MACHINE LEARNING ASSESSMENT 2 – BUILD A CLASSIFIER

Students:

Jack Duggan - C16350866

Buchita Gitchamnan - C16348651

Task description:

In this assignment you will develop a classifier that uses data to predict the outcome of a bank phone marketing campaign. The classifier model has to be one of those studied in this course (you can develop it yourself or use scikit-learn python library)

Description of the data:

Trainingdataset.csv:

Categorical:

- Poutcome - outcome of the previous marketing campaign.

  This categorical feature has a cardinality of 4. 81% of this feature is unknown this has led to the decision to drop this feature.

- Contact - contact communication type.

  This categorical feature has a cardinality of 3. 30% of this feature is unknown at a glance we thought we could imputate this data. But because 1st mode (cellular) has a value of 65% we decided to drop this feature.

- Job - type of job

  This categorical feature is missing .7%. Using imputation, we filled in these values.

- Education

  This categorical feature is missing 4%. Using imputation, we filled in these values.

Continuous:

All continuous data seems to be in order. There are no missing values, irregularities in min/max and cardinality values.

Queries.csv:

The same was true for this data set as its training set. The same cleaning should take place on this dataset.

Prepare data:

To prepare the data, we dropped the “poutcome” and “contact” features. Then we encoded the categorical values using a label encoder. Finally, imputated missing feature values below 20%.

Classifier choice and testing:

In choosing the classifier for this problem we looked at naïve bayes, knn and logistic regression. To split the data into training and testing sets we used Train_test_spilt(), 80% training and 20% testing.

Over an average of 10 runs, NB gave us an average accuracy of 82.87%, LR gave us an average accuracy of 88.35% and KNN gave us an average accuracy of 88.33% against the testing set.

When running these algorithms against our actual data, LR out putted a file that was all no and NB out putted a file that has 75 yes. Therefore, we believe that LR is overfitting the data. The value we use for K in the KNN algorithm has a large effect on the output of this classifier choosing the wrong value can lead to under or over fitting. n_neighbors should be odd to give us a tie breaker. Because of this we chose to use naïve bayes.