**CSG2341.3 Assignment 2.1**

**Approach/Implementation Section (individual):**

**- Describe implemented approach, explain algorithm**

The k-Nearest Neighbours algorithm is a simple supervised machine learning algorithm that can be used to make classification or regression predictions, based on the assumption that similar data points are likely to be grouped together in a dataset. The algorithm works by finding the *k* closest data points to a given value, i.e. its *nearest neighbours*, and simply assigning it whichever class represents the majority. For example, if the majority of a data point X’s nearest neighbours are of class A, then a k-NN algorithm will predict that X is also of class A. In cases where more than two classes are present but none has an outright majority, the algorithm will predict whichever class is represented more than the others, also known as the plurality.

To determine the ‘closeness’ of potentially abstract data points, a distance metric must be chosen. Examples of distance metrics used in k-NN algorithms include Manhattan Distance, which \*\*\*quote from https://www.educative.io/answers/what-is-manhattan-distance-in-machine-learning here\*\*\* and Hamming Distance, used to compare strings \*\*\*quote from https://www.analyticsvidhya.com/blog/2020/02/4-types-of-distance-metrics-in-machine-learning/#h-minkowski-distance here\*\*\*. This paper will use Euclidean Distance to determine the nearest neighbours of numerical datapoints. Euclidean Distance can be visualised by plotting points of numerical data as Cartesian coordinates, and measuring the length of a line segment joining two given points. While this distance metric limits the implementation of the k-NN algorithm to numerical data only, \*\*\*something about easy to understand implementation\*\*\*

The k-NN algorithm is limited in that it can be inefficient when dealing with large, varied datasets and is only useful for simple predictions. However, this does not preclude its use in a wide range of applications today, such as content suggestion, bla blah blah.

**- Address all steps with maths and illustrations**

1. Find k nearest neighbours using Euclidean distance

2. Find majority vote

3. assign to data point

**- Include flowchart and describe full implementation process**

**- data collection**

**- training**

**- validation**

**- testing**

**- Include all used parameters**

Dataset from scikit-learn, description of it

**- Describe language/software/tools used**

Python, scikit-learn

**Performance Evaluation Section (individual):**

**- Describe dataset, report performance of algorithm**

Iris dataset

**- Demonstrate findings with tables/figures**

Figs, etc

**Performance Comparison Section (joint)**

**- Compare algorithm performance on common dataset**

**- Demonstrate findings with tables/figures**

**REFERENCES:**