A close up of a logo

Description automatically generated

**WEEK 6: COLLABORATIVE GROUP PROJECT – BANK MARKETING**

RONAK SOMANI

SHREENIDHI PRADEEP TARTE

BINDU MOUNIKA VUNDRAKONDA

RUI WANG

**NORTHEASTERN UNIVERSITY**

**MPS ANALYTICS – SPRING 2020**

**SUBJECT: ALY 6020 81998 PREDICTIVE ANALYTICS**

**INSTRUCTOR NAME: PROF. NA YU**

**SUBMISSION DATE: May 11, 2020**

**INTRODUCTION**

Marketing of bank items is the total capacity retained at giving the office to fulfill client's financial needs more than the competition keeping in view the authoritative targets. Banking is a customized service-oriented industry and subsequently ought to give services which fulfill the clients' needs. The marketing strategy incorporates grouping, reacting and fulfilling the clients' needs and needs effectively, expertly, and gainfully.

The fundamental job of the bank isn't just to achieve and win an ever-increasing number of clients but also to provide them with the best possible services. Marketing as related with banking is to disclose an appropriate guarantee to a client through an assortment of items and administrations and furthermore to affirm usable dissemination through fulfillment.

**Objective:** There has been a revenue decline for the Portuguese bank, and they would like to know what actions to take. After investigation, we found out that the root cause is that their clients are not depositing as frequently as before. Knowing that term deposits allow banks to hold onto a deposit for a specific amount of time, so banks can invest in higher gain financial products to make a profit. In addition, banks also hold better chance to persuade term deposit clients into buying other products such as funds or insurance to further increase their revenues. As a result, by using machine learning algorithm the Portuguese bank would like to identify existing clients that have higher chance to subscribe for a term deposit and focus marketing effort on such clients. Term deposit is defined as fixed-term investment that includes the deposit of money into an account at a financial institution.

**Data Source and Dataset Information**

The dataset is related with the marketing campaigns of a Portuguese banking institution. This data was picked up from UCI Machine Learning Repository.

Link: <https://archive.ics.uci.edu/ml/datasets/bank+marketing>

The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or would not be ('no') subscribed.

**Data Observations**

The following categories of information are included in the data set:

* age (numeric)
* job: type of job (categorical: “admin”, “unknown”, “unemployed”, “management”, “housemaid”, “entrepreneur”, “student”, “blue-collar”, “self-employed”, “retired”, “technician”, “services”)
* marital: marital status (categorical: “married”, “divorced”, “single”; note: “divorced” means divorced or widowed)
* education (categorical: “unknown”, “secondary”, “primary”, “tertiary”)
* default: has credit in default? (binary: “yes”, “no”)
* balance: has balance loan, in euros (numeric)
* housing: has housing loan? (binary: “yes”, “no”)
* loan: has personal loan? (binary: “yes”, “no”)
* contact: contact communication type (categorical: “unknown”, “telephone”, “cellular”)
* day: last contact day of the month (numeric)
* month: last contact month of year (categorical: “jan”, “feb”, “mar”, …, “nov”, “dec”)
* duration: last contact duration, in seconds (numeric)

# other attributes:

* campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
* pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
* previous: number of contacts performed before this campaign and for this client (numeric)
* poutcome: outcome of the previous marketing campaign (categorical: “unknown”, “other”, “failure”, “success”)

**Output variable:**

* y - has the client subscribed a term deposit? (binary: “yes”, “no”)

**ANALYSIS**

In order to classify whether the bank deposit term would be subscribed (“Yes”) or (“No”) by a client we have used Decision tree and Random Forest machine learning algorithms. Let’s have a look as to how Decision tree machine learning algorithm is used to classify whether the deposit term would be subscribed or not.

**Step 1 – Importing the necessary packages**

For implementing Decision Tree and Random Forest in r, we need to import “C50”, “caret”, “rpart.plot” and “randomForest” packages. As we mentioned above, caret helps to perform various tasks for our machine learning work. The “rpart.plot” package will help to get a visual plot of the decision tree. “lattice” package is used for creating trellis graph, which are graphs that display a variable or the relationship between variables. We have also inserted other packages like “gmodels”, “corrplot”, “tidyverse”, “ggpubr” and “ggplot2” package for data visualization purpose:

A screenshot of a cell phone

Description automatically generated

**Step 2 – Collecting Data**

For importing data into an R data frame, we can use read.csv() method with parameters as a file name and whether our dataset consists of the 1st row with a header or not. If a header row exists then, the header should be set TRUE else header should set to FALSE. But before that we need to set the working directory in RStudio where the downloaded dataset is stored. The head function is then used to display the first 6 rows of the dataset.



A picture containing bird

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Step 3 – Exploring and preparing the data**

For checking the structure of data frame we can call the function str() over bank which determines the structure of the data that is the data types of every attribute.

A screenshot of text

Description automatically generated

From the above output we determine that the dataset consist of 45211 observations and 17 attributes and their corresponding data types. The variable “y”, which is a factor, represents if the client will subscribe a term deposit or not by “yes” or “no” respectively. The “y” column is used for prediction of the dataset for understanding the bank term deposit classification. The rest of the features are combination of factor and integer data types.

We can also use summary(bank) function for more detailed insights. For instance, for all the integer variables summary function can give us the descriptive statistics like min., max., mean, median and quantiles. Similarly, for factor variables it shows us the number of counts for each attribute inside a variable. Also, by using summary function we can check if all the required variables are of same scale or if they need to be normalized to apply classification models.

A close up of a receipt

Description automatically generated

**Checking if there are any NA values**

We will be using anyNA(bank) function to check if there are any NA values in the dataset. The function will return True or False for the presence of NA values in the dataset.

A picture containing knife

Description automatically generated

We thus conclude there are no NA values in the dataset since the function returns false.

**Step 4 – Exploratory Data Analysis**

After the data is explored and preprocessed, using data visualizations we will gain insights about the dataset. We will be plotting correlation matrix, bar plot and histogram in order to determine the patterns in our data so that we can make meaningful inference. The correlation matrix has been plotted on all the numerical values like age, balance, day, duration, campaign, pdays and previous. While the bar plot considers variables like job and balance. Histogram uses the variable duration.

**Correlation Matrix:**

Correlation matrix determines the degree of association between the numerical variables present in the data. In our case it will be age, balance, day, duration, campaign, pdays and previous. We have first extracted the numerical variables by using slicing and stored it in variable bank1 and then implemented “cor()” function to generate correlation matrix called bank2. We have then used corrplot() function to visualize the correlation matrix.

A screenshot of a cell phone

Description automatically generated

A close up of a logo

Description automatically generated

From the above correlation matrix, we infer that the variables are very weakly correlated with each other indicating that there is no approximately correlation between them which means there is very less multicollinearity, which is a good sign.

**Distribution of duration of loan:**

Histogram determines the distribution of a continuous variable; we can determine whether our variable follows a normal distribution or is the distribution positively or negatively skewed. In our dataset we will be analyzing the distribution of the continuous variable duration using ggplot() function.

A picture containing knife

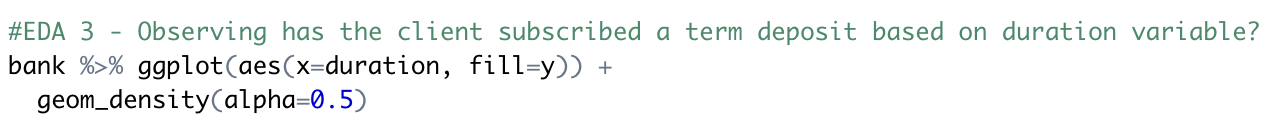
Description automatically generated

A close up of a device

Description automatically generated

From the above output we determine that most of the duration values are concentrated in the 0-500 seconds while there are very less values for more than 1000 seconds duration.

**Observing if the client subscribed a term deposit based on duration variable?**

****

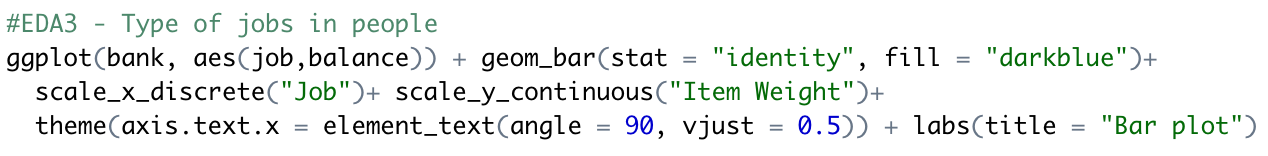
**A close up of a logo

Description automatically generated**

Variable duration is a leakage and cannot be sensibly used in a predictive model as it is not known before a call is made. It is highly predictive of subscription, i.e. people who spent more time on the call were more likely to subscribe. Initial models which included this variable show that it had the highest importance. Moro et al. (2014) also showed that this variable is the most important.

**Types of jobs in people:**

Bar-plot determines the frequency of occurrence of a categorical variable in the dataset. In our dataset we have made a bar-plot of the outlet type variable versus the item weight.



A picture containing drawing

Description automatically generated

From the above output we determine that the job “management” had the most balance followed by “blue collar” and “technician”” making them most likely to be subscribed for the bank deposit term.

**Marital status in people**

This is to determine the kind of marital clients the bank has, and does it affect the loan default.

A picture containing bird

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Education status in people**

What kind of education clients the bank has, if you cross education with default, loan or housing, there is no relation.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Distribution of loan default**

This is to check how many people actually default on their loan. From the graph, we can see that very few people default. This is a good indication for the bank as there will not be a problem in processing payments from the bank’s end.



A screenshot of a cell phone

Description automatically generated

**Distribution of output variable – y**

This plot shows the distribution of output variable. From the plot we can see that most of the data falls under the category “no”. Which means that a person doesn’t subscribe to a term deposit with the bank.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Age Distribution**

**A screenshot of a cell phone

Description automatically generated**

**A close up of a device

Description automatically generated**

The bulk of clients are between the ages of 33 (1st Quartile) and 48 (3rd Quartile) with mean lying on 41 visualized on the histogram with red vertical line. Moreover, we can see outliers above the age of 65.

**Observing the client subscribed term deposit or not based on age.**

****

**A picture containing text

Description automatically generated**

Most clients that subscribe are between age 25 to 45. Mean age for all clients is above 40 years of age. Also, we can observe the sharp drop of clients above age 60.

**Observing whether clients having housing loans subscribe for term deposit or not.**

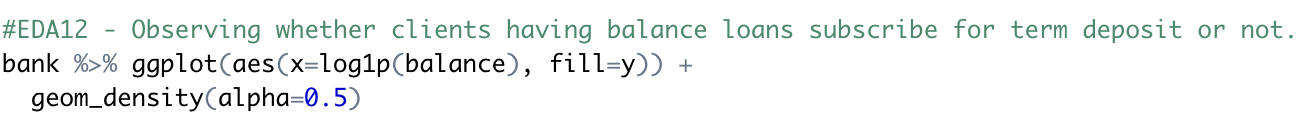


A screenshot of a cell phone

Description automatically generated

From the above bar-plot we can state that, clients who don’t have housing loans are more inclined towards subscribing term-deposit as compared to clients who have housing loans.

**Observing whether clients having balance loans subscribe for term deposit or not.**



A close up of a logo

Description automatically generated

This plot is designed to shows the relationship between clients subscribing term deposit and clients having balance loans. Since the distribution range of balance is too large and extremely right skewed, we perform log1p () transformation on balance. Clients with high balance are usually unwilling to subscribe the term deposit. However, customers with low balance loans are usually willing to subscribe the bank term deposit.

**Step 5 – Training models on the data**

**1st Model: DECISION TREE:**

Decision trees are versatile Machine Learning algorithm that can perform both classification and regression tasks. They are very powerful algorithms, capable of fitting complex datasets. Besides, decision trees are fundamental components of random forests, which are among the most potent Machine Learning algorithms available today.

Let’s see its implementation in using statistical software in R:

**Data slicing**

Data slicing is a step to split data into train and test set. Training data set can be used specifically for our model building. Test dataset should not be mixed up while building model. Even during standardization, we should not standardize our test set. To make our answers replicable, we need to set a seed value. During partitioning of data, it splits randomly but if our readers will pass the same value in the set.seed() method. Then we both will get identical results. The caret package provides a method createDataPartition() for partitioning our data into train and test set. We are passing 3 parameters. The “y” parameter takes the value of variable according to which data needs to be partitioned. In our case, target variable is at y, so we are passing bank$y.

The “p” parameter holds a decimal value in the range of 0-1. It’s to show that percentage of the split. We are using p=0.8. It means that data split should be done in 80:20 ratio. The “list” parameter is for whether to return a list or matrix. We are passing FALSE for not returning a list.

A screenshot of a cell phone

Description automatically generated

A picture containing knife

Description automatically generated**Training the Decision Tree classifier with criterion as information gain**

We will first use trainControl() method. It controls the computational nuances of the train() method. We are setting 3 parameters of trainControl() method. The “method” parameter holds the details about resampling method. We can set “method” with many values like “boot”, “boot632”, “cv”, “repeatedcv”, “LOOCV”, “LGOCV” etc. For this tutorial, let’s try to use repeatedcv i.e., repeated cross-validation.

The “number” parameter holds the number of resampling iterations. The “repeats” parameter contains the complete sets of folds to compute for our repeated cross-validation. We are using setting number =10 and repeats =3. This trainControl() methods returns a list. We are going to pass this on our train() method.

Before training our Decision Tree classifier, set.seed().

We are passing our target variable y. The “y~.” denotes a formula for using all attributes in our classifier and y as the target variable. The “trControl” parameter should be passed with results from our trainControl() method.

A screenshot of a cell phone

Description automatically generated

A screenshot of text

Description automatically generated

**Plot Decision Tree**

We can visualize our decision tree by using prp() method.

A picture containing knife

Description automatically generated

A picture containing text, map

Description automatically generated

The decision tree visualization shown above indicates its structure. It shows the attribute’s selection order for criterion as information gain.

**Evaluating Model Performance**

**Predicting the model on test data**

Now, our model is trained with cp = 0.001654064. We are ready to predict classes for our test set. We can use predict() method.

A picture containing knife

Description automatically generated

A screenshot of text

Description automatically generated

The above results show that the classifier with the criterion as information gain is giving 89.91% of accuracy for the test set. The term-deposit variable i.e. the “y” variable is compared with the prediction variable to check the appropriate classification. The data is classified into true positive, true negative, false positive and false negative categories. Here using confusion matrix, we can compare the predicted values with the actual values. It shows that 7748 clients who did not subscribed term deposit were correctly classified showing true positive results. Similarly, 381 clients who did subscribe were also correctly classified showing true negative. But 236 clients who actually did not subscribed but were misclassified as subscribed and 676 who did subscribe but were misclassified as did not are representing false positive and false negative respectively. Moreover, we can also check other parameters like sensitivity, specificity, Positive and negative prediction values, kappa values etc. Using other parameters, we can check whether the model can be improved.

**Training the Decision Tree classifier with criterion as Gini index**

Let’s try to program a decision tree classifier using splitting criterion as Gini index. It is showing us the accuracy metrics for different values of cp. Here, cp is complexity parameter for our dtree.

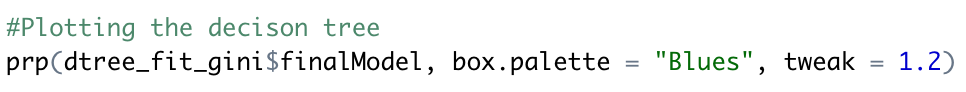
A screenshot of a cell phone

Description automatically generated

A screenshot of text

Description automatically generated

**Plotting the decision tree**

****

**A picture containing text, map

Description automatically generated**

The decision tree visualization shown above indicates its structure. It shows the attribute’s selection order for criterion as Gini index.

**Evaluating Model Performance**

**Predicting the model on test data**

Now, our model is trained with cp = 0.001654064. We are ready to predict classes for our test set. Now, it’s time to predict target variable for the whole test set.

A picture containing knife

Description automatically generated

A screenshot of a cell phone screen with text

Description automatically generated

The above results show that the classifier with the criterion as Gini index is giving 90.03% of accuracy for the test set. In this case, our classifier with criterion Gini index is giving slightly better results. The term-deposit variable i.e. the “y” variable is compared with the prediction variable to check the appropriate classification. The data is classified into true positive, true negative, false positive and false negative categories. Here using confusion matrix, we can compare the predicted values with the actual values. It shows that 7728 clients who did not subscribed term deposit were correctly classified showing true positive results. Similarly, 412 clients who did subscribe were also correctly classified showing true negative. But 256 clients who actually did not subscribed but were misclassified as subscribed and 645 who did subscribe but were misclassified as did not are representing false positive and false negative respectively. Moreover, we can also check other parameters like sensitivity, specificity, Positive and negative prediction values, kappa values etc.

**2nd Model: RANDOM FOREST:**

Random forests are an ensemble learning method that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees habit of overfitting to their training set. Random forests differ in only one way from tree bagging: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. Tree bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples. This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. In R the main implementation of random forest is found in the randomForest library.

The Random Forest method takes the 17 features from the dataset in which 16 predictors and 1 dependent variable is used for making the Breiman’s random forest classification towards the final decision based on dependent variable. The syntax for the random forests suggests the classification of the dataset features using multiple variables represented by ‘~’ variable. The random forest provides some basic options to select the number of mtry and tree control variables. By default, the ntree is selected 500 and mtry is selected as 4.

A picture containing knife

Description automatically generated

The above statements execute the random forest on the training dataset, the classifier creates an object which is used for making the prediction labels. This stores the basic information about the tree suggesting the number of samples and the number of predictor attributes. The random forest is built based on the different attributes from the dataset. The output suggests the proportion of the data used from the factors for building the random forest classifier output. The OOB is out of bag error suggesting the data that is not be used in training and being left for testing purposes and the data will be used when multiple test models are compiled.

This is the summary of the random forest which gives the output with confusion matrix and description about the classification. This shows the information of the different features with respect to their allocation and misclassification. The confusion matrix also shows the error rate for every factor and its misclassification percentage for the random forest model.

A screenshot of a cell phone

Description automatically generated

The confusion matrix shows the performance data for Random. It is indicating the class error rate of minimum 3.67% till maximum of 50.54%. 9.15% records were misclassified from 36170 training data. The confusion matrix shows the breakdown of false positive and false negative i.e. 2139 clients who didn’t subscribed term deposit where misclassified as they did subscribe and 1171 who did subscribe were misclassified as didn’t subscribed respectively.

**Evaluating Variable Importance**

Using importance(rf) function we can check the importance of each variable in our dataset for random forest model and rule out those variables which are less significant

**A screenshot of text

Description automatically generated**

From the above output, we can see that variables like duration, age, balance, month, day, job and poutcome are highly significant as compared to other variables. The same can be depicted in graph format as per the level of significance below using varImpPlot(rf) function in R.

**A screenshot of a social media post

Description automatically generated**

Higher the value of mean decrease accuracy or mean decrease Gini score, higher the importance of the variable in the model. In the plot shown above, Duration is most important variable.

**Mean Decrease Accuracy** - How much the model accuracy decreases if we drop that variable.

**Mean Decrease Gini** - Measure of variable importance based on the Gini impurity index used for the calculation of splits in trees.

This step evaluates the performance of the model based on the labels provided. This includes the verification of the results in order to verify whether there are any misclassifications to the model. The prediction vector is matched with the labels to identify the performance. By using the function “predict”, we can get a vector of predicted values:

A picture containing table

Description automatically generated

A screenshot of text

Description automatically generated

A picture containing knife

Description automatically generated

The above results show that the random forest classifier is giving 90.52% of accuracy for the test set. The term-deposit variable i.e. the “y” variable is compared with the prediction variable to check the appropriate classification. The data is classified into true positive, true negative, false positive and false negative categories. Here using confusion matrix, we can compare the predicted values with the actual values. It shows that 7694 clients who did not subscribed term deposit were correctly classified showing true positive results. Similarly, 490 clients who did subscribe were also correctly classified showing true negative. But 290 clients who actually did not subscribed but were misclassified as subscribed and 567 who did subscribe but were misclassified as did not are representing false positive and false negative respectively. Moreover, we can also check other parameters like sensitivity, specificity, Positive and negative prediction values, kappa values etc.

**ROC and AUC for the model**

To better inspect the model, we could use Receiver Operating Characteristics Curve which traces the percentage of true positives accurately. There are useful statistics that can be calculated from this curve, like the Area Under the Curve (AUC) and the Youden Index. These tell you how well the model predicts and the optimal cut point for any given model (under specific circumstances). The cutoff is analyzed with the values to identify best threshold.

**A picture containing knife

Description automatically generated**

**A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated**

**A close up of a logo

Description automatically generated**

From the plot above, the area under the curve is 71.36%.

**3rd Model: Logistic Regression**

Logistic Regression is used when the data is linearly separable or classifiable and the outcome is Binary or Dichotomous. The model tries to fit a two-class problem like ham or spam, or malign or benign tumor (i.e. response variable can only take 0 or 1). Logistic regression models the probability of the membership to a particular group or category without directly modeling the values of the dependent variable.

To build a model using the logistic regression, first we need to set seed using set.seed() function to make our outputs replicable. We will first use trainControl() method. It controls the computational nuances of the train() method. We are setting 4 parameters of trainControl() method. The “method” parameter holds the details about resampling method. We can set “method” with many values like “boot”, “boot632”, “cv”, “repeatedcv”, “LOOCV”, “LGOCV” etc. For this tutorial, let’s try to use cv i.e., cross-validation.

The “number” parameter holds the number of resampling iterations. We are using setting number =10. The “classProbs” parameter is a logical parameter. If we set it to “TRUE” then it computes the class probabilities for classification models (along with predicted values) in each resample. The “summaryFunction” parameter is a function to compute performance metrics across resamples. We are passing multiClassSummary which returns more metrics than binary classification.

A screenshot of a cell phone

Description automatically generated

**Build Model by fitting Logistic Regression**

The train() function is used to build our GLM model on training dataset. We are passing our target variable y. The “y~.” denotes a formula for using all attributes in our classifier and y as the target variable. In method parameter we will pass “glm” which specifies which classification or regression model to use. The procedure with family="binomial" will build the logistic regression model on the given formula. The “trControl” parameter should be passed with results from our trainControl() method which is a list of values that tells how the model should act.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

From the above model results we can see that there are 16 predictors and 2 classes indicating results as yes or no. There is 10-fold classification as we defined in the code above. Our accuracy in the model is 90.15% and kappa is 40.43%. Kappa is the measure of inter-rater agreement. Then we have percentage of logLoss, AUC, predicted AUC, F1, Sensitivity, specificity, positive and negative predictive values, precisions, recall, detection rate and balanced accuracy.

**Evaluate Model Performance**

Predict the Logistic Regression model on testing data and evaluate by finding the accuracy and calculating/plotting ROC curve.

For evaluating the logistic regression prediction model we will use predict.train() function. In that we will pass our glm model which we created above and run it on our testing dataset. This function has a parameter type which takes 2 inputs either “raw” or “prob”. For predict.train, a vector of predictions if type = "raw" or a data frame of class probabilities for type = "probs". In the latter case, there are columns for each class.

A screenshot of a cell phone

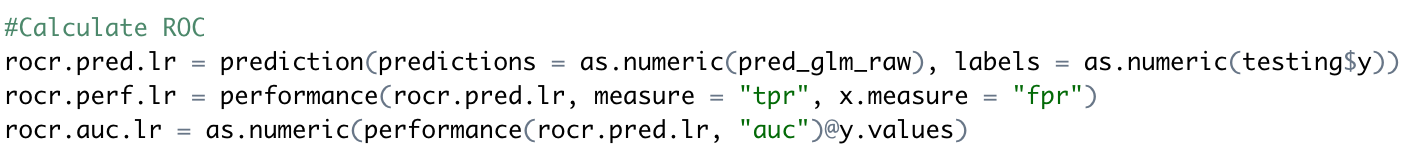
Description automatically generated

The confusion matrix results below show that the logistic regression model is giving 90.03% of accuracy for the test set. The term-deposit variable i.e. the “y” variable is compared with the prediction variable to check the appropriate classification. The data is classified into true positive, true negative, false positive and false negative categories. Here using confusion matrix, we can compare the predicted values with the actual values. It shows that 7795 clients who did not subscribed term deposit were correctly classified showing true positive results. Similarly, 345 clients who did subscribe were also correctly classified showing true negative. But 189 clients who actually did not subscribed but were misclassified as subscribed and 712 who did subscribe but were misclassified as did not are representing false positive and false negative respectively. Moreover, we can also check other parameters like sensitivity, specificity, Positive and negative prediction values, kappa values etc.

A screenshot of text

Description automatically generated

To better inspect the model, we could use Receiver Operating Characteristics Curve which traces the percentage of true positives accurately. There are useful statistics that can be calculated from this curve, like the Area Under the Curve (AUC) and the Youden Index. These tell you how well the model predicts and the optimal cut point for any given model (under specific circumstances). The cutoff is analyzed with the values to identify best threshold.



A picture containing knife

Description automatically generated

The figure below describes the graphical presentation of the ROC curve and the area under the curve below states the true positive ratio which is 65.13%.

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generated

**EXPLAIN GLM MODEL**

For creating an explainer, we will use DALEX package in R. DALEX package xrays any model and helps to explore and explain its behavior. It contains various methods that help to understand the link between input variables and model output. The DALEX architecture can be split into three primary operations:

* Any supervised regression or binary classification model with defined input (X) and output (Y) where the output can be customized to a defined format can be used.
* The machine learning model is converted to an “explainer” object via DALEX::explain(), which is just a list that contains the training data and meta data on the machine learning model.
* The explainer object can be passed onto multiple functions that explain different components of the given model.

Although DALEX does have native support for some ML model objects. To make DALEX compatible with these objects, we need three things:

* x: Our feature set needs to be in its original form.
* yTest: Our response variable needs to be a numeric vector. For regression problems this is simple, as it will already be in this format. For binary classification this requires you to convert the responses to 0/1.
* P\_fun: a custom predict function that returns a vector of numeric values. For binary classification problems, this means extracting the probability of the response.

Once we have these three components, we can create explainer objects for GLM model.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

From the above results we can see that a new explainer is initiated where the model label is glm, the data consist of 9041 rows and 17 columns in the test data. The model is using caret package to build a model and predicted values are max 0.9998 & mean is 0.112. Moreover, residuals ranges from -0.9991 to max 0.9980 where mean is 0.0044.

**Residual diagnostics**

Now let’s see the reverse cumulative distribution of the residuals. Assessing residuals of predicted versus actuals can allow you to identify where models deviate in their predictive accuracy. We can use DALEX::model\_performance to compute the predictions and residuals. Printing the output returns residual quantiles and plotting the output allows for easy comparison of absolute residual values across models.

A screenshot of a cell phone

Description automatically generated

From the above results, we can see that our accuracy is 90.03% and residual percentages are increasing along with increase in the percentages. Let’s see the graphical presentation of the cumulative distribution of residual as follows.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Results**

The algorithms used on the data set consist of decision trees, random forest and logistic regression which show an average accuracy of 90%. We can notice that in all three models, sensitivity is lower compared to specificity. A possible explanation for this may be because of the distribution of the output variable (most of the observations fall under “no”).

**Discussion/Conclusions**

In this project we have implemented machine learning techniques to a Portuguese bank marketing data and tried understanding the factors that influence subscription to the term deposit. Multiple classification algorithms have been implemented. Cross validation was used for parameter selection in decision trees.

**REFERENCES**

Moro, S., Cortez, P., & Rita, P. (2014, June). Bank Marketing Data Set. Retrieved from

[https://archive.ics.uci.edu/ml/datasets/bank marketing#](https://archive.ics.uci.edu/ml/datasets/bank+marketing)