Software Engineering Course

Lecturer: Damir Dobric / Andreas Pech

2023-2024

Investigate and Implement KNN Classifier

Group Name: Global Variable

Group Member: Muhammad Haris (1440274)

Zaka Ahmad (1436851)

Project Goal

1. KNN Experiment and Unit Test demonstrating how KNN works.
2. KNN Implementation with SP generated SDR’s.
3. KNN Implementation with TM generated SDR’s and Unit Test of the experiment.
4. Replace the HTMClassifier with your KNN Classifier

.

Table of contents

Table of contents

KNN Classifier Introduction

The K-Nearest Neighbors (KNN) algorithm is a simple yet effective method for classification tasks in machine learning. It is a non-parametric, instance-based learning algorithm, which means it doesn't make strong assumptions about the underlying data distribution and instead relies on the data itself during the prediction phase.

At its core, KNN makes predictions based on the majority class of the K nearest neighbors to a given data point. In other words, it classifies a new instance by finding the K most similar instances in the training data and assigning the most common class label among them to the new instance. However, it's important to note that KNN's performance can be sensitive to the choice of the number of neighbors (K) and the distance metric used to measure similarity between data points.

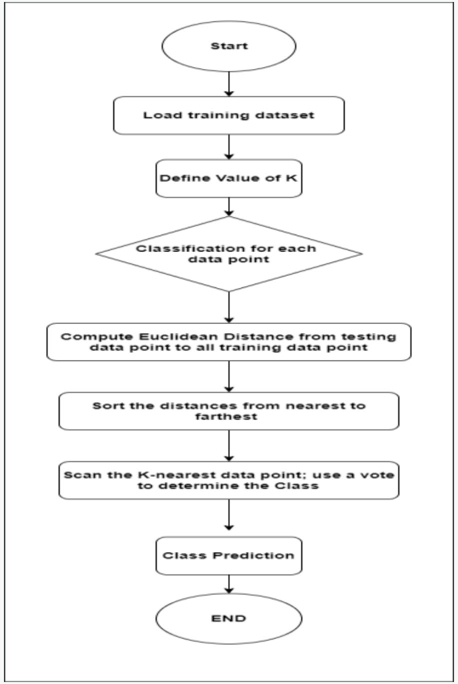
01

KNN Classifier Working Principle

Navigating the principles that form the Classifier of in-depth KNN studies

KNN Classifier Working Principle

1. Loading Dataset
2. Defining the Value of K
3. Calculate the distance between the testing data point and all the training data point.
4. Sort the distances and select the K nearest neighbors.
5. Assign the class label of test data by majority vote.
6. Predicting the Class



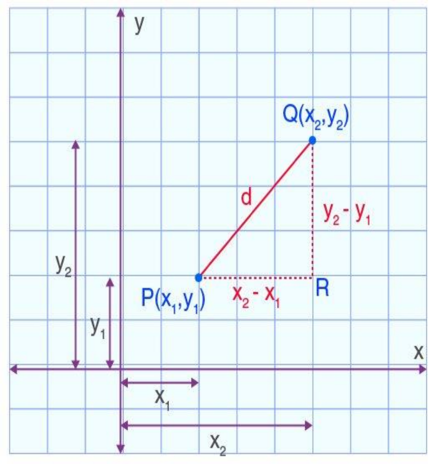
02

KNN Parameters and Matrix

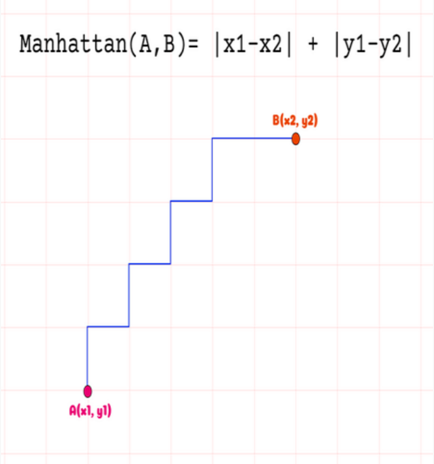
Distance matrix, K-Value selection, and Voting

Distance Metrix

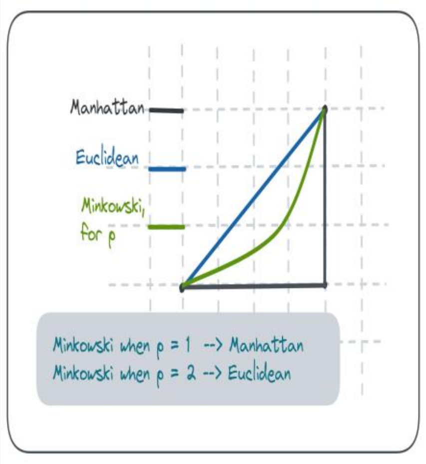
Euclidean Distance



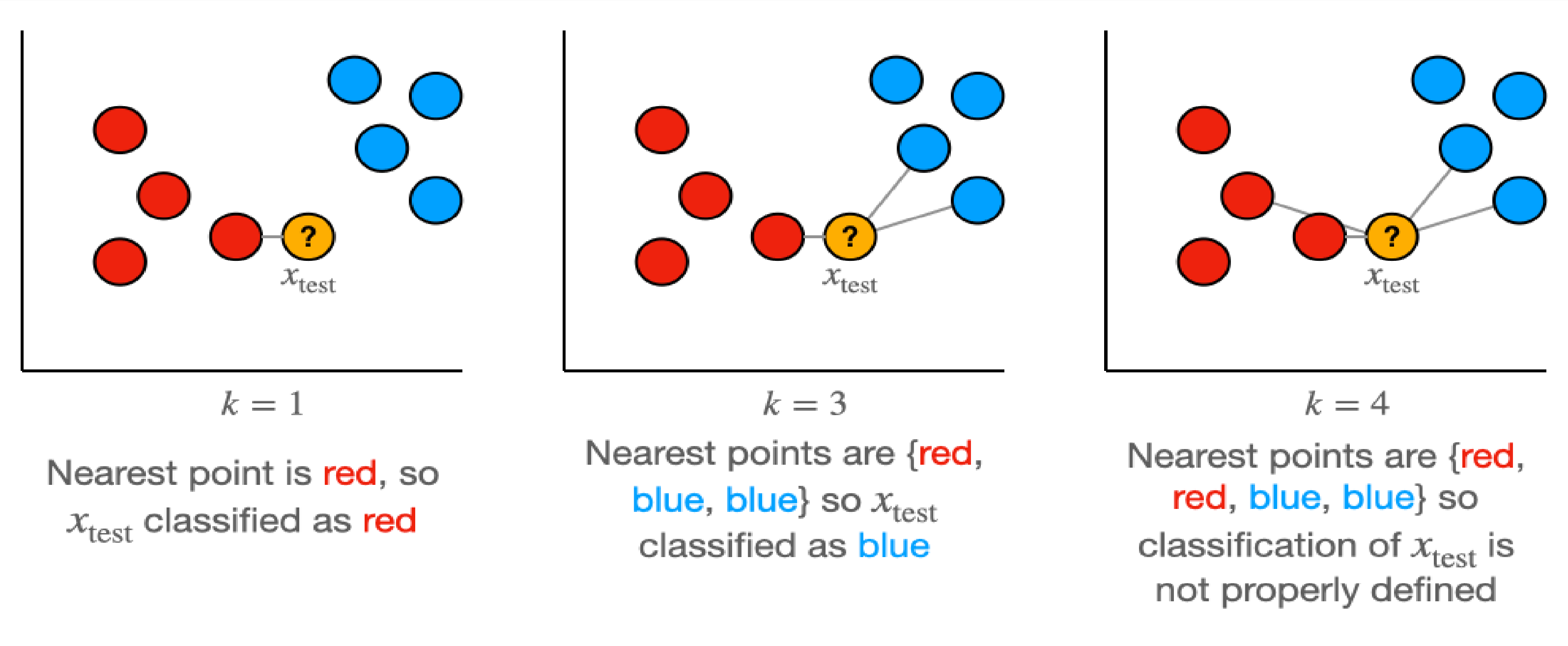
Manhattan Distance



Minkowski distance

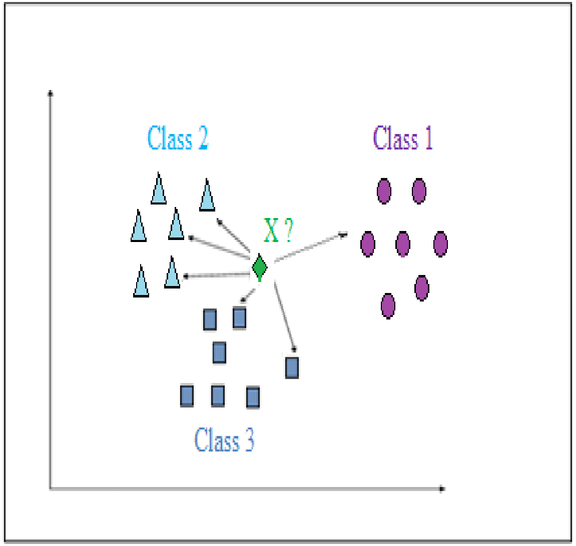


Define the Value of K



Voting Method

- Prediction in KNN is based on K nearest neighbors.
- The predicted or Winning class determined by majority voting principle.
- Class with highest number of votes assigned to the element.
  Important Thing to Consider
  - Choice of K impacts voting outcome.
  - Higher K values may lead to less confident predictions.
  - Imbalanced data scenarios can bias majority voting.
  - One class outweighing the other can skew predictions.
  - Techniques to resolve bias in majority voting are Distance-weighted voting and Weighted voting.

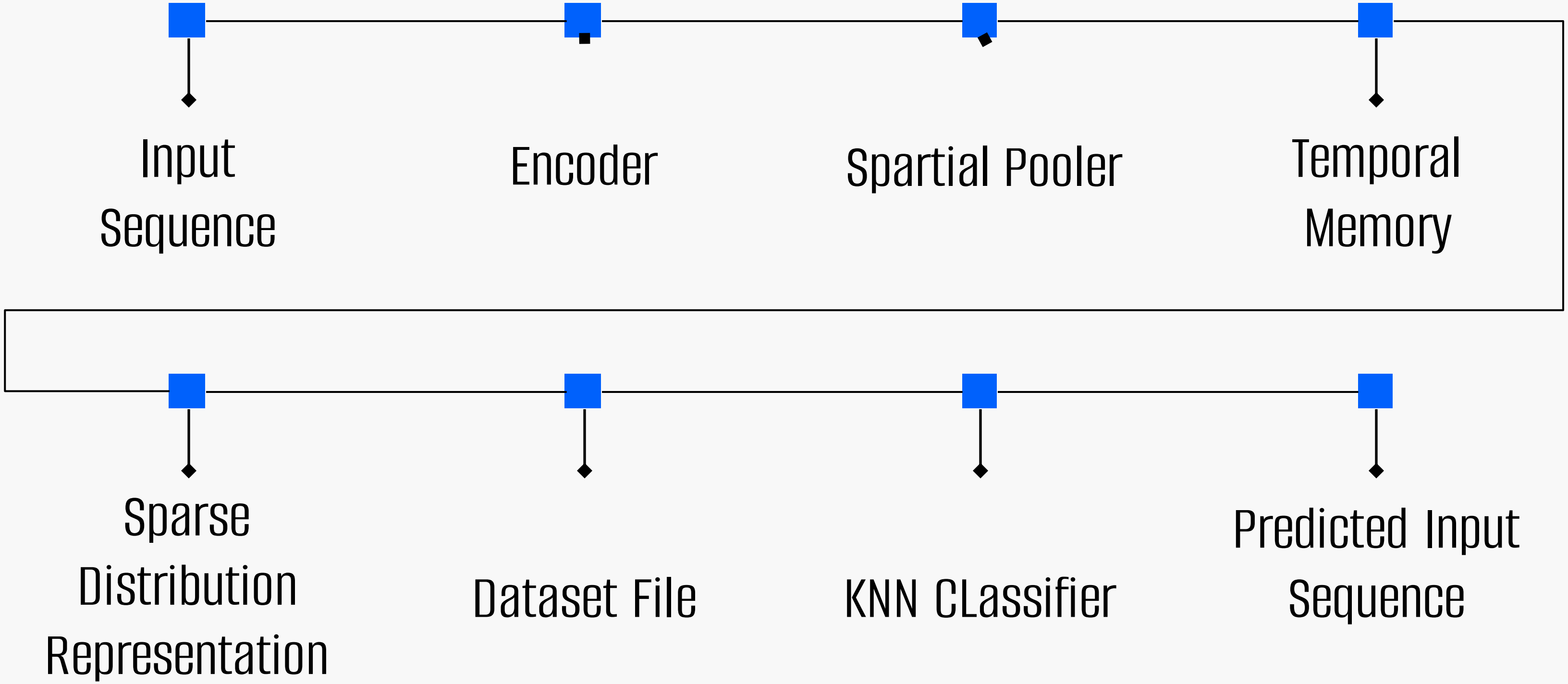


03

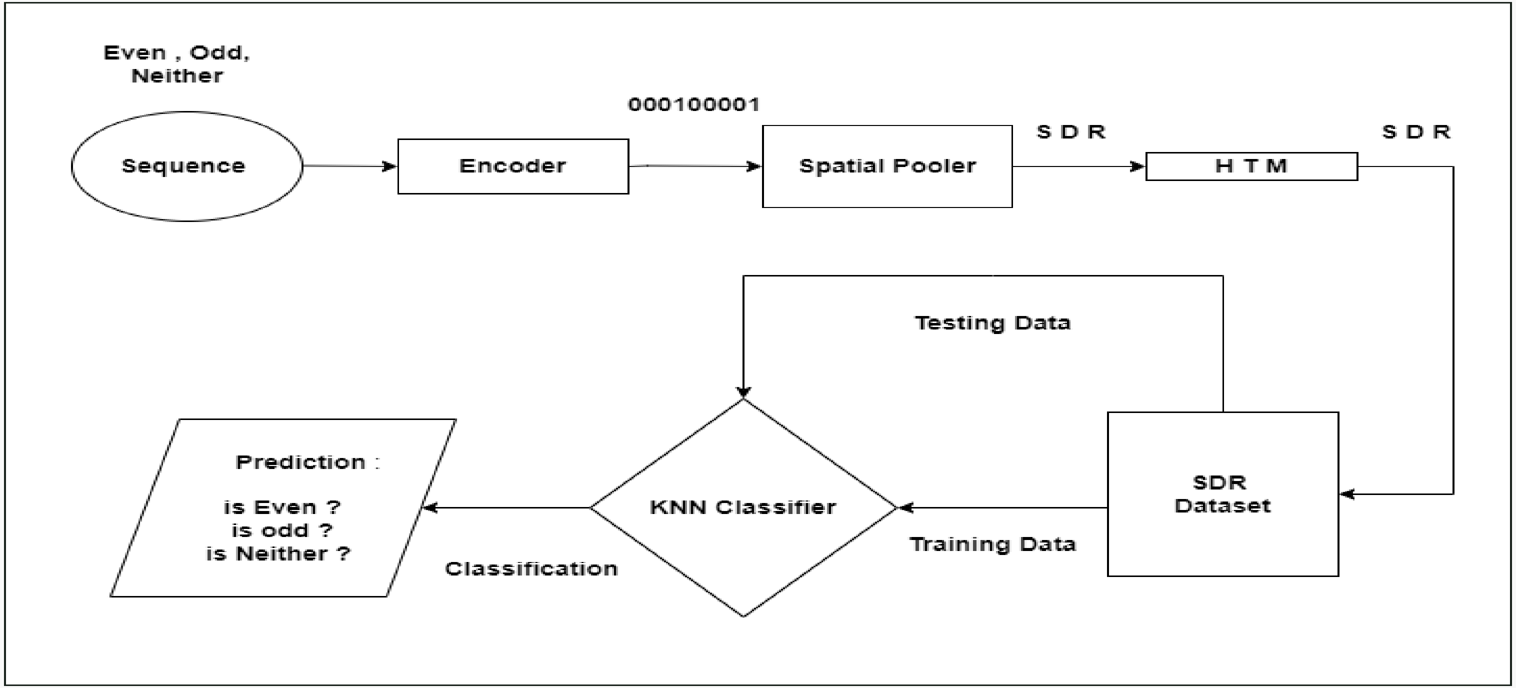
Project Methodology

Following a structured approach to efficiently manage project tasks

Project WorkFLOW



Process Diagram



04

Approaches and Challenges

Approaches and challenges face while executing the project

Approaches and Challenges

1. Classification of numbers from 0 to 9 in a sequence based on Sparse Distributed Representations (SDRs)

Classified data = [
{ 7, 18, 24, 29,……, 1012, 0 };
{ 25, 31, 44, 48,…… ,188, 1 };
{ 118, 123, 127, 156,………, 340, 2 };
{ 240, 242, 257, 266,………., 444, 3 };
{ 302, 314, 324, 327,……., 518, 4 };
{ 393, 405, 428, 429,…….., 624, 5 };
{ 483, 487, 500, 509,………, 726, 6 };
{ 579, 587, 595, 607,…………, 814, 7 };
{ 676, 691, 700, 707,……… , 916, 8 };
{ 772, 779, 780, 800,…………., 1007, 9 };
]

Unclassified data = {461, 495, 515, 501,…, 712 }

Approaches and Challenges

2. Classification of different Sets of Numbers sequences based on Sparse Distributed Representations (SDRs)


```
[
  {
    "SequenceName": "S1",
    "SequenceData": [8039, 8738, 9334, 9558, 9604, 9697, 9772, 9841, 9851, 9922, 9963, 10023, 10121, 10197, 10373, 10459, 10594, 10629, 10664, 11124]
  },
{
    "SequenceName": "S2",
    "SequenceData": [9051, 9075, 9133, 9178, 9365, 9448, 9481, 9599, 9635, 9740, 10032, 10224, 10281, 10762, 10778, 10934, 11143, 11306, 11494, 11763]
  },
{
    "SequenceName": "S3",
    "SequenceData": [10808, 10834, 11053, 11085, 11434, 11471, 11479, 11553, 11597, 11634, 11720, 11743, 11766, 11812, 11872, 11897, 11909, 12094, 12332, 12504]
  }, ...
]
```

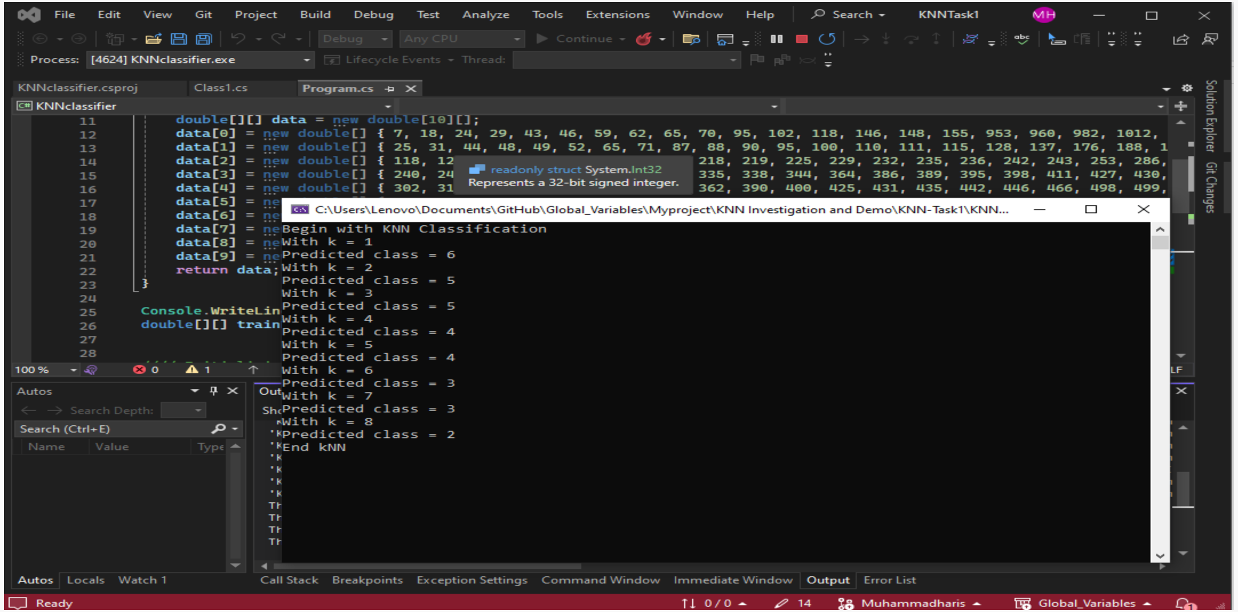
05

Result

Classification of the SDR Class on Output Window

Result

1. Classification of numbers from 0 to 9 in a sequence based on Sparse Distributed Representations (SDRs)



Result

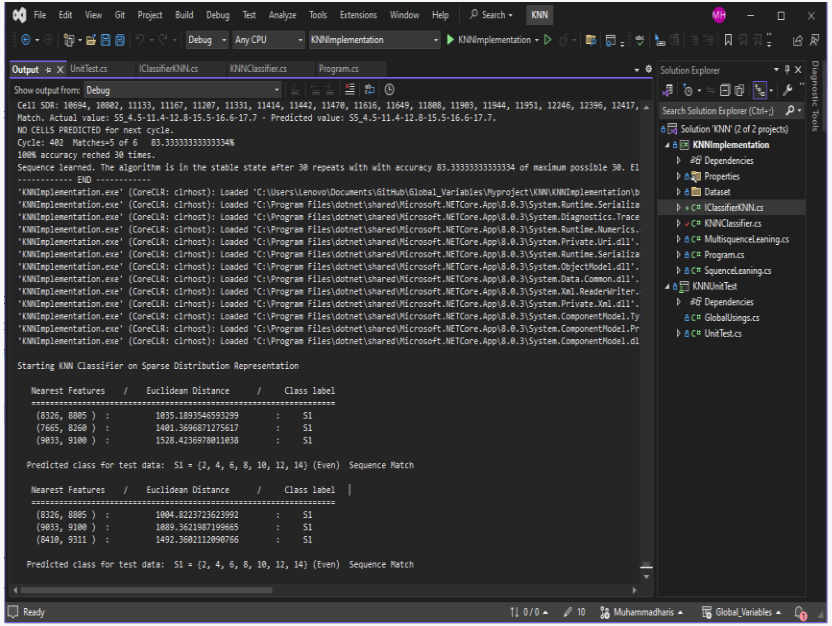
2. Classifying Sets of Numbers based on SDR's Value

K	Accuracy	Random Generated Test Data Accuracy in Percentage						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	100	90.9	90.9	90.9	90.9	90.9	90.9	100
2	100	100	90.9	90.9	90.9	90.9	90.9	100
3	100	100	100	100	100	90.9	90.9	100
4	90.9	100	100	90.9	90.9	90.9	90.9	100
5	100	100	100	90.9	100	90.9	90.9	100

TABLE I

ACCURACY OF THE KNN CLASSIFIER FOR DIFFERENT VALUE OF K FOR DIFFERENT TESTING DATA

Result



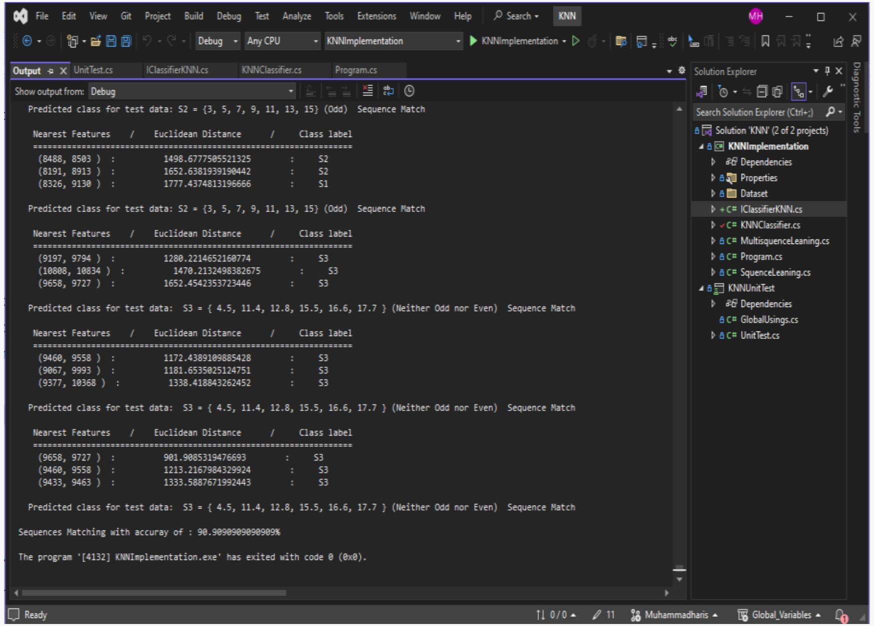
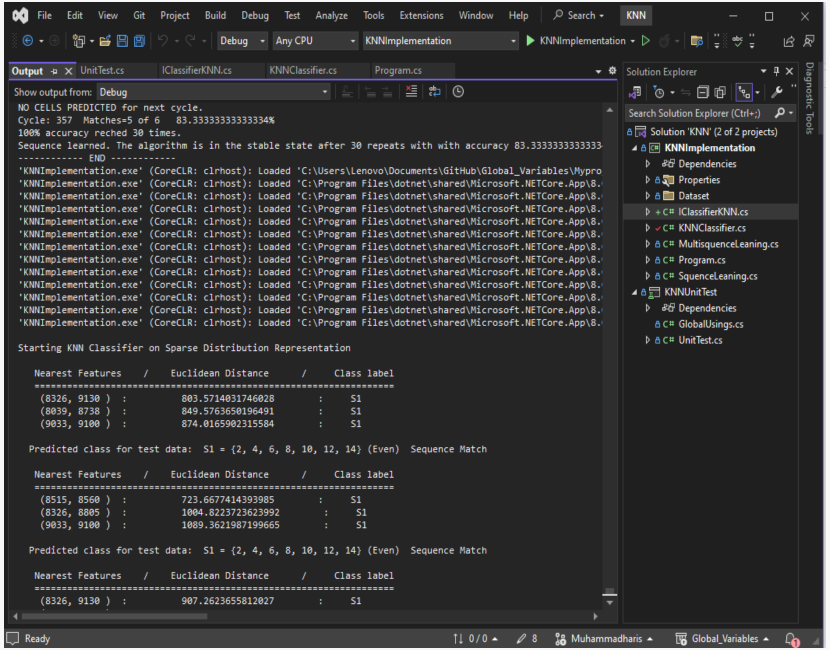


Figure 8 and 9 Output Window presenting the predicted result for test data with accuracy around 90 %.

Result



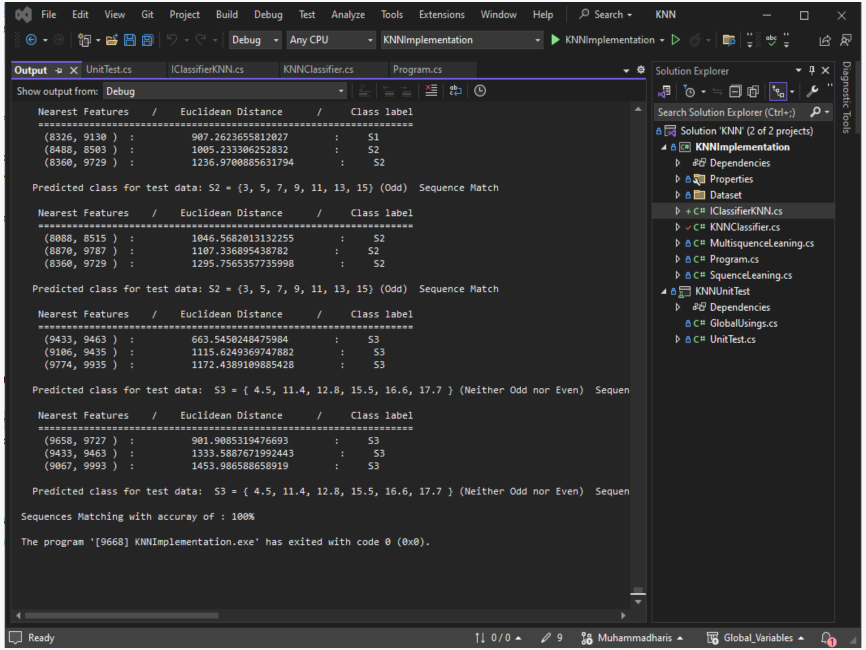


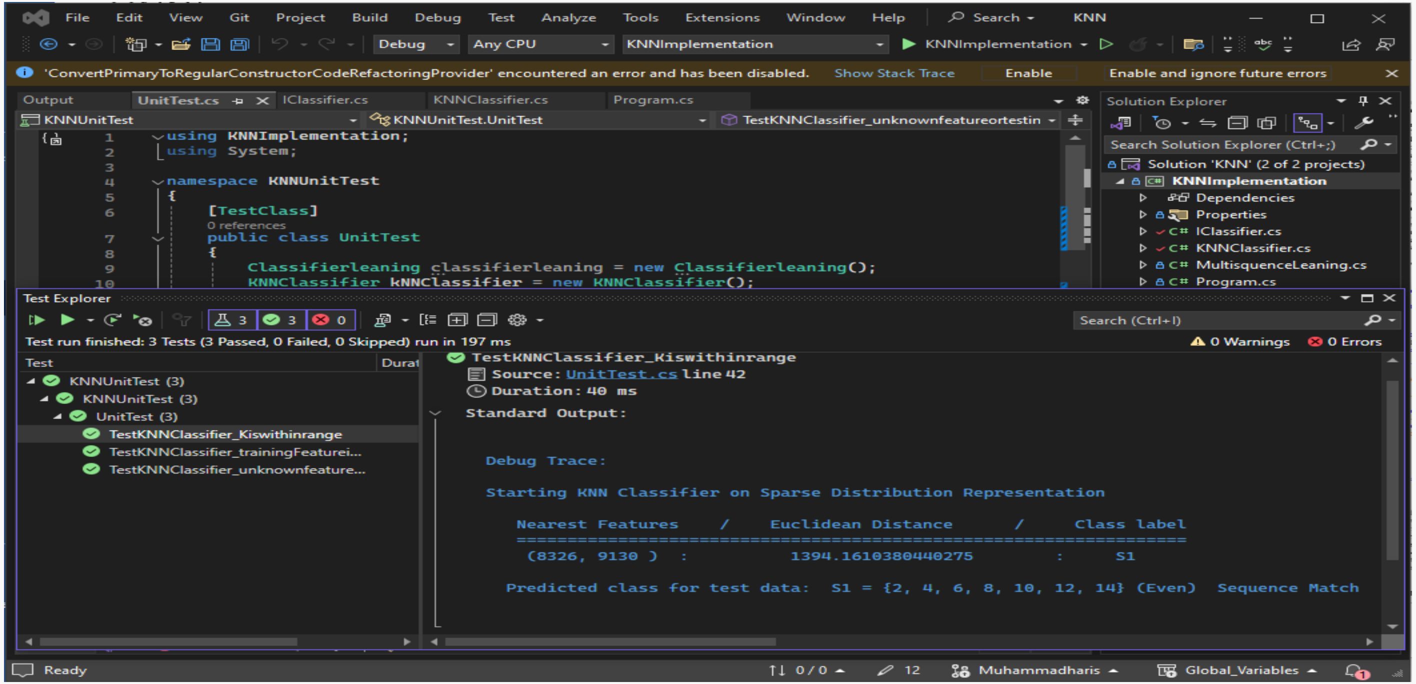
Figure 10 and 11 Output Window presenting the predicted result for test data with accuracy around 100 %.

06

Unit Test

Unit test preform to test the algorithm

Unit Test



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search KNN
Debug Any CPU KNNImplementation KNNImplementation
'ConvertPrimaryToRegularConstructorCodeRefactoringProvider' encountered an error and has been disabled. Show Stack Trace Enable Enable and ignore future errors
Output UnitTest.cs IClassifier.cs KNNClassifier.cs Program.cs
KNNUnitTest KNNUnitTest.UnitTest TestKNNClassifier_unknownfeatureortestin
using KNNImplementation;
using System;
namespace KNNUnitTest
{
[TestClass]
0 references
public class UnitTest
{
Classifierleaning classifierleaning = new Classifierleaning();
KNNClassifier kNNClassifier = new KNNClassifier();
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'KNN' (2 of 2 projects)
KNNImplementation
Dependencies
Properties
IClassifier.cs
KNNClassifier.cs
MultisquenceLeaning.cs
Program.cs
Test Explorer
Test run finished: 3 Tests (3 Passed, 0 Failed, 0 Skipped) run in 197 ms
0 Warnings 0 Errors
Test Durat
KNNUnitTest (3)
KNNUnitTest (3)
UnitTest (3)
TestKNNClassifier_Kiswithinrange
TestKNNClassifier_trainingFeaturei...
TestKNNClassifier_unknownfeature...
TestKNNClassifier_Kiswithinrange
Source: UnitTest.cs line 42
Duration: 40 ms
Standard Output:
Debug Trace:
Starting KNN Classifier on Sparse Distribution Representation
Nearest Features / Euclidean Distance / Class label
(8326, 9130) : 1394.1610380440275 : S1
Predicted class for test data: S1 = {2, 4, 6, 8, 10, 12, 14} (Even) Sequence Match
Ready 0 / 0 12 Muhammadharis Global_Variables

07

Conclusion

Concluding the solution of the problem with results.

Conclusions

KNN Prototype

We initiated the project by designing a generic KNN prototype, achieving the desired outcomes.

KNN integration

we developed the KNN model, seamlessly integrating it with the Neocortex API. The integrated model efficiently processes data streams to predict outcomes, with the KNN classifier accurately classifying sequences as matches or mismatches

Desired Result

our model demonstrates an exceptional accuracy rate of 100% across most input sequences. To ensure robustness, comprehensive unit tests, particularly referencing the HTM Classifier, have been implemented, yielding consistently satisfactory results.

08

Reference

Research Paper reference used for the project

References

- [1] N. F. F. E. B. S. L. P. d. L. Marcelo Beckmann, "A KNN Undersampling Approach for Data Balancing," Journal of Intelligent Learning Systems and Applications, vol. Vol.7 No.4, November 11, 2015.
- [2] T. &. H. P. Cover, "Nearest neighbor pattern classification.," IEEE transactions on information theory, 13(1), 21-27., 1967.
- [3] K. G. K. V. Han EH.., "“Text Categorization Using Weight Adjusted k-Nearest Neighbour Classification”.," In: Cheung D., Williams G.J., Li Q. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2001. Lecture Notes in Computer Sci, 2001.
- [4] Z. Z., " Introduction to machine learning: k-nearest neighbors.," Ann Transl Med. 2016 Jun;4(11):218., no. doi: 10.21037/atm.2016.03.37. PMID: 27386492; PMCID: PMC4916348..
- [5] J. H. D. Z. H. Z. a. C. L. G. Song Yang, " “KNN: Informative knearest neighbour pattern classification,”," Knowledge Discovery in Databases, , p. Pg. 248– 264., 2007.
- [6] N. A. M. G. R. M. J. Norsyela Muhammad Noor Mathivanan, "A comparative study on dimensionality reduction between principal component analysis and k-means clustering," Indonesian Journal of Electrical Engineering and Computer Science 16(2):752 , Vols. 10.11591/ijeecs.v16.i2.pp752-758, November 2019. [7]
- "Euclidean Distance, https://byjus.com/maths/euclidean-distance/".
- [8] L. &. L. C. &. M. N. &. M. A. Liberti, "Euclidean Distance Geometry and Applications," vol. SIAM Review. 56. 10.1137/120875909., 2012.

References

- [9] R. &. S. Z. &. Z. E. Suwanda, "Analysis of Euclidean Distance and Manhattan Distance in the K-Means Algorithm for Variations Number of Centroid K. Journal of Physics: Conference Series. 1566. 012058. 10.1088/1742-6596/1566/," 2020.
- [10] "How to Decide the Perfect Distance Metric For Your Machine Learning Model,," https://www.turing.com/kb/how-to-decide-perfectdistance- metric-for-machine-learning-model.
- [11] B. &. C. M. &. B. C. &. H. P. Lu, "The Minkowski approach for choosing the distance metric in geographically weighted regression.," International Journal of Geographical Information Science. , Vols. 30. 1-18. 10.1080/13658816.20, 2015.
- [12] A. &. K. V. &. R. T. Bookstein, "Generalized Hamming Distance.," vol. Information Retrieval. 5. 10.1023/A:1020499411651. , 2002.
- [13] K. Bala Priya C, "Distance Metrics: Euclidean, Manhattan, Minkowski,," https://www.kdnuggets.com/2023/03/distance-metricseuclidean-manhattan-minkowski-oh.html.
- [14] "What is the k-nearest neighbors (KNN) algorithm?," https://www.ibm.com/topics/knn.
- [15] J. R. S. H. G. S. R. Goldberger, "Neighbourhood components analysis.," NIPS (2004).
- [16] Y. J. D. &. C. D. Cai, "A KNN Research Paper Classification Method Based on Shared Nearest Neighbor.," NTCIR Conference on Evaluation of Information Access Technologies., 2010.
- ....

Thank You!