ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

MSc in Business Analytics

Statistics for Business Analytics II

Project II

Taklakoglou Chidiroglou Argyrios

ID: f2822114

Contents

Classification

The Bank marketing dataset has 39883 rows with 21 variables. The response attribute is the SUBSCRIBED variable, denotes client subscribed to term deposit or not (yes or no). The goal is to build predictive models to classify whether a client will buy or not the new product.

Dataset:

This campaign were based on phone calls. Often, more than one calls were done to the same client to access if their product "term deposit" will be subscribed (yes) or not subscribed(no).There are 20 input variables and 1 output variable (desired target). The dataset had different types of client data like age, job, martial, education, default, housing, loan, contact, month, day_of_week, duration, campaign, pdays, previous, poutcome, em.var.rate, cons.price.idx, cons.conf.idx, euribor3m, nr. Employed and one output variable SUBSCRIBED that denotes if client subscribed to term deposit or not. These dataset attributes denote customer data, socio-economic data, telemarketing data and some other data.

- # bank client data:
    1. age: Client Age (should be numeric)
    2. job : Type of job (should be categorical: 'admin.','bluecollar','entrepreneur','housemaid','management','retired','selfe mployed', 'services,'student','technician','unemployed','unknown')
    3. marital : marital status (should be categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
    4. education (should be categorical: basic.4y','basic.6y','basic.9y', 'high.school','illiterate','professional.course', 'university.degree','unknown')
    5. default: has credit in default? (should be categorical: 'no','yes','unknown')
    6. housing: has housing loan? (should be categorical: 'no','yes','unknown')
    7. loan: has personal loan? (should be categorical: 'no','yes','unknown')

- # related with the last contact of the current campaign:
    8. contact: contact communication type (should be categorical: 4 'cellular','telephone')
    9. month: last contact month of year (should be categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
    10. day_of_week: last contact day of the week (should be categorical: 'mon','tue','wed','thu','fri')
    11. - duration: last contact duration, in seconds (should be numeric)

- # other attributes:
    12. campaign: number of contacts performed during this campaign and for this client (should be numeric, includes last contact)
    13. pdays: number of days that passed by after the client was last contacted from a previous campaign (should be numeric; 999 means client was not previously contacted)

14. previous: number of contacts performed before this campaign and for this client (should be numeric)
15. poutcome: outcome of the previous marketing campaign (should be categorical: 'failure','nonexistent','success')

# social and economic context attributes

16. emp.var.rate: employment variation rate - quarterly indicator (should be numeric)
17. cons.price.idx: consumer price index - monthly indicator (should be numeric)
18. cons.conf.idx: consumer confidence index - monthly indicator (should be numeric)
19. euribor3m: euribor 3 month rate - daily indicator (should be numeric)
20. nr.employed: number of employees - quarterly indicator (should be numeric)
21. SUBSCRIBED - has the client subscribed a term deposit? (should be binary: 'yes','no')

By exploring the data, I observe that the majority of the variables are not defined correctly. So I change the data type of variable: 1. age, duration, campaign, pdays, previous, emp.var.rate, cons.price.idx, euribor3m and nr.employed to numeric. 2. job, marital, education, default, housing, loan, contact, month, day_of_week, poutcome, and SUBSCRIBED to factor. I also noticed that in pdays columns there the value 999 doesn't make any sense, so I replaces the value 999 with the value 0. I make this change because the value 999 is an outlier and in may effect some methods either in classification or in clustering.

To begin with, I loaded the dataset in R Studio and checked for any missing values using `is.na` function and found that it didn't have any missing values. So, we have a clean dataset. Then I proceed to data scaling. Scaling is a method used to normalize the range of features of data. In data processing, it is also known as data normalization. This means that I transform the data so that it fits within a specific scale because both in clustering and in classification in some methods(like support vector machines, or SVM or k-nearest neighbors, or KNN.) which are based on measures of how far apart are the data points, there will be a problem and won't be executed correctly.

In the first part, the goal is to make 3 models for Classification. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y)

The next step is to split 70% of the data in training, 15% in testing and 15% in validation. I use training dataset to train the models. Testing dataset to test how well the model predicts(in sample) and observe some metrics I am interesting in and validation dataset to test how well the model predicts(out of sample). A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model’s hyperparameters.

The validation dataset is different from the test dataset that is also held back from the training of the model, but is instead used to give an unbiased estimate of the skill of the final tuned model when comparing or selecting between final models.
I used the caret library to implement the models. This library contains ‘trainControl()’ function in order to modify the resampling method. The option method controls the type of resampling. I used method='repeatedcv' because this option is used to specify repeated K-fold cross-validation. The argument ‘repeats’ controls the number of repetitions. I used `repeats=3`. K is controlled by the ‘number’ argument. I used `number=10`. Also, in order to choose different measures of performance, additional arguments are given to ‘trainControl()’. The `summaryFunction` argument is one of them. It is used to pass in a function that takes the observed and predicted values and estimate some measure of performance. I used `summaryFunction = twoClassSummary` because this option is used to compute measures specific to two-class problems, such as the area under the ROC curve, the sensitivity and specificity. Since the ROC curve is based on the predicted class probabilities another option is required. This option is the `classProbs = TRUE` which includes these calculations. Caret library has another option called ‘search’ in which you define the way that the hyperparameters for the model will be chosen. I used search =’grid’ because the aim is to choose good hyperparameters in order to have a good model for prediction. In train() function I use `search = ‘ROC’` because ROC looks at various probability cutoffs and gives probability cutoff at which accuracy will be the best.

The metrics we are interesting in are the:

- Precision = TP/TP+FP
- Recall = TP/TP+FN
- F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Where:

True Positives (TP) are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.
True Negatives (TN) are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.
False Positives (FP) – are the wrongly predicted positive values which means that the actual value is no and predicted value is yes
False Negatives (FN) – are the wrongly predicted negative values which means that the actual value is yes but predicted class in no.

The first method I used is the Naive Bayes classifier. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being

classified is independent of each other. Bayes' theorem is also known as Bayes'
Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where:

- P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
- P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
- P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
- P(B) is Marginal Probability: Probability of Evidence.

After the implementation of the model, I use the command ` predict()` to test the predictive ability of the model. The aim is to know the models' prediction ability in unobserved data. For this reason, I made the validation dataset. The frequency of the dataset is 10% for the 'yes' characteristic. That's the reason why we don't see accuracy. Because a model with 90% accuracy won't be good. It may has predicted that each value in the target variable is 'no'. As a result, will be 90% accuracy but the recall will be 0. In order to observe the output I execute `confusionMatrix()`. So I get the following results.

Precision	0.41075
Recall	0.48913
F1	0.44653

By using the `tuneGrid()` command to test different hyperparameters, the results were worse than the above, so the best model with Naïve Bayes algorithm was the one with the auto grid search.
From the above table can be inferred that Naïve Bayes Model it's not reliable. We can't be sure about the predictions of this model.

The next model I implement is the K Nearest Neighbor (KNN). This algorithm KNN aims for pattern recognition tasks. KNN works on a principle assuming every data point falling in near to each other is falling in the same class. In other words, it classifies a new data point based on similarity. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification).

Again the best model that was implemented, was the one with the option = 'grid' and not the models I created with 'tuneGrid' in which I tried to find the best k. In order to observe the output I execute `confusionMatrix()`. So I get the following results.

Precision	0.61184
Recall	0.33696
F1	0.43458

From the above table can be inferred that K Nearest Neighbor Classifier it's better from Naïve Bayes Classifier if we want to predicts the 'yes' but if we want to predict 'no' it's worse than Naïve Bayes Classifier. We can't be sure about the predictions of this model.

The last model I implemented, was the Random Forest. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

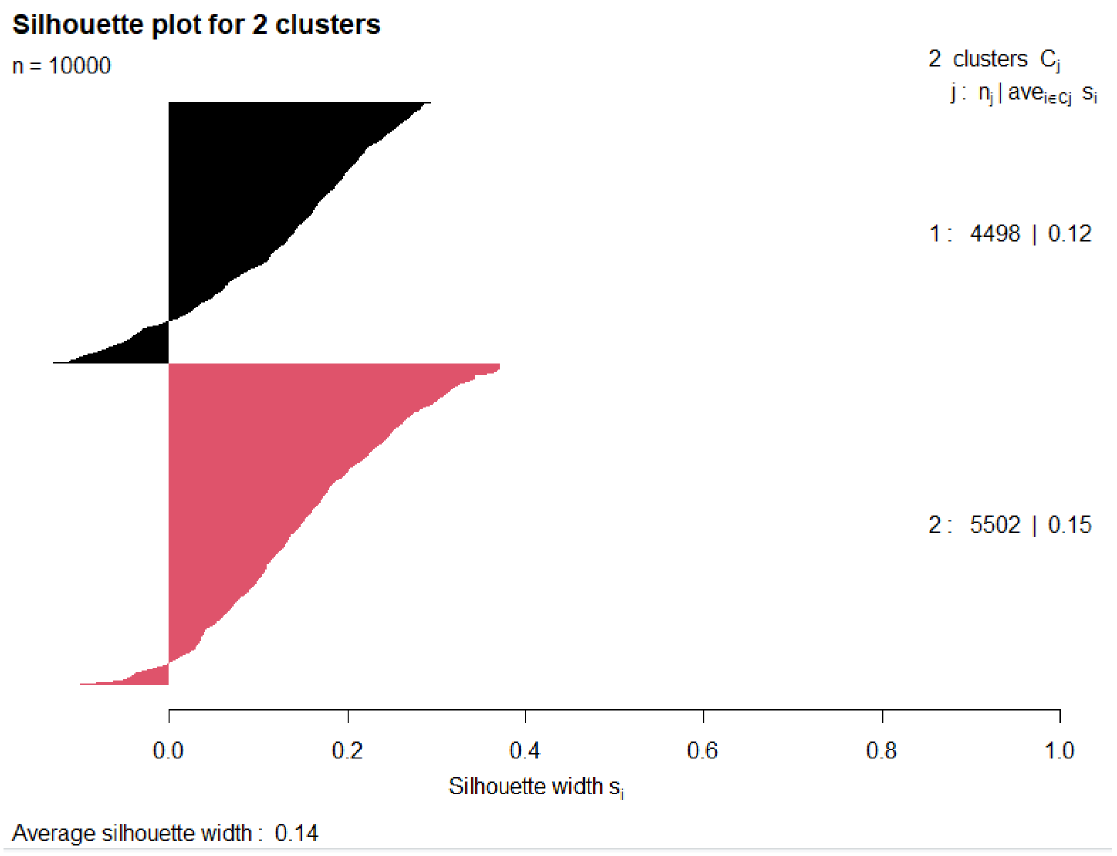
Again the best model that was implemented, was the one with the option = 'grid' and not the models I created with 'tuneGrid' in which I tried to find the best hyperparameters such as the number of trees. . In order to observe the output I execute `confusionMatrix()`. So I get the following results.

Precision	0.60684
Recall	0.51449
F1	0.55686

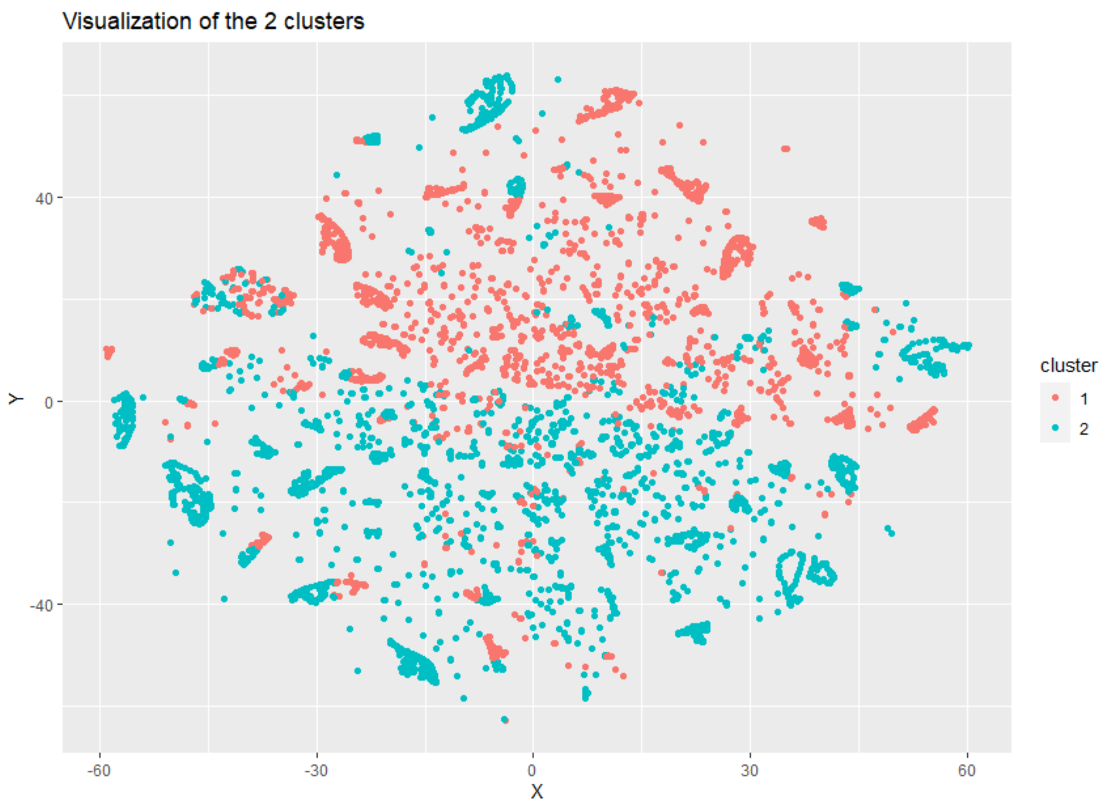
From the above table can be inferred that Random Forest is the best model from the three and it's more reliable.

In the second part the goal is to cluster the clients based on their characteristics. Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). In simple words, the aim is to segregate groups with similar traits and assign them into clusters.
To begin with, I create a new dataframe containing only the 11 variables for the clients. Due to the fact that clustering demands a distance matrix, it's impossible to compute it with large amount of data. For this reason, I took a sample of 10000 observations.
The next step is to import cluster library in order to use daisy function to calculate gower distance. I used gower distance because the dataframe I am working on, contain mixed data. Gower Distance is a distance measure that can be used to calculate distance between two entity whose attribute has a mixed of categorical and numerical values. The next step is to use pam() functon with the matrix of gower distance for the number of clusters I want. use pam() functon can be used to compute PAM algorithm. The simplified format is pam(x, k), where "x" is the data and k is the number of clusters to be generated. I used several times pam() function because I wanted to see the silhouette value and visualize the clusters in order to select the ideal number of clusters for this problem. The silhouette coefficient is a measure of how similar a data point is within-cluster (cohesion) compared to other clusters

(separation). So, after the pam() I use silhouette() and plot() functions to calculate and visualize the silhouette value for the clusters that have been created.



By comparing the silhouette values for 2 up to 7 clusters I saw that the value for 2 clusters was the same with the value for 3 or 4 clusters and the value for 5,6,7 clusters was slightly lower. So I choose the 2 clusters for parsimony reason.
Let's visualize the clusters in order to observe whether the clusters are well separated or not.



So by observing the visualization and the silhouette value I can say that the 2 clusters haven't been separated appropriately.