**A picture containing clipart

Description automatically generated**

**CST8390 - Lab 4**

*k* Nearest Neighbor (kNN)

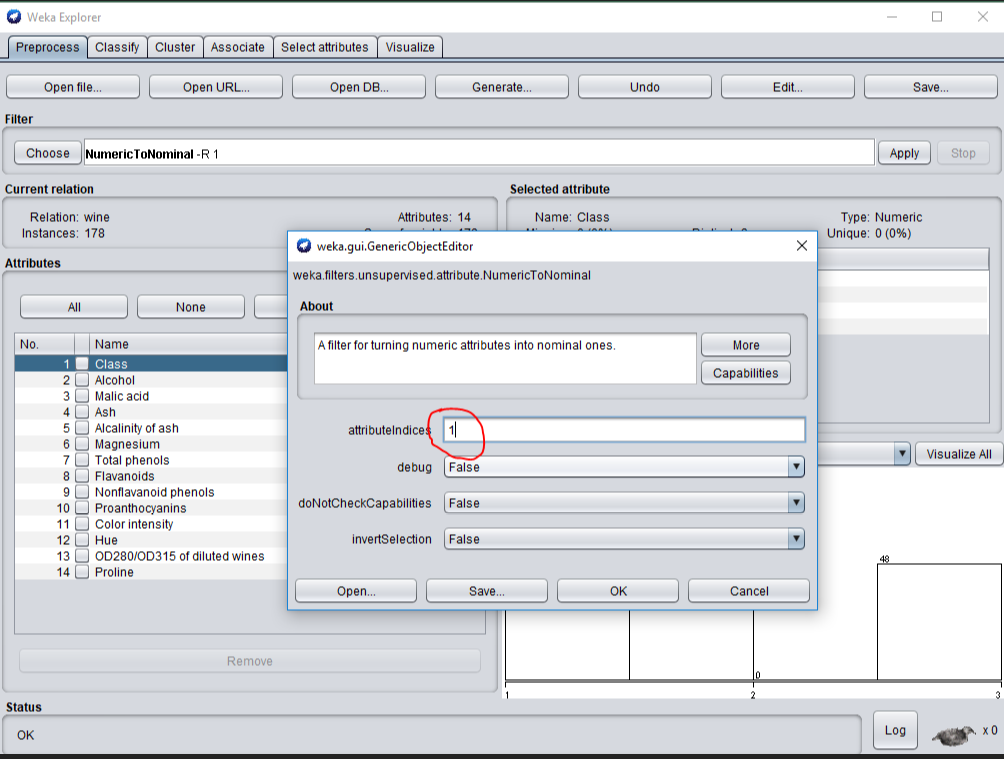
**Due Date:** Week 4 in corresponding lab sessions

**Introduction**

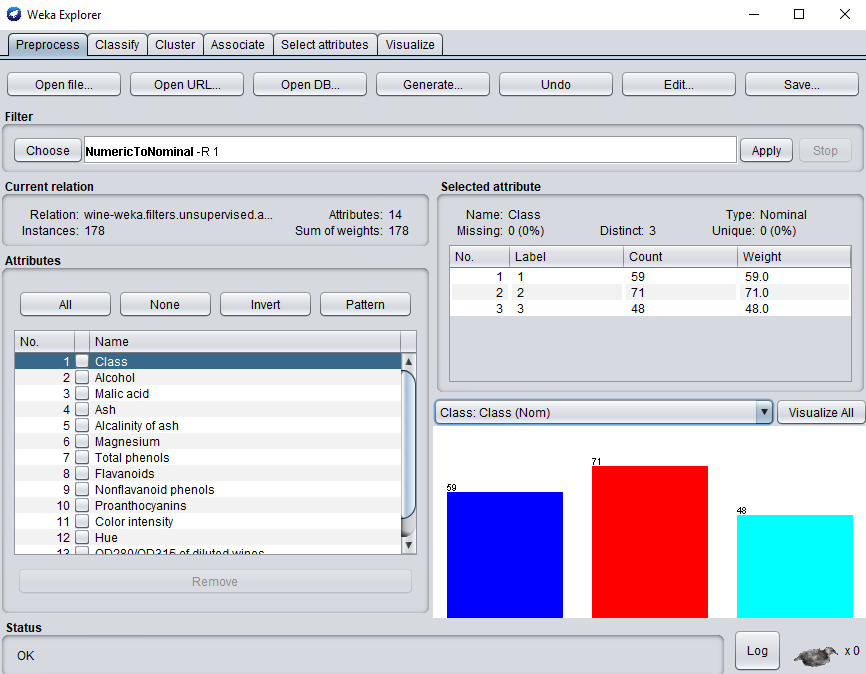
The goal of this lab is to perform classification on wine dataset using **kNN**.

**Steps:**

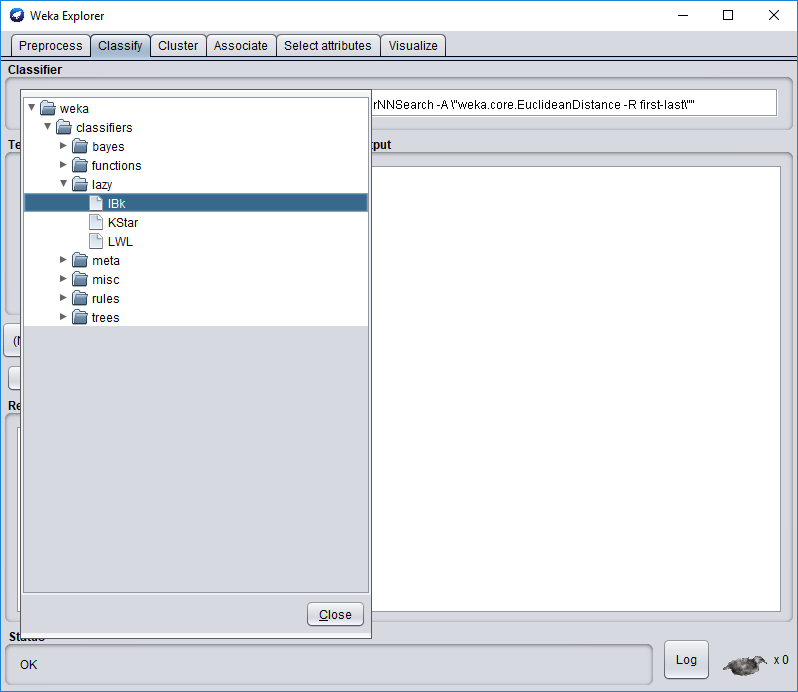
1. Get “Wine” dataset from <https://archive.ics.uci.edu/ml/datasets.html> (or <https://archive.ics.uci.edu/ml/index.php>) and save it as a **CSV** file. (**Data** is in Wine.data and info is in data.names). Add **attribute** names as the first row in the csv file. (For every row, first value is the class, remaining values are various attributes)
2. Explore and learn about the **relevance** of various attributes of the dataset
3. Load the file to **Weka**.
4. Check how various attributes are converted in **Weka**. Class is considered as **numeric** instead of **nominal**. Apply **filter** NumericToNominal to convert class datatype to nominal. When you apply filter, you need to specify the index of the attribute you need to apply the filter.



Now, you should see like this:

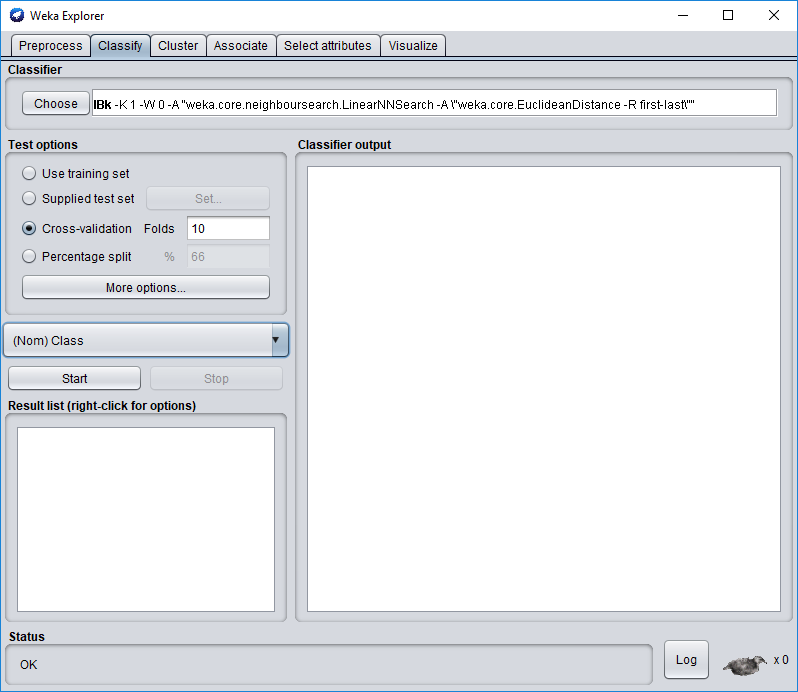


1. Now, we need to perform classification using kNN method. For that, click on “Classify” tab. For this lab, we use kNN. For that, choose IBk which is **Instance Based k Nearest Neighbors** from Lazy in the tree view.



1. As mentioned earlier, our first attribute is the class label. We need to set that now in the classify panel. (Marked in red below)

IBk –K 1 –W 0 –A “weka.core.neighboursearch.LinearNNSearch ….” This is the **parameter list** for the algorithm (Marked in blue). Click on this text to set the value of k. Set k as 3. Close the window. Now, set the cross-validation to 10 Folds if it’s not already there. Now click “Start” to run the algorithm.

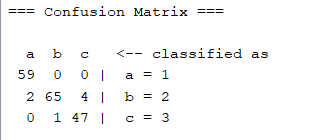


1. There should be a lot of text in the right-hand side of the window with the results of the algorithm. Find the line that says “Correctly classified instances”.
   1. What is the **percentage** of correctly classified items?
   2. What are the **True Positive (TP)** rates of each class?
   3. Look at the **confusion matrix**, which class is incorrectly classified?
2. Now click on the “Choose” button to modify the number of neighbours that are used in the **kNN** search to 5.
3. What is the **percentage** of correctly classified instances? 94.9438 %
4. What are the **True Positive (TP)** rates of each class? \_\_\_\_\_1.000, 0.873, 1.000\_\_\_\_\_\_\_\_\_\_
5. Look at the **confusion matrix**, which classes are incorrectly classified? \_\_\_\_\_\_\_b\_\_\_\_\_\_\_
6. Run the algorithm **several times**, always increasing the value of N by two, and always an **odd number**: 1, 7, 9, 11, 13. Each of your tests will be in the window of the lower left. Fill in the following table.

|  |  |
| --- | --- |
| K | percentage of correctly classified instances |
| 1 | 94.9438 % |
| 7 | 94.9438 % |
| 9 | 96.0674 % |
| 11 | 97.191 % |
| 13 | 96.6292 % |

Which class is being mis-classified? \_\_\_\_\_\_\_\_b&c number9\_\_\_\_\_\_\_\_\_\_





1. Repeat **step 9** with “Percentage Split” of 70. Fill in the following table.

|  |  |
| --- | --- |
| K | percentage of correctly classified instances |
| 1 | 96.2264 % |
| 3 | 100 % |
| 5 | 98.1132 % |
| 7 | 100 % |
| 9 | 100 % |
| 11 | 100 % |
| 13 | 100 % |

**REMEMBER:**

**Show your answers to the lab professor when you are done (in Weka and document).**

**This lab has 5 marks so ensure that you have all your answers filled in**.

***FOR YOUR ANALYSIS:***

*What is the purpose of “confusion matrix”? What is its importance?*

*Explain with your own words the kNN method.*

Ottawa, Jan 2020.

The purpose of “confusion matrix” is to show how do the class being classified, whether the classes are being classified correctly. The importance of “confusion matrix” is through analyzing data we can see the data is right or wrong, good or bad.

**What is KNN and how it works:**

Let’s head by setting some definitions and notations. We will take x to denote a feature and y to denote the target.

KNN falls in the **supervised learning** **algorithms**. This means that we have a dataset with labels training measurements (x,y) and would want to find the link between x and y. Our goal is to discover a function h:X→Y so that having an unknown observation x, h(x) can positively predict the identical output y.

在[模式识别](https://zh.wikipedia.org/wiki/%E6%A8%A1%E5%BC%8F%E8%AF%86%E5%88%AB)领域中，**最近鄰居法**（**KNN**算法，又譯**K-近邻算法**）是一种用于[分类](https://zh.wikipedia.org/wiki/%E5%88%86%E7%B1%BB%E9%97%AE%E9%A2%98)和[回归](https://zh.wikipedia.org/wiki/%E8%BF%B4%E6%AD%B8%E5%88%86%E6%9E%90)的[無母數統計](https://zh.wikipedia.org/wiki/%E7%84%A1%E6%AF%8D%E6%95%B8%E7%B5%B1%E8%A8%88)方法[[1]](https://zh.wikipedia.org/wiki/K-%E8%BF%91%E9%82%BB%E7%AE%97%E6%B3%95#cite_note-1)。在这两种情况下，输入包含[特徵空間](https://zh.wikipedia.org/w/index.php?title=%E7%89%B9%E5%BE%B5%E7%A9%BA%E9%96%93(%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92)&action=edit&redlink=1)（Feature Space）中的***k***个最接近的训练样本。

* 在*k-NN分类*中，输出是一个分类族群。一个对象的分类是由其邻居的“多数表决”确定的，*k*个最近邻居（*k*为正[整数](https://zh.wikipedia.org/wiki/%E6%95%B4%E6%95%B0)，通常较小）中最常见的分类决定了赋予该对象的类别。若*k* = 1，则该对象的类别直接由最近的一个节点赋予。
* 在*k-NN回归*中，输出是该对象的属性值。该值是其*k*个最近邻居的值的平均值。

最近鄰居法採用向量空間模型來分類，概念為相同類別的案例，彼此的相似度高，而可以藉由計算與已知類別案例之相似度，來評估未知類別案例可能的分類。