Joseph Oluwasanya, joseph.oluwasanya2@mail.dcu.ue

# **CA270 Classification Assignment**

## **Section 1: Description**

The Iris dataset was used in R.A. Fisher's classic 1936 paper, The Use of Multiple Measurements in Taxonomic Problems

**Dataset description:**

name: Iris

Attributes:

**Id**: Unique identifier for each iris plant

SepalLengthCm: Sepal length in centimetres

SepalWidthCm: Sepal width in centimetres

PetalLengthCm: Petal length in centimetres

PetalWidthCm: Petal width in centimetres

Species: species of iris, one of the following: {setosa,versicolor, virginica}

In this module I will attempt to implement classification algorithms to predict the species of an iris.

## **Section 2: Analysis**

To begin the analysis, I get the summary statistics for the entire dataset.

A screenshot of a cell phone screen with text

Description automatically generated

From this it appears that the deviation in values of Petal Length are the highest so I would like to see its distribution of values. I would also like to see the relationship between (Sepal length, Sepal width), (Petal length, Petal width). I will colour the scatter plots by species. The dataset is also small which implies that a high complexity algorithm is suitable

Before plotting I will check the counts of occurrences of each class in the dataset.

Graphical user interface, text

Description automatically generated

This is perfect. The training set is unlikely to contain biases due to scarcity of certain classes in the data. The data is all quantitative. Off the bat I rule out the Naïve Bayes algorithm as it performs best with categorical variables.

Distribution of Petal lengths: Chart, histogram

Description automatically generated

Sepal Length vs Sepal Height (colour-grouped by Species)

Chart, scatter chart

Description automatically generated

Petal Length vs Petal Width (colour-grouped by Species)

Chart, scatter chart

Description automatically generated

The **first plot** (Histogram) shows a bimodal distribution with a high mode of Petal lengths between 1 and 2cm. Referencing with the third plot (above) reveals that this lower mode is that of the setosa species of iris. It seems that setosa iris’ have unique dimensions from the other 2 species, suggesting that their classifications should be more accurate than that of the other species.

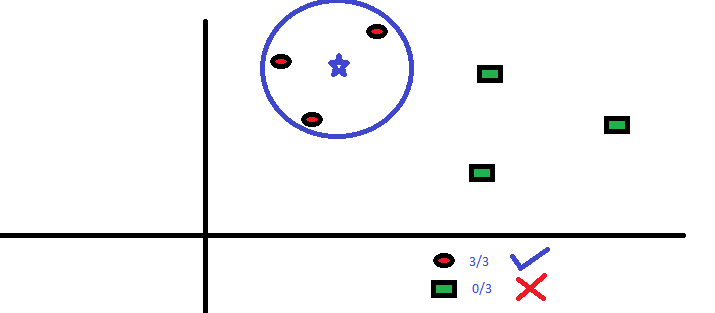
The **second plot** of sepal dimensions reveals that setosa sepal widths are generally above the median (3cm). There is a strong positive correlation between setosa sepal length and width. Both versicolor and virginica iris’ have a positive correlation between sepal length and width also. Just based on the colour-grouping it’s clear the sepal dimensions are fairly clustered amongst the different species and particularly with setosa species.

The **final plot** of petal dimensions is the most statistically significant. There is a strong positive correlation between petal lengths and widths. As well as this the colour-grouping shows that the Petal lengths and widths are clustered by species. This is motivation for the use of the K-nearest neighbours classification algorithm to predict the species of unseen instances. The dimensions of petals of the different species appear to rank as follows.

Setosa < Versicolor < Virginica

## **Section 3: Describing the algorithm**

The algorithm I will be using is K nearest neighbours. It estimates the classification of an unseen instance using the classification of instances closest to it. This model does not need to be pre-trained which saves time that would be spent on the training phase of implementation. I find it easiest to describe this model in R2 space:



The diagram above is a visual representation of a 3-nearest neighbours’ implementation on a 2-dimensional dataset. The training data points are mapped onto the R2 plane and their symbols represent classes. (red ellipse and green square) The unsees instance is then mapped onto the plane and the “distance” between this instance and all the training points are found. The 3 closest points are used to classify the unseen instance. If there is a combination of classes in the 3 points, the model goes by a majority rules calculation (this is often the case). The distance measure between the points must satisfy a few conditions:

1. The distance between a point and itself is 0: dist(X, X)
2. The distance between point X1 and X2 is the same as the distance between point X2 and X1: dist(X1, X2) = dist(X2, X1)
3. The shortest distance between any 2 points is a straight line

In practice, KNN is usually used in a hyper dimensional dataset (> 3) but the principle remains the same. A common distance measure is Euclidean distance. With the distance between two points a = (a1, a2, …, an) and b = (b1, b2, …, bn) being:

The model is easily skewed by disproportionate ranges of attributes. For example, I can have one attribute for a person’s height in cm and another for the diameter of their wrist in cm. The dimension defined by the height in cm will skew the model unless I normalize the values. Mapping a value x to the range [0, 1] is performed using the formula:

Where Xp is the vector of all X values.

**Section 4: Results**

The following section shows my results after running the KNN algorithm using K = 5 neighbours:

A picture containing monitor, screen, sitting, side

Description automatically generated

On both tests I picked 2 random samples of 20% of the dataset to use as the test set, meaning that 80% of the data was mapped onto the R2 plane in the training phase. The size of my dataset is small, so this leads to potential volatility of results between different sample choices. Also, I used k=5 neighbours as this is a typical value but at this point, I did not know the k range for maximum accuracy. In order to find this, I tested the algorithm for k values between 1 and 30 and plotted the %error of each test. From this I found that although the randomness of the test samples affects results, maximum accuracy is generally in the k range 5 <= k <= 10. Below are the plots of error rates of 90 tests of the KNN algorithm. The first and second plots show an expected trend, but the third plot show an ~3% error rate with k values in the expected accurate range. The last plot even shows maximum accuracy for k > 25! This variation in results is likely due to the potentially unbalanced random samples (due to size of dataset) which make up the training data.

I believe that my results were very good because of the clustered nature of the data I was working with. The iris plants have dimensions (particularly petal sizes) which are distinctive based on species. The KNN model worked very well as all the attributes are quantitative. The model can become less accurate with high dimensionality data or categorical attributes but the absence of these made it the optimal choice here. A mistake I made which affected my results at the start in was normalizing the testing dataset without incorporating the max and min values from the training set. This had the potential to make the test values inaccurate. The values taken by the iris dimensions are all in the [0,10]cm, so the algorithm’s results would not be skewed by a particular dimension so normalization may not have been necessary. The second plot below shows the results of 30 KNN tests without normalization. The trend in %error is close to what I expected. Random variations in error for (k < 5, k > 15) with higher accuracy in the range 5 <= k <= 15.

Chart, line chart

Description automatically generated

**Chart, line chart

Description automatically generated**

**Chart, line chart

Description automatically generated**