Machine Learning CA2: Build a Classifier

The classifier model

The classifier model I choose was the decision tree algorithm. I choose this for a number of reasons. Firstly, it is a model that I am comfortable with as we worked with it earlier in the course, so I had more time to understand how it works. It is a model that is suited for the problem as it can handle both numerical and categorical data relatively simply. It does not require data normalisation also unlike KNN where results can be greatly improved when the data is normalised.

One problem with using a decision tree is that it may succumb to overfitting but as there is only two possible outcomes for this classifier, I believed this may not be too much of an issue.

Issues with the data

I did not find any particular issues with the data. The '?' in the column for y is the only part of the data that gave any problems and this was simply fixed by converting it to 'NA'.

The Algorithm

First all the importing is done, this includes pandas, NumPy and sklearn. Then the training set is loaded in. The target feature (y) must be passed in as a separate parameter so this is done next. Some pre-processing must be done to the dataset, the numerical features, id, and y are all dropped to allow for the categorical features to be processed.

The categorical features are converted to numeric encoding by using a vectorizer, and then these values are mapped appropriately to the dataset. These values are then merged back with the numerical features that were dropped earlier.

At this point, the decision tree is created. It is created using entropy-based learning and then it is fitted using the numeric representations of the training model.

Now that the model is created, queries.txt can be read in as the testing data. Like before, the numerical features of the testing data are dropped to allow for the categorical features to be converted to numeric encoding using a vectorizer. The features are then merged back together again like before.

Now the predictions can begin. A list is initialised to store all the predictions being made. The predictions are made in a for loop and each prediction is added to the list. All the predictions are then added to the y column in the testing data data-frame. These 'raw' predictions are then outputted to a csv file with their respective id. These 'raw' predictions are in the wrong format for the desired output so some of the characters must be replaced and this is done next. After replacing these characters, the file 'predictions.txt' is written and saved.

Finally, the user is notified of the algorithm finishing and the final 5 rows are displayed to the user.