Machine Learning Assignment - Documentation

Jake Bolger – C18395341

**CA2 – Build a Classifier**

The data that was given to build the classifier for this assignment was data from a bank marketing campaign. The classification goal was to predict whether a client will subscribe to a term deposit after they have been contacted by phone. These predictions were then to be outputted into a text file in the form of “TypeA”, or “TypeB”. This classifier was to be designed using the ‘queries.txt’, ‘datadescription.txt’, and ‘trainingset.txt’ files.

**How I solved the problem?**

First, I selected what classifier I was going to use. The **KNN** classifier was chosen which will be discussed in detail in the next section. Next, I started coding the solution and classifier in ‘Jupyter Lab’.

Below are the process and steps I took when writing my program:

* First, I imported all the relevant python libraries and operations from libraries like ‘sklearn’ etc.
* Next, I used the datadescription.txt file to set all the feature headers.
* Here are all the Feature names that were used:
* Id, age, job, marital, education, default, balance, housing, loan, contact, day, month, duration, campaign, pdays, previous, poutcome, Output.
* I then read in the trainingset.txt file and set the headers of the categorical table and exported them to a csv file
* I did the same for the numerical features.
* Next, I made a csv file with all the outputs.
* I then replaced all the empty values ‘?’.
* The numerical and categorical features were then combined.
* The data was then stored in a file called ‘myData.npy’.
* The ‘features\_file.csv’ with all the outputs features was read in.
* Then, the KNN classifier was imported from sklearn.
* An array to hold the data was created and then the KNeighborsClassifier() was created and integrated into the model using ‘fit’.
* The predictions were then created.
* An instance was made for the KNN classifier, it was added and saved.
* The ‘queries.txt’ file was read in, and the feature names were added to the headers, and it was exported to a csv file called ‘queriesHeaders.csv’.
* The same process for the ‘trainingset.txt’ was completed for this file.
* The outcomes were put into ‘outcomes.txt’ and were used to create the ‘solutionFile.txt’ with all the predictions.
* This file was in the format of:

**TEST1,”TypeA”**

**TEST2,”TypeA”**

**TEST3,”TypeB”**

…

…

**Why did I choose the classifier?**

When building the classifier, the K-Nearest Neighbours algorithm was used on the model. This was chosen because the algorithm is one of the simplest in machine learning for classification. This classifier uses the data and classifies based on the problem being solved such as using datasets to predict outcomes which was perfect for this assignment.

There were many advantages to using this classifier. Whilst being able to read through the whole dataset and being simple to implement this classifier has no assumptions and no training step which means there are no data assumptions you must meet before the model can be implemented. This classifier is also memory based so it works well when collecting data.

Although this is a good classifier there were some disadvantages that I noticed. As I used more data, KNN became a bit slow when it was calculating count variables, plotting graphs and accuracy statistics.

I noticed that a huge portion of the predictions were ‘TypeA’. When using KNN the model normally gives preference to the more common variable which made ‘TypeB’ very uncommon.

**Issues with the data and how were the issues handled?**

**Missing Values:**

When handling the data, I noticed that there were empty values through the datasets. This was true for both the categorical and numerical data. to fix this issue I replaced all the empty slots with ‘NA’.

Shown below:

Graphical user interface, text

Description automatically generated

**Outliers:**

Another issue that occurred that was mentioned before was imbalanced data. where most of the data was ‘TypeA’ so ‘TypeB’ was uncommon. To check if this was a major error or not, I tested the accuracy of the data. Document error and maybe use different classifier in future.

**How testing was performed?**

To test the data using KNN I decided to use a plot diagram which counts and tests the accuracy of the data. I printed out some of the accuracy outputs. Below are a sample:







The accuracy of the model using KNN was around **0.88** most times. When researching I found that when using models like decision trees, the accuracy tends to be lower than this. For example, projects like this would have an accuracy of **0.83** instead. The ROC plot below was used to represent how accurate the tests were and how accurate the predictions were. Normally the closer the curve is to the 45-degree line the less accurate the predictions are. This seems to be the case for this classifier.

For future models maybe another classifier would be a more suitable choice.

Chart, line chart

Description automatically generated

X

Y