# Machine Learning for Data Analytics Assignment 2

Student name: Chenxi Zhang, Jianyu He

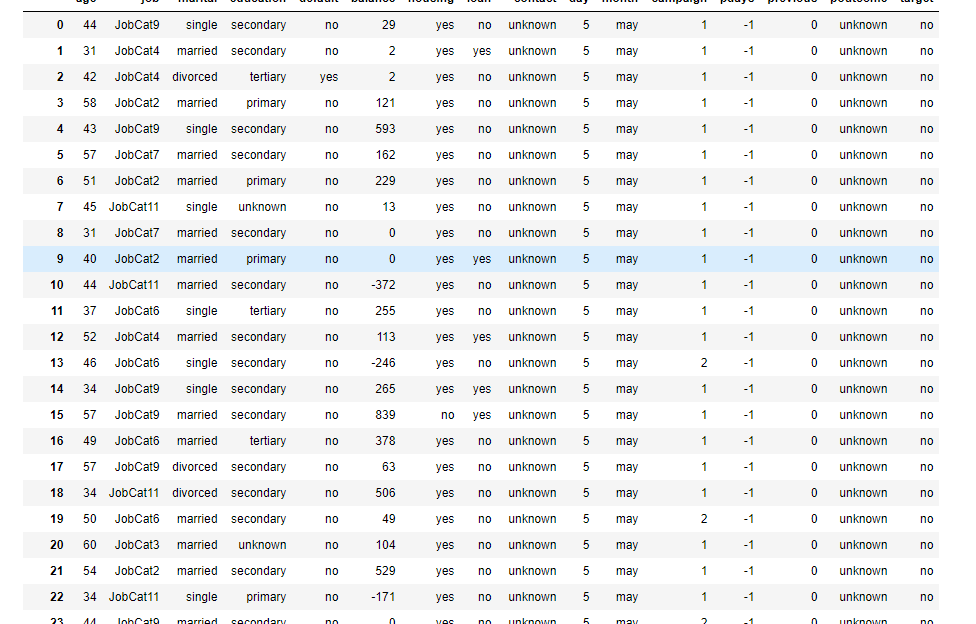
Student number: C16434996, D17124591

## Data Exploration & Cleaning

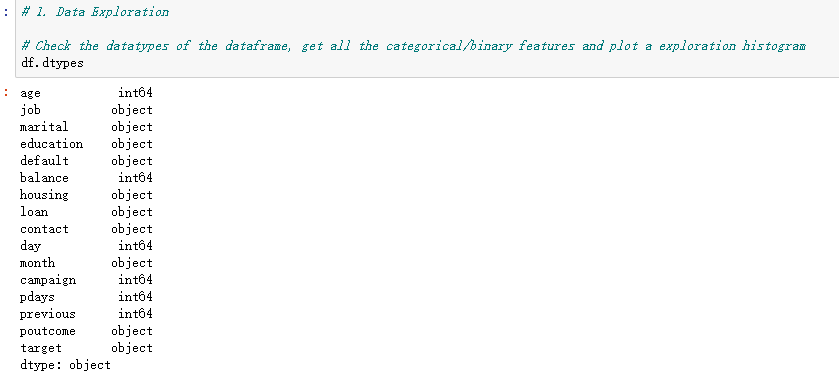
Set initial header to each column for accessibility:



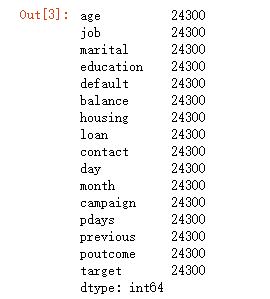
Firstly, we investigated on the data that have been loaded into the code and try to spot some irregular/suspicious data that need to be fixed.



Next is to check the datatypes of the columns, to check which of the fields are objects and store the headers of those columns. That will be used later for plotting the graphs and one hot encoding:

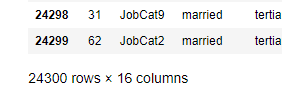


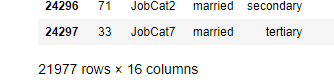
After that is to check for missing values, by using the count() function, no missing values were identified:



From a deeper analysis of the data by plotting them onto a graph, we have spotted some outliers in the ‘balance’, ‘pdays’ have many -1 values and many 0 values in the ‘previous’ column. On the categorical data, ‘poutcome’ has some high occurrence of ‘unknown’ which was one of the categories explained in the data description text file, this should be investigated or truncate the data to a smaller scale to prevent dominance.

By research on dealing with outliers, z-score was found to be the easiest and efficient way to clean the outliers from the data, it set a threshold value of 3 among all the rows of the dataset, by calculating a score for each row, it deduct the outlier rows and leaving the data with better consistency. Below is the result of the z-score:

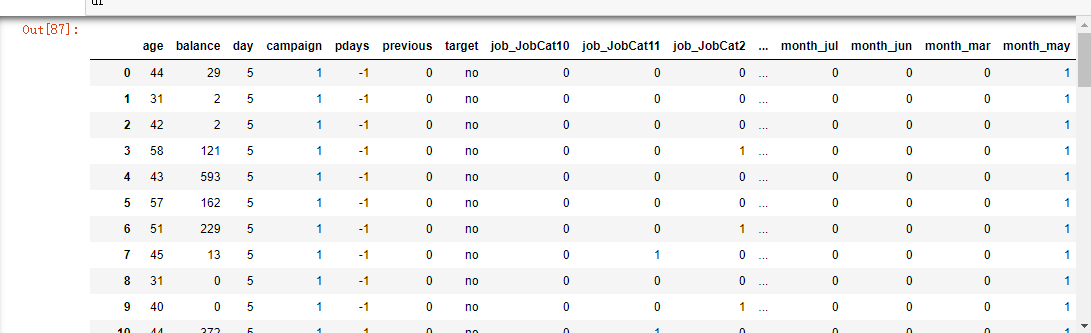
 Before cleaning up

 After cleaning up

Now the data is cleaned up, we can do some preprocessing of the data, first by one hot encoding, we could turn all the categorical features to a matrix with each category as a column, this is achieved by using the pandas get\_dummies() function

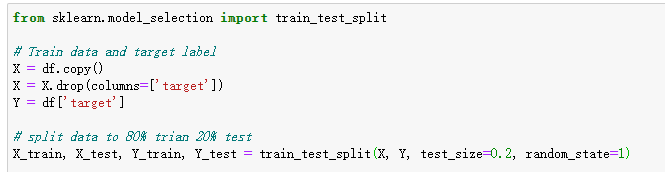


And we use df to print out the new dataframe

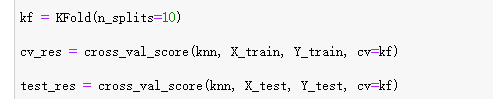


## Building a classifier

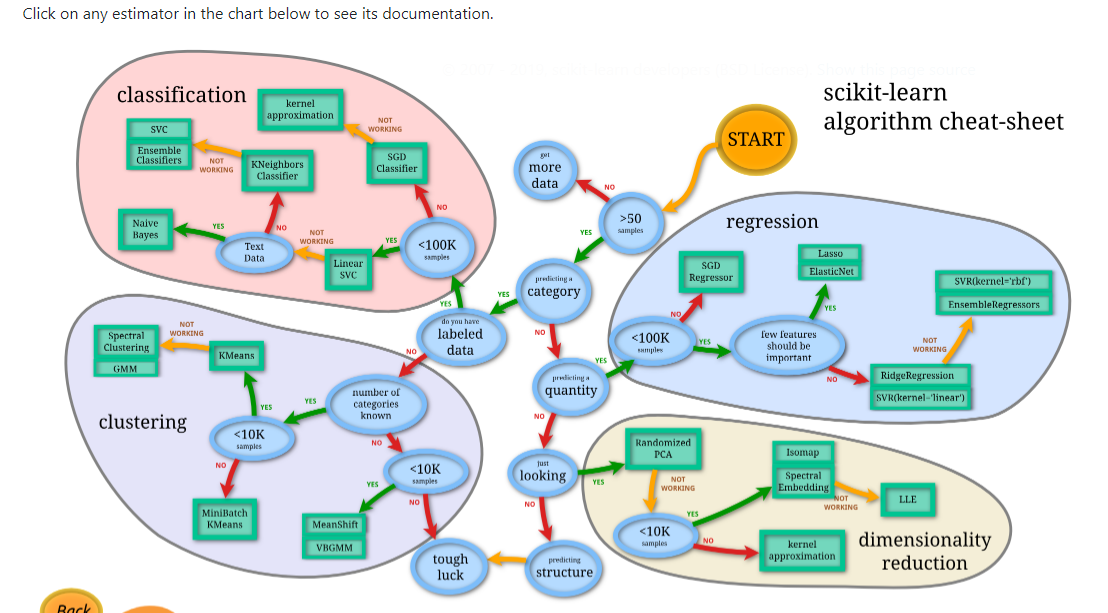
After the preprocessing has done, we can now split the data by train\_test\_split() function, this is to split the dataframe into training data and test data, from best practices, we split the data of 80% training and 20% testing.



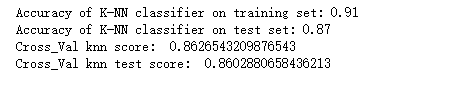
This is one of the mechanism for splitting the data for training, we have also tried the k-fold cross validation method:

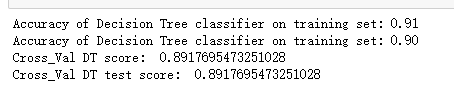


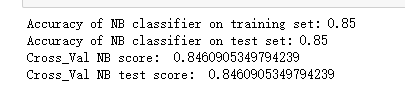
When the data is ready to be fitted to a machine learning classifier, we have checked what was good for our purpose on sklearn’s diagram of machine learning algorithms.

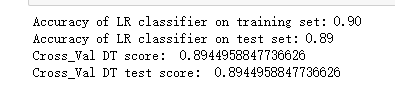


Based on what we have learned in the lecture, we tested Decision Tree, Naïve Bayes, K Nearest Neighbor and Logistic Regression. This is the outcome we got from different classifiers:



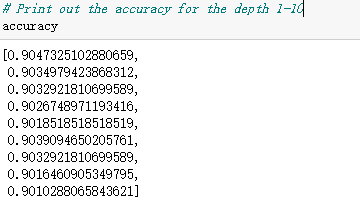






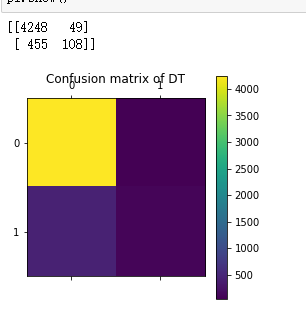
To ensure we use the correct algorithm before predicting the query set, we have tested and run the program many times to tweak the number of neighbors of KNN, the depth of the decision tree, number of folds and random seed of the train test data.

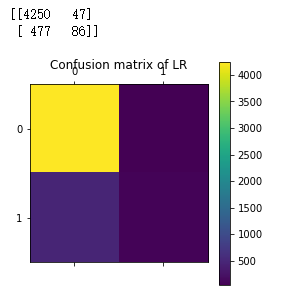
An accuracy testing of decision tree based on depth is shown below:



Based on the testing, two classifiers were found to be the best solution, that is Decision Tree and Logistic Regression.

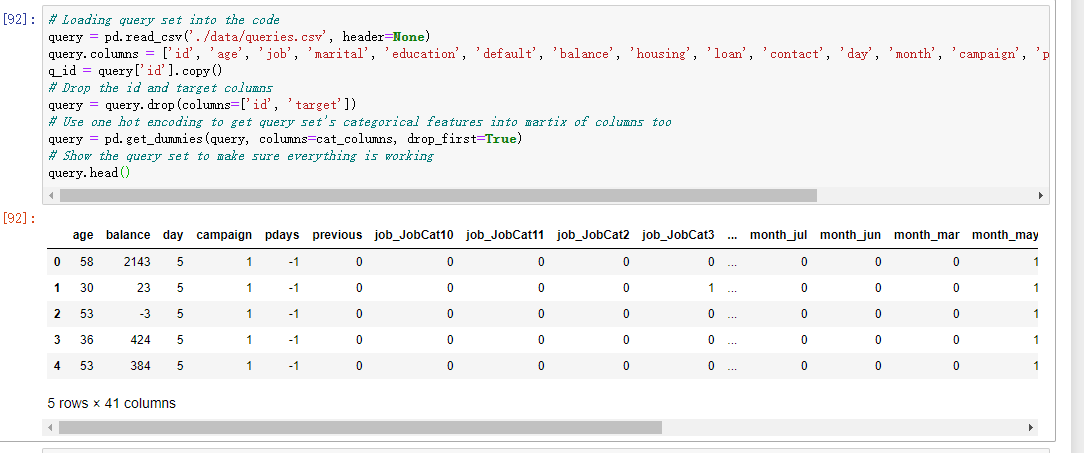
We have created a confusion matrix to check what are the TP, FP TN and FN of the model:



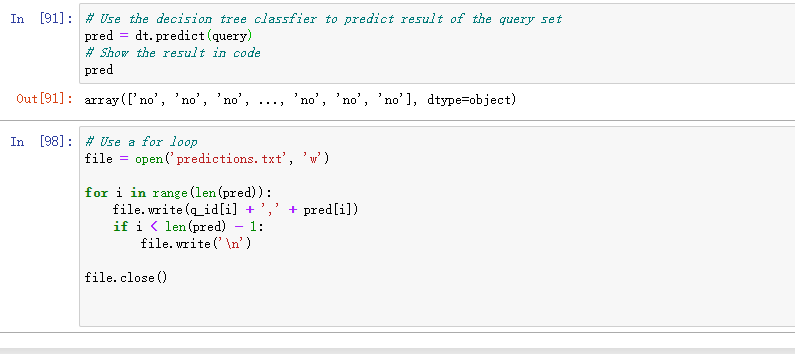


Based on the confusion matrix result, we have chosen decision tree to be our classifier because it predicted more TN than Logistic Regression, this model will be more viable when predicting that kind of queries.

Finishing up on the assignment, we have loaded the query set into the code, by keeping the id column and dropping the id and target column, using one hot encoding to get the queryset categorical features to a matrix, the result is as follows:



The next step is to use the decision tree to predict the outcome of the query set, this is done by the classifier.predict() function. As we got our result for predictions of query set, using Python’s open() function to write a text file following the assignment brief’s formatting of ‘TEST<n>,<result>’



Output of the prediction.txt file:

