Build and test Classifier KNN and Weighted-KNN with un-scaled and scaled data:

Objective of the project is to build K-Nearest Neighbor and Weighted-KNN algorithms and test the degree of accuracy.

K-Nearest Neighbors algorithm:

K-Nearest Neighbor algorithm (KNN) is a simple , easy to implement supervised machine learning algorithm that can be used to solve both classification and regression problems. Here it is used as a classifier.

Based on the value of the K the nearest K points are chosen based on distance metric.

Weighted-K-Nearest Neighbors algorithm:

In weighted KNN, the nearest K points are given a weight using a function called as the kernel function. The intuition behind weighted KNN, is to give more weight to the points which are nearby and less weight to the points which are farther away.

The function which is used is the inverse distance squared function.

Dataset:

In below project classifier was built and tested on train, test datasets related to red variants of the Portuguese "Vinho Verde" wine.

There are 11 numerical features listed 1-11 and a binary class or target variable (number 12).

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides 6
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol
12. Quality (target variable +1 or -1)

Pseudo code:

Below pseudo code was followed while predicting for test data from train data.

- Load data.
- Initialize K to chosen number of neighbors.
- For each example in train data
    - > Calculate the distance between the query example (each sample in test dataset) and current example from the train data.
    - > Add the distance and the index of the train example to the ordered collection.
- Sort the ordered collection of distances and indices in ascending order by the distance.
- Pick the first K entries from the sorted collection

- Get the labels based on the selected K entries
- Return the mode of the K labels

Choosing the right value of K:

To select the right value of K, KNN algorithm was run for all the odd values between 2 and 40.
The purpose here is to choose the K-value that reduces the number of errors while maintaining the algorithms ability to accurately make predictions when a given data is not seen before.

Distance metrics:

This project is executed based on two distance metrics:

1. Euclidean distance
2. Manhattan distance

1. **Euclidean distance:**
   Euclidean distance measures distance diagonally from one point to another, in other words the shortest distance between two points.

   Formula for Euclidean Distance for an n-dimensional space:

$$D(P,Q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

   Where, n = number of dimensions
   pi, qi = data points

2. **Manhattan distance**:
   Manhattan distance is the sum of absolute differences between points across all the dimensions.

   Formula for Manhattan Distance for an n-dimensional space:

$$D(P,Q) = \sum_{i=1}^{n}|q_i - p_i|$$

   Where, n = number of dimensions
   pi, qi = data points

Data Normalization:

Two data normalization techniques are used to make every feature data point have same scale.

1) Min-Max normalization
2) Standard normalization(Z-score)

1. Min-Max normalization:

Min-Max normalization rescales feature dataset into the range between 0 and 1.
Formula as given below:

$$\acute{v} = \frac{v - min}{max - min}(new\_max - new\_min) + new\_min$$

Where , min = minimum value of the feature
max = maximum value of the feature
new_min = 0
new_max = 1

2. Standard normalization(Z-score)

Z-Score rescales feature dataset according to stand normalization.
Formula as given below:

$$\acute{v} = \frac{v - mean}{std\_dev}$$

Where , meam = mean value of the feature
std_dev = standard deviation value of the feature

Performance Evaluation Measurement:

For evaluation of the created KNN classifier Accuracy (%) is used. Accuracy is calculated based on below formula:

$$Accuracy = \frac{Number\ of\ collectly\ classified\ test\ data}{Total\ number\ of\ test\ data} * 100$$

For evaluation of Accuracy line plot is used where Accuracy (%) is along Y-axis and the values of K along X-axis. K values are all the odd values between 2 and 40.

Evaluation of the results achieved without data normalization:

First KNN was build and executed based on Euclidean distance to find nearest neighbors on un-scaled data. The accuracy results are given below in 'Fig1' .

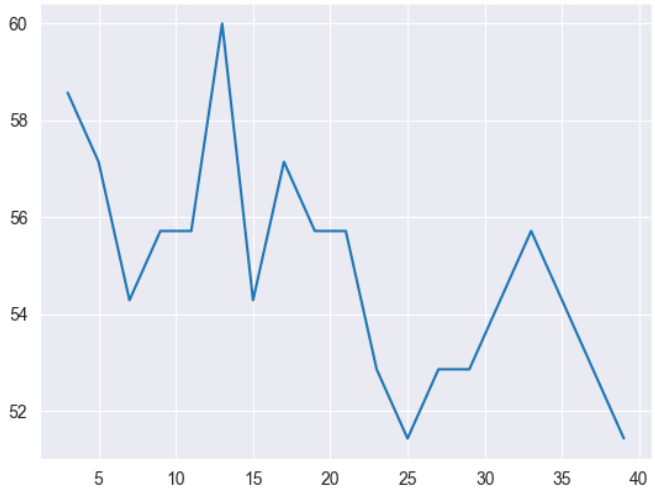

Fig1: Accuracy(%) vs. K values - KNN on Euclidean distance

Based on 'Fig1' for K value 13 the highest accuracy 60% achieved for KNN, and the accuracy was more than 51% for all k values.

A distance weighted KNN algorithm was build and executed based on Euclidean distance on un-scaled data. Inverse distance squared was used as distance weighted metric. The accuracy results are given below in 'Fig2'.

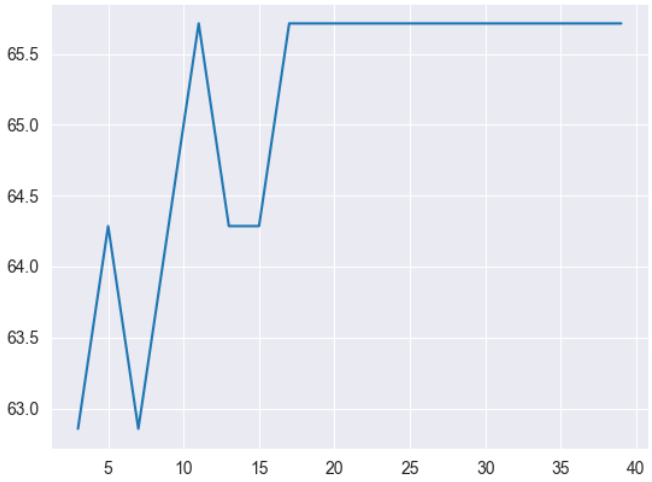

Fig2: Accuracy (%) vs. K values - Weighted KNN on Euclidean distance

Based on 'Fig2' for K value 11 and all values above 17 inclusive provide the highest accuracy result (65.7%). Accuracy was always more than 62% on all k values.

Third time both KNN and Weighted KNN were executed base on Manhattan distance for all odd K values between 2 and 40: the results are given below.

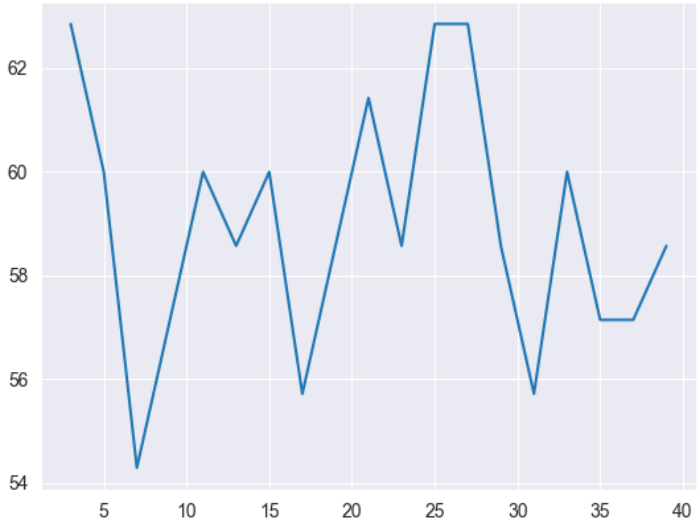

Fig3: Accuracy(%) vs. K values – KNN on Manhattan distance

'Fig3' shows the accuracy rate for differenct K values for KNN based on Manhattan distance. The maximum accuray 62.7% was for K values 3, 25 and 27. The accuracy was more than 54% for all k values.

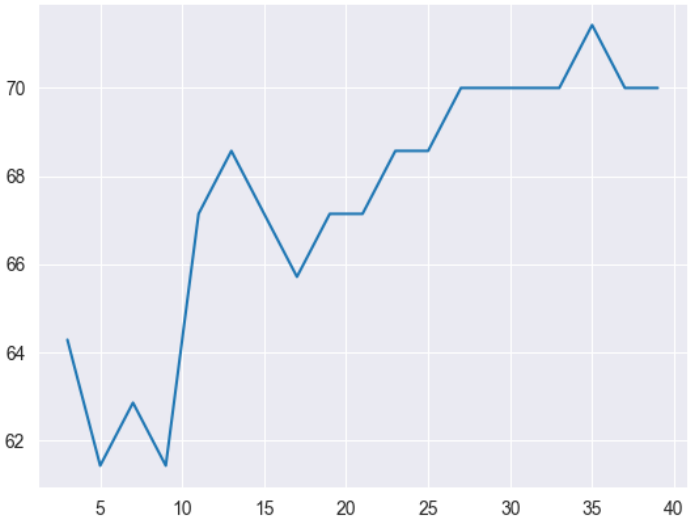

Fig4: Accuracy(%) vs. K values – Weighted KNN on Manhattan distance

'Fig4' shows the accuracy rate for differenct K values for weighted KNN based on Manhattan distance. This procuded the best accuracy result among all the test performed on un-scaled data. The highest accuracy rate is almost 71% for K value of 35. The accuracy was more than 61% for all k values.

Distance metric	Maximum KNN Accuracy (%)	Lowest K -Value for accuracy	Maximum Weighted-KNN Accuracy (%)	Lowest K -Value for accuracy
Euclidean	60	13	65.7	11
Manhattan	62.7	3	71	35

Table1: Highest Accuracies achieved for the lowest K values for different Distance metric on un-scaled data

'Table1' shows that the highest accuracy achieved for un-scaled data is 71% for Weighted –KNN on Manhattan distance with K value 35. Overall Weghted-KNN performed better than KNN and better accuracy results are seen with Manhattan distance as compared to Eucledian distance with un-scaled data.

Evaluation of the results achieved with data normalization:

KNN and Weighted KNN were executed for Scaled data using Min-Max Normalization and Standard Normalization (Z-Score). The results are discussed below.

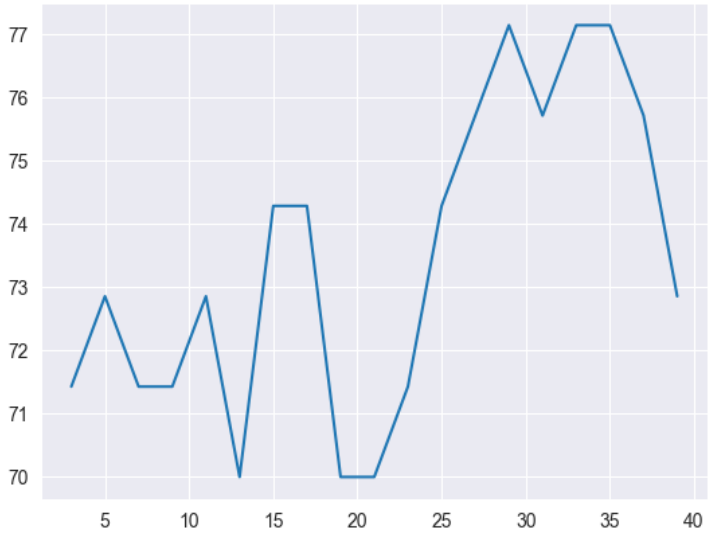

Fig5: Accuracy(%) vs. K values – KNN on Euclidesan distance with Min-Max Normalization

'Fig5' shows the accuracy results for different k values for KNN performed on Eucledian distance with min-max normalization on the feature train and test data set. The highest accuracy of 77.1% was achieved for k value 29, 33 and 35. The minimum accuracy was 70%.

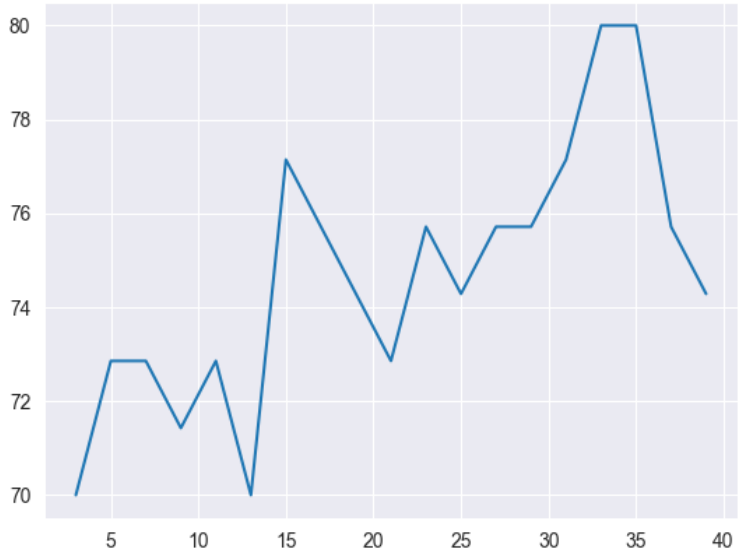

Fig6: Accuracy(%) vs. K values – Weighted KNN on Eucledian distance with Min-Max Normalization

'Fig6' shows the accuracy results for different k values for Weighted-KNN performed on Eucledian distance with Min-Max normalization on the feature train and test data set. The highest accuracy of 80% was acieved for k value 33 and 35. The minimum accuracy was 70%.

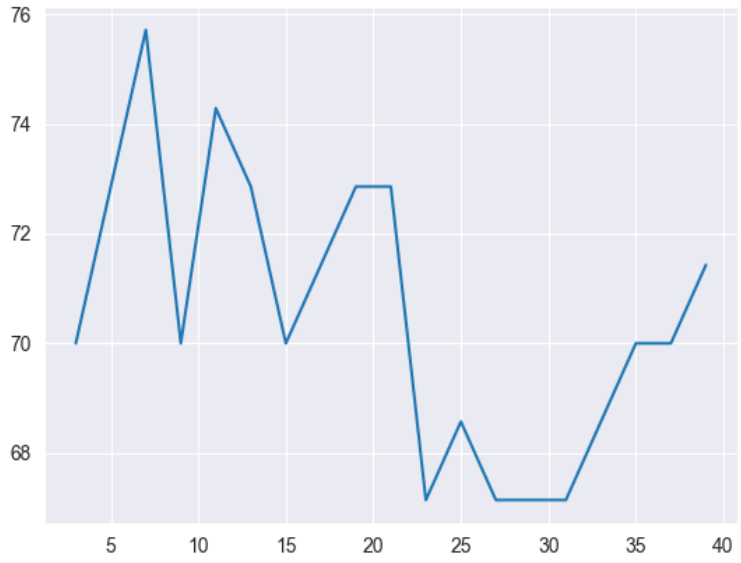

Fig7: Accuracy(%) vs. K values – KNN on Manhattan distance with Min-Max Normalization

'Fig7' shows the accuracy results for different k values for KNN performed on Manhattan distance with Min-Max normalization on the feature train and test data set. The highest accuracy of 75.8% was acieved for k value 7. The lowest accuracy was more than 67% for all values of K.

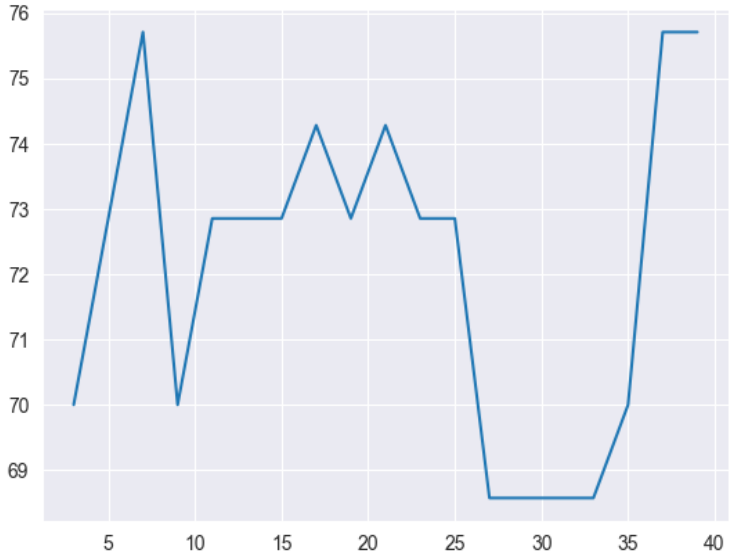

Fig8: Accuracy(%) vs. K values – Weighted KNN on Manhattan distance with Min-Max Normalization

'Fig8' shows the accuracy results for different k values for weighted-KNN performed on Manhattan distance with Min-Max normalization on the feature train and test data set. The highest accuracy of 75.8% was achieved for k value 7, 37 and 39. The lowest accuracy was more than 68% for all values of K.

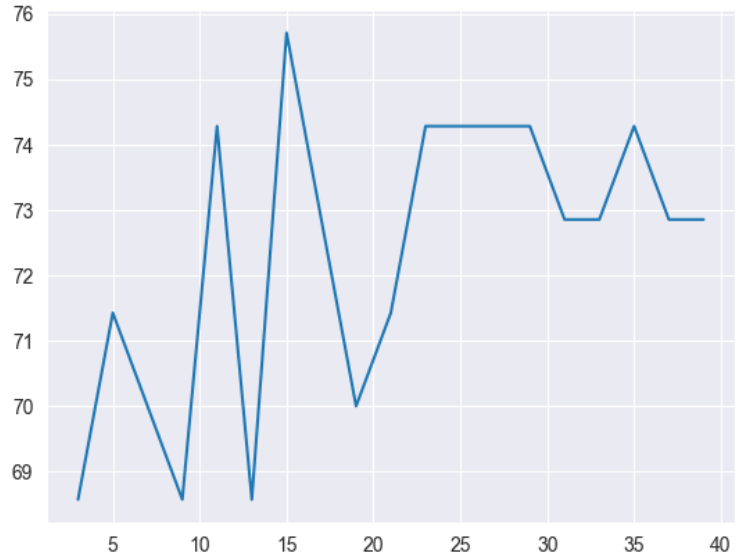

Fig9: Accuracy(%) vs. K values – KNN on Euclidean distance with Z-Score Normalization

'Fig9' shows the accuracy results for different k values for KNN performed on Euclidean distance with Z-score normalization on the feature train and test data set. The highest accuracy of 75.8% was achieved for k value 15. The lowest accuracy was more than 68% for all values of K.

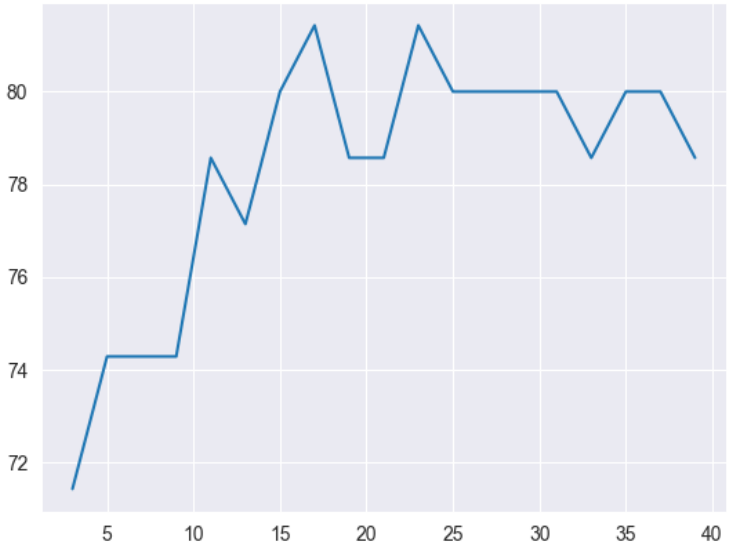

Fig10: Accuracy(%) vs. K values – WeightedKNN on Euclidean distance with Z-Score Normalization

'Fig10' shows the accuracy results for different k values for weighted KNN performed on Euclidean distance with Z-score normalization on the feature train and test data set. The highest accuracy of 81.2% was achieved for k value 17 & 23. The lowest accuracy was more than 71% for all values of K.

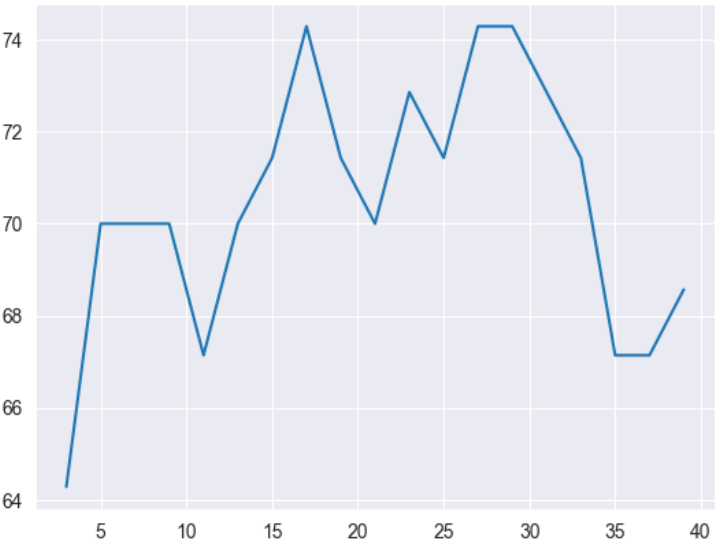

Fig11: Accuracy(%) vs. K values – KNN on Manhattan distance with Z-Score Normalization

'Fig11' shows the accuracy results for different k values forKNN performed on Manhattan distance with Z-score normalization on the feature train and test data set. The highest accuracy of 74.1% was achieved for k value 17 , 27 and 29. The lowest accuracy was more than 64% for all values of K.

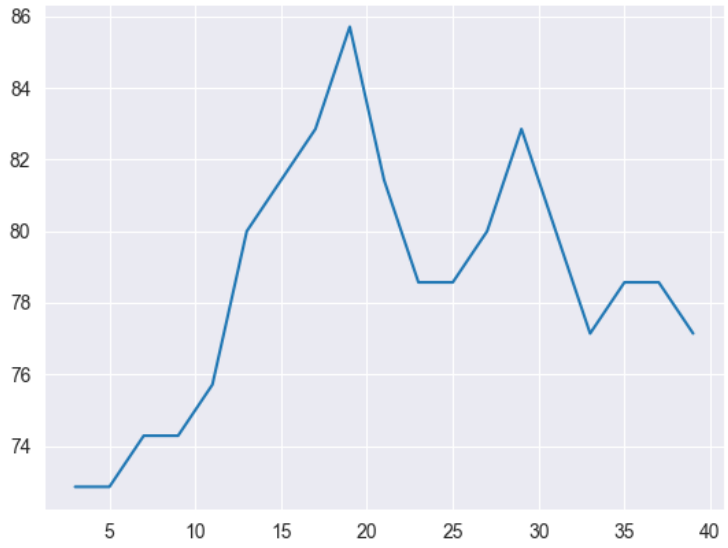

Fig12: Accuracy(%) vs. K values – WeightedKNN on Manhattan distance with Z-Score Normalization

'Fig12' shows the accuracy results for different k values forKNN performed on Manhattan distance with Z-score normalization on the feature train and test data set. The highest accuracy of 85.8% was achieved for k value 19. The lowest accuracy was more than 73% for all values of K.

	With Scaled data Using Min-Max Normalization				With Scaled data Using Z-Score Normalization			
Distance metric	**Maximum KNN Accuracy(%)**	**Lowest K -Value for accuracy**	**Maximum Weighted-KNN Accuracy (%)**	**Lowest K -Value for accuracy**	**Maximum KNN Accuracy(%)**	**Lowest K -Value for accuracy**	**Maximum Weighted-KNN Accuracy (%)**	**Lowest K -Value for accuracy**
Euclidean	77.1	29	80	33	75.8	15	81.2	17
Manhattan	75.8	7	75.8	7	74.1	17	85.8	19

Table2: Highest Accuracies achieved for the lowest K values for different Distance matric on Normalized data

From 'Table2' it is seen that for Min-Max normalization, better results were achieved for KNN and Weighted-Knn with Eucledian distance, for Standard normalized data beter performance was achieved for Weighted-KNN with Manhattan distance and for KNN with Euclidean distance. Over all Weighted-KNN performed better than KNN.

The best accuracy was 85.8%. procuded for Weighted-KNN with Manhattan distance with Z-Score normalization for k value 19.

Conclusion:

From the results described above it is clear that the performance in both classifiers are better with normalized data as compared to un-scaled data,
Weighted-KNN performed better than KNN. The best accuracy(85.8%) was produced for Weighted-KNN on Manhattan distance with Z-score normalised data for k value 19.

Refrerences:

1) P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by datamining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
2) https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
3) https://www.geeksforgeeks.org/weighted-k-nn/#:~:text=In%20weighted%20kNN%2C%20the%20nearest,points%20which%20are%20farther%20away.&text=The%20simple%20function%20which%20is%20used%20is%20the%20inverse%20distance%20function
4) https://www.codecademy.com/articles/normalization#:~:text=Min%2Dmax%20normalization%20is%20one,decimal%20between%200%20and%201.&text=That%20data%20is%20just%20as%20squished%20as%20before
5) https://www.analyticsvidhya.com/blog/2020/02/4-types-of-distance-metrics-in-machine-learning/#:~:text=Euclidean%20Distance%20represents%20the%20shortest,measure%20the%20similarity%20between%20observations.