

Artificial Intelligence 2

Continuous Assignment 2

DT282

BSc in Computer Science International

**Amandeep Singh & Vardaan Sharma**

**C14712039 & D14122804**

School of Computing

Dublin Institute of Technology



For this Assignment, both of us first sat down and looked at the data set to choose the best algorithm for this application. We were provided with three files in total which included a "datadescription.txt" file which included all the headers and descriptions, "queries.txt" file which contains all the query instances, it has all the data in it, but the target column is filled with '?'. The last file we looked at which was provided was the "trainingset.txt" file, which is also a data file but is complete and will be used to train the classifier.

To start with the assignment, we had to organise and categorise the training set that was provided to us. Using our knowledge form the last assignment we added the headings of the data respectively from the "datadescription.txt" file and separated the data into Categorical and Continuous features. A Continuous feature is a feature that can have an infinite number of possibilities, examples of this are height, weight and age. In this data set, the Continuous features were as follows:

* id, age, balance, day, duration, campaign, pdays, previous

However, Categorical features are a feature that can take one of the limited options available, examples of this would be gender, marital status or month. The Categorical features in this dataset are:

* job, marital, education, default, housing, contact, month, poutcome, y

After the data set was split categorised the data was cleaned. If the dataset included an 'unknown' feature or '?' Question mark symbol was replaced by 'NA'. The data was then split into training and testing sets. This was an 80/20 split, 80% was the training set, and the remaining 20% was the testing. The training set was used for training the classifiers, and the testing set was then used to make predictions and compare the predictions made by the classifier to the actual data.

After this, we trained an ID3 classifier from the Scikit library using the training data. We ran the classifier a number of times and got an accuracy around 0.83 every time.

We then tested the dataset with another algorithm just to compare. This was the K Nearest Neighbours (KNN) algorithm. We first used the cross-validation with the training set and the KNN classifier so that we could get the best number of neighbours. The optimal number of neighbours that we got was 41. After that, we used a Scikit library to check it and got an accuracy of around 0.88 in each round of training and testing.

Since we decided to implement the Iterative Dichotomizer 3 (ID3) algorithm and the K Nearest Neighbours (KNN) algorithm to see which one was more accurate; we were able to choose one algorithm than to train the classifier. As you can see above the accuracy of the KNN if rounded up is 88% whereas ID3 if rounded is only 83%. Using this we chose to implement KNN to train the classifier. But, this wasn't the only reason why we decided to implement the KNN model, there were many advantages to applying this model to the ID3 model. The KNN model was suitable for large training sets, as the trainset provided had over 24 thousand lines of data. Also, the KNN model is robust to noisy data. Since KNN can handle noisy data well, we didn't have to take out columns with missing data such as the contact and duration column.

In saying that, it is true that the ID3 may be faster than the KNN model because it uses a "divide and conquer" approach since it prunes down the tree and it uses less memory than the KNN model, the accuracy rate was lower than the KNN. This was due to different services both algorithms provide. The ID3 algorithm is supervised learning and used for classification. The KNN model, on the other hand, uses clustering, with this it uses statistical concepts to split datasets into subsets that have similar features. So by testing both the algorithms both, myself and my team member decided to choose the KNN algorithm due to the higher accuracy.

After choosing KNN, we were able to test it with the "queries.txt" file. This time though we trained the KNN classifier again, but this time only with the training set rather than a split of the training set and testing set. We then ran the queries from the "queries.txt" file and put the prediction into a prediction file. The Id's were split into "Type A" or "Type B" as per the prediction. We saved the predictions in a file called "predictions.txt".