n.d.). *Knn Classifier, Introduction to K-Nearest Neighbor Algorithm*. Retrieved January 27, 2022, from https://dataaspirant.com/k-nearest-neighbor-classifier-intro/

Sarang Anil, G. (n.d.). *Most Popular Distance Metrics Used in KNN and When to Use Them - KDnuggets*. Retrieved January 27, 2022, from https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html

Vikram Aruchamy. (2021, September 29). *How To Plot Confusion Matrix In Python And Why You Need To? - Stack Vidhya*. https://www.stackvidhya.com/plot-confusion-matrix-in-python-and-why/