**Assignment 2 – Build a Classifier**

**Vlad Zat**

**C14714071**

**Classifiers**

Three classifiers were tested for this project: Decision Tree, K-Nearest Neighbours, and Naïve Bayes. The best results for the Decision Tree was using the entropy coefficient. For the K-Nearest Neighbours, a good result was generated using 11 neighbours. The Naïve Bayes classifier was using a Gaussian process.

The accuracy of the classifiers was around 83%, 88%, and respectively 83%. The accuracy of the K-Nearest Neighbour and Naïve Bayes classifiers varied a lot around this number, while the Decision Tree was more stable. Another difference was with the percentage of *TypeA* and *TypeB* results generated. In the training set, approximately 11% of the total rows are of *TypeB*. The Decision Tree generated about 12% *TypeB* rows, while K-Nearest Neighbours generated 6% and Naïve Bayes even fewer. In a randomly occurring set, the ratio of the results should be similar. Because of the two reasons mentioned previously, a Decision Tree was used as a classifier.

**Testing**

In order to test each classifier, the initial set was split into two subsets: a training set and a testing set. The testing set consisted of 40% of the total records. The three classifiers (Decision Tree, K-Nearest Neighbours, and Naïve Bayes) were trained using the training set, and then used to predict the values from the testing set. A normalized accuracy score was then created using the function *accuracy\_score* from the *metrics* class. The results were 83% for the Decision Tree, 88% for K-Nearest Neighbours, and 83% for the Naïve Bayes classifier.

**Data issues**

To improve the classifier’s training, some modifications were made to the data. One issue discovered was that the *duration* column had a cardinality of one. The only value present in the field was 0. Because so much data was missing, this column was removed. The same was done to the id field as it only contains unique fields which are not useful in classifying data.

Another change was made to the *balance* column. This column contained some high outliers which could potentially skew the result. In order to fix this, the maximum value of the balance was clamped to the mean value of the column plus two times the standard deviation.

The final change was made to the columns contains categorical values. If the percentage of missing data from a column is less than 10%, then the values were replaced with the mode of the column, which is the most frequent value.

**Sklearn**

This classifier was implemented using *sklearn*. It provided the necessary functions to train the classifier, predict data, and test the accuracy of the data. One problem encountered with this library was that it does not accept categorical values. To fix this, these values were converted to numeric data. This was done using *DictVectorizer*. This was done by firstly converting the categorical columns to a dictionary, which was then transformed to numeric values using the *fit\_transform* function from a *DictVectorizer* object. These values, together with the numeric columns, were then combined into a single data frame. This was done using the *hstack* function from the *numpy* library, which horizontally stacks multiple arrays.

**Predictions file**

Another problem encountered was adding double quotes to the second column in the predictions file. While the *to\_cs*vfunction from *pandas* allows the ability to control the amount of quoting to add, the result given by multiple types was to either quote both columns or neither. To fix this problem, double quotes were manually added to the predictions column, and no quotation was added from the *to\_csv* function. This was done by setting the quoting parameter to *QUOTE\_NONE*, which is found in the *csv* library.