SP.TP.IN DATA-DRIVEN DECISION MAKING

Şaban Dalaman

Summer, 2019

DIRECT MARKETING CAMPAIGNS OF A PORTUGUESE BANKING INSTITUTION

Report of Final Project

Group Meh

Burakhan Sel

Mehmet Burak Çakır

Mustafa Kemal Şaşmaz

Sergen Tuğ Toraman

Contents

Abstract

We propose a Machine Learning approach to predict the success of telemarketing calls to bank long-term deposit selling. A Portuguese retail bank was addressed including the effects of the recent financial crisis. We analyzed a series of 16 features related to bank customer, product and social-economic characteristics. Via methods that are based on Random Forest and Logistic Regression algorithms were utilized to select/eliminate features to be examined in the modeling stage. We compared four classification models: logistic regression, random forest(RF), decision trees (DT), Gradient Boost (GB). Using two measurements, the area of the operating characteristic curve of the receiver (AUC) and the accuracy of the four models were tested. Our model Random Forest model provided the best results with 92% AUC and 83% accuracy.

Introduction

Marketing selling campaigns constitute a typical strategy to enhance business and asset management for the retail banks. Companies use direct marketing when targeting segments of customers by contacting them to meet a specific goal such as deposits. Centralizing customer remote interactions in a contact center eases operational management of campaigns. Such centers allow communicating with customers through various channels, telephone (fixed-line or mobile) being one of the most widely used. Technology enables rethinking marketing by focusing on maximizing customer lifetime value through the evaluation of available information and customer metrics, thus allowing us to build longer and tighter relations in alignment with business demand.

Decision support systems use information technology to support managerial decision making. There are several classification models, such as Random Forest (RF), Logistic Regression (LR), decision trees (DTs) and support vector machines (SVMs). In this study we have gathered our data from archive.ics.uci.edu. We analyze a recent and large dataset (45.211 records) from a Portuguese bank. The paper is organized as follows presents the data structure and summary, then our Machine Learning Approach, Models and finally the conclusions are drawn.

Methods Review

In this study we will implement 4 different Machine Learning Classification methods:

1. Logistic Regression
2. Random Forest (RF)
3. Decision Tree (DT)
4. Gradient Boosting (GB)

Before forecasting whether the telemarketing calls to bank long-term deposit selling ends with success or not, we will choose the best combination of features, models and tuning the parameters.

For the selection of significantly important features we have used two different methods:

1. Random Forest (RF)
2. Logistic Regression

Background

The recent 2007 financial crisis that has spread around the world has caused a considerable slowdown in the global economic landscape, affected financial markets and decreased economic growth; above all, it creates a deep downturn for developing countries particularly at Eastern European countries and their financial institutions. After the financial crisis, management of customer lifetime value, long-term deposit and asset management models have significantly improved their importance.

In our example we have Portuguese retail bank, with telemarketing calls to bank long-term deposit selling. The success of long-term deposit selling may depend on several variables including Personal information such as Age, Type of job, Education, financial information such as current credit in default, housing loan, personal loan and also campaign information: number of contacts performed during this campaign and for this client, number of days that passed by after the client was last contacted from a previous campaign.

So, our aim is to develop feature selection algorithm, model the relationship between these features and success of the telemarketing calls.

Data Structure and Summary

The dataset has been constructed with the responses of customers that a Portuguese banking institution reached for their marketing campaign. It is beneficial to know that the bank reaches their customers for multiple times via phone calls and tries to persuade them to subscribe a term deposit.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
age          45211 non-null int64
job          45211 non-null object
marital      45211 non-null object
education    45211 non-null object
default      45211 non-null object
balance      45211 non-null int64
housing      45211 non-null object
loan         45211 non-null object
contact      45211 non-null object
day          45211 non-null int64
month        45211 non-null object
duration     45211 non-null int64
campaign     45211 non-null int64
pdays        45211 non-null int64
previous     45211 non-null int64
poutcome     45211 non-null object
y            45211 non-null object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

# of Unique Values	
age	77
job	12
marital	3
education	4
default	2
balance	7168
housing	2
loan	2
contact	3
day	31
month	12
duration	1573
campaign	48
pdays	559
previous	41
poutcome	4
y	2
dtype: int64	

Figure 1 The summary of dataset.

There are 45,211 instances, 16 features and a response column in the dataset. By observing the following outcomes of our exploration script, we may conclude that there are no NA values. However, we know that there are "unknown" values in the dataset, and these values are -in fact- NAs of the data. They will be handled after splitting the data. Also, "month" is originally categorical. We will create a new feature as "month_num" to have a numeric feature of months.

Before transforming "unknown" values to NAs, we will analyze its proportions shown in figures above. The highest percentage of "unknown" values is in respect of "poutcome". Initially, transformation will be done, and no feature will be crossed out even if it is more than 50 percent. In case there exists a condition that is unexpected after handling all NAs (i.e. "unknown" values), corresponding feature will be, then, crossed out.

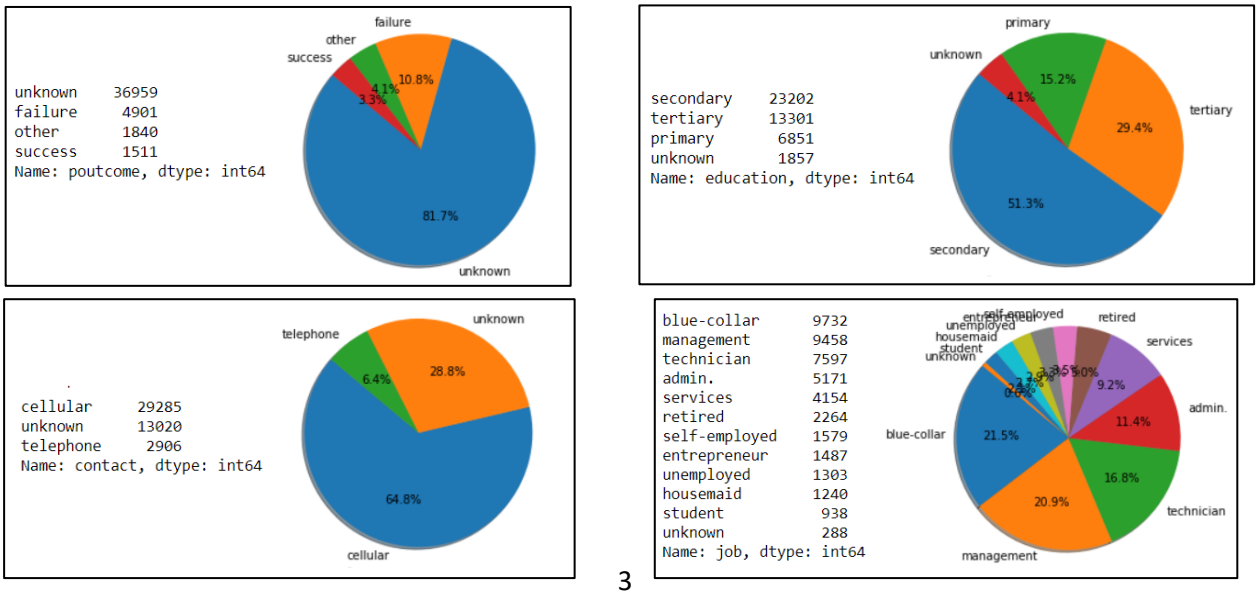

Figure 2 Proportions of "unknown" values.

To analyze whether there are extreme cases among any feature, we will get the summary of numeric columns as follows. There negative values in “balance” and “pdays”. Former will be scaled while others being normalized to minimize the effect of negativity. It is required to mention that scaling will be done. For the latter one, value of -1 implies that related customer is not called at all. That is why -1 seems wrong; therefore, we have changed this value to 10,000.

	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

Figure 3 The summary of numeric features of the data.

As we try to solve a binary classification problem, while preparing the dataset for any type of machine learning model, response column is transformed into binary column with values of 0 for “no” and 1 for “yes”. Data is checked in terms of balance of responses. Following bar chart illustrates that the dataset is imbalanced. We will balance the data after splitting the data.

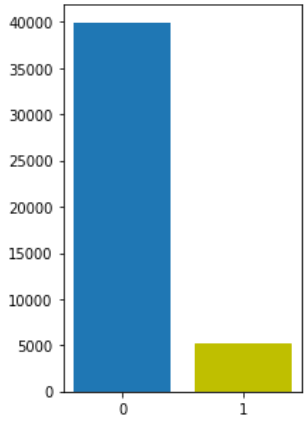

Figure 4 Imbalance of the responses.

Data Split and Restructuring

Data is divided into two groups named as "train" and "test". "train" dataset is 70% of the main data while remaining 30% is the "test" dataset. From now on,

- NA values will be handled by replacing with the most repeated values.
- Datasets will be balanced distinctively via oversampling algorithm.

# of handled NAs

```
for train dataset
job              194
education       1273
contact         9043
poutcome       25901
dtype: int64

for test dataset
job               94
education        584
contact         3977
poutcome       11058
dtype: int64
```

Balance of train dataset	Balance of test dataset
before 0.0 27909 1.0 3738 Name: y, dtype: int64 after 1.0 27909 0.0 27909 Name: y, dtype: int64	before 0 12013 1 1551 Name: y, dtype: int64 after 1 12013 0 12013 Name: y, dtype: int64

Figure 5 The brief information on the replacement of NAs. Balancing datasets of train and test.

Datasets will be scaled but categorical columns must be transformed into Boolean dummies since methods require such datasets. After this transformation, number of columns increased to 49, and there is no categorical feature with the type of string. The next step is done that all columns are scaled via min-max scaler.

Features and the response are separated from each other to employ machine learning models and algorithms.

Statistical Analysis and Data Exploration

The boxplot (Figure 6) illustrates that the duration on the call is related to the response of the client. As the conversation time gets longer, the probability of having a positive answer increases.

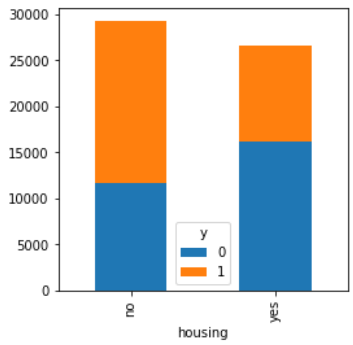

Figure 7

Figure 7 shows that a customer with a house is more likely to subscribe a term deposit. Furthermore, it is valid for the other way around. Having no house is a trigger to respond the subscription offer of the bank.

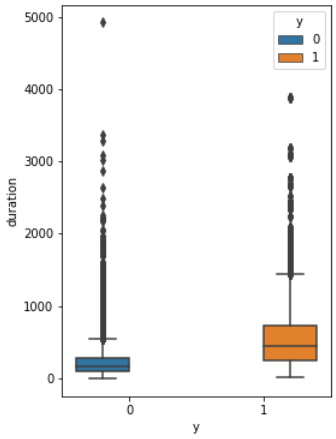

Figure 6

Following graphs in Figure 8 are generated by considering the normalized values of frequencies. It may be claimed that education and some of the jobs have impacts on the subscription decision. For example, students and retired people are more likely to say “yes” to the marketing campaign.

Age is an important feature to gain insight about the tendency of the customer. After a certain period of lifetime, positive response ratios scale up dramatically.

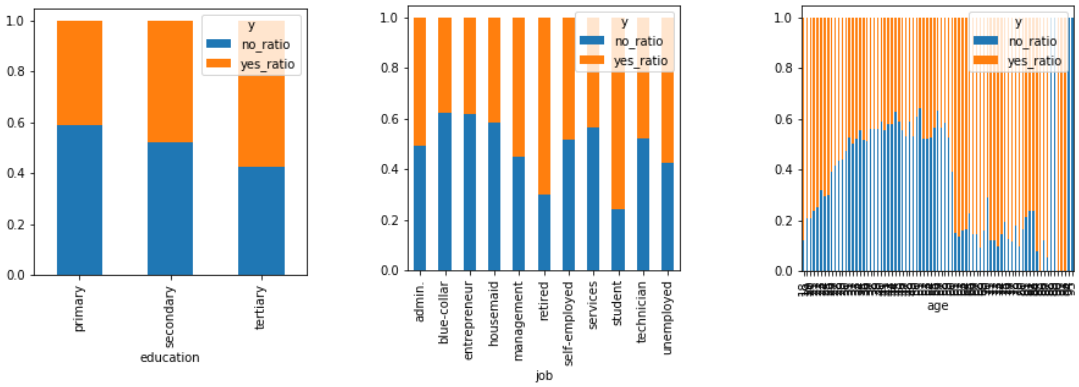

Figure 8

The boxplot below (Figure 9) compares the outcomes of previous marketing campaign with durations of calls. It may be concluded that success of previous campaign leads customers to be persuaded quicker to subscribe than a failed previous campaign. This situation satisfies our expectation. Hence, we will keep “poutcome” as one of the features even though it had 81% of NA values.

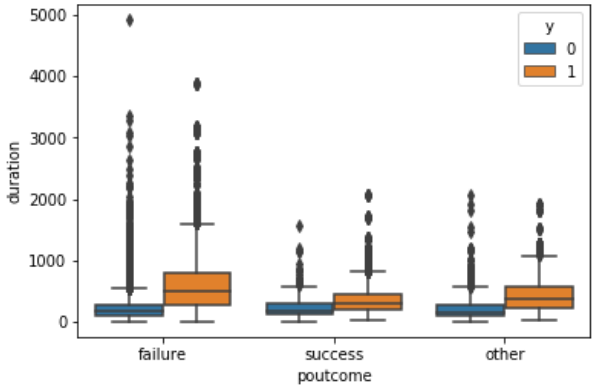

Figure 9

As client's balance increases, their tendency to keep the talk shorter and refuse the subscription increases. It may be observed in the following scatter plot (Figure 10).

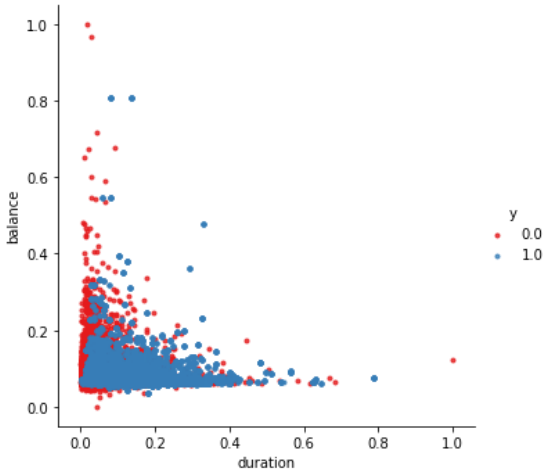

Figure 10

Following heat map, Figure 11, provides the overall understanding of the dataset. It is a good way to observe expected situations versus unexpected situations. In the next step, while eliminating features, this map can be accepted as an intuitive tool if needed.

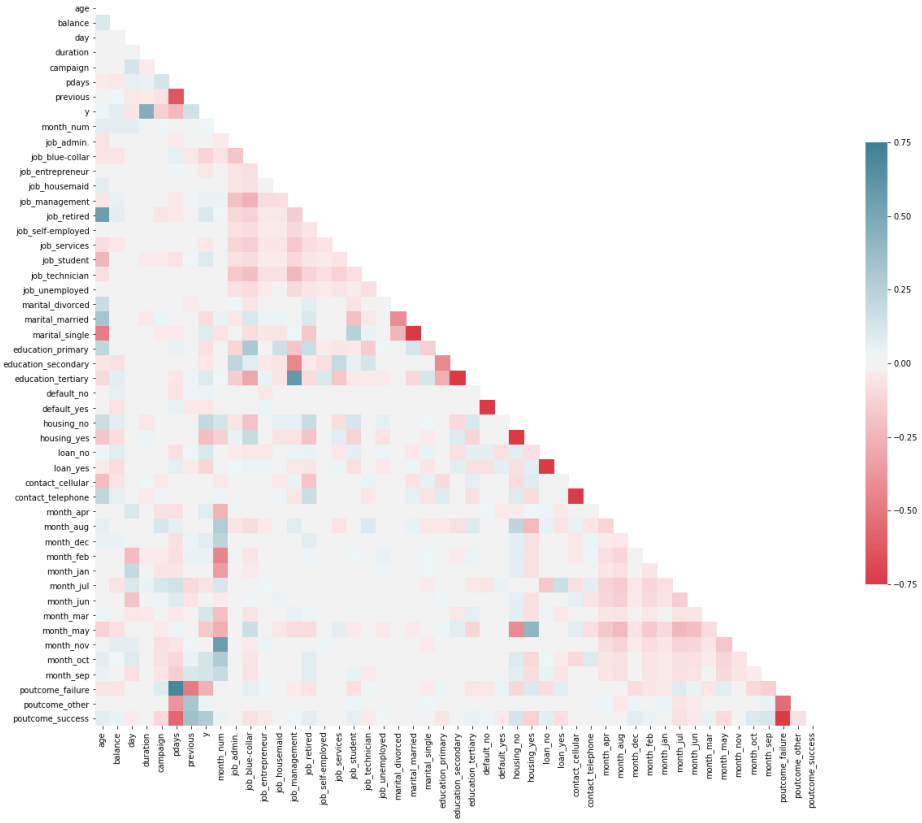

Figure 11

Feature Selection

We tried two methods to select significantly important features. First of them is a feature selection via random forest based method. Importance levels are as follows. We will generate feature lists accordingly. Two lists will be created: (List 1) one with the importance percentages are greater than 0.5% and (List 2) other one with the importance percentages are greater than 1%.

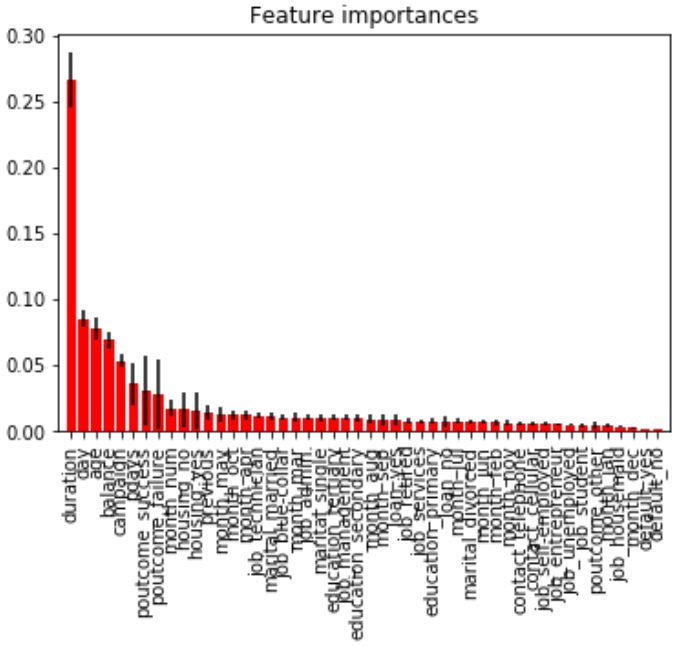

Figure 12 Feature importance via random forest based method.

Second one is a feature selection algorithm via logistic regression. It concludes the feature list with respect to its p-values. Result of this method is as follows.

```
Index(['poutcome_success', 'housing_no', 'duration', 'pdays', 'month_mar',
       'month_oct', 'month_sep', 'month_apr', 'loan_no', 'campaign',
       'marital_single', 'job_blue-collar', 'month_feb', 'month_dec',
       'job_student', 'education_tertiary', 'job_retired', 'month_jun',
       'job_admin.', 'month_num', 'education_primary', 'job_entrepreneur',
       'balance', 'loan_yes', 'day', 'previous', 'job_housemaid', 'default_no',
       'poutcome_other'],
      dtype='object')
```

Finally, one list will involve the common features of List 2 and List 3. All in all, we have got 4 lists.

1. 41 features via random forest based method (List 1)
2. 24 features via random forest based method (List 2)
3. 31 features via logistic regression (List 3)
4. Common features of List 2 and List 3 (List 4)

Methods

To accurately predict the output, there are many things to decide both in the data preparation and the model fitting stages. The decisions and implementations in the data preparation stage have been explained so far. However, there are still some issues to consider finding the best method to predict whether the customer will purchase the product or not.

The first issue is to choose the best combination of the features since a reduction in the number of features may increase the performance of the models. Therefore, as explained earlier, we have 4 different datasets to be given as inputs for each prediction model.

The second issue is which models will be tried and how to tune their parameters. The decisions about these concerns are made individually for each model by considering the tradeoff between Area Under Curve of Reciever Operating Characteristic (AUC-ROC) and computational complexities of the models. The reason for choosing AUC-ROC as a performance measure is that it provides us a measure independent from which threshold is used to convert probabilities to binary predictions.

We started building models with Logistic Regression. For the Logistic Regression, there is a hyperparameter called link function which can be “probit” or “logit”. We implemented both link functions to the 3 of the 4 datasets by excluding the one with the 41 features because using this number of features in the logistic regression resulted in non-converging solution. In total, 6 (i.e. 3*2) different results are obtained via Logistic Regression.

The second model that we tried was the Random Forest (RF). Since there are many hyperparameters to decide for the Random Forest, we used random grid search method with 5-folds cross-validation to save time. The grid utilized to randomize is shown in Table 1. For each dataset, the best parameters are chosen by considering the AUC-ROC as a performance measure. At the end, these implementations added 4 new results to ultimate result table.

The third model was the Decision Tree. Unlike the Random Forest algorithm, Decision Tree method fits only one tree and it does not take much time. Therefore, we could use the full grid search method instead

Table 1 The grid that was used to randomize for Random Forest.

Parameter	Levels			
n_estimators	200	350	500	
max_features	auto	sqrt		
max_depth	10	30	50	None
min_samples_split	2	5	10	
min_samples_leaf	1	2	4	
bootstrap	True	False		

of random grid search method. The grid that was used for the Decision Tree model is shown in Table 2. The reason for choosing larger min_samples_leaf than we did for the RF was to prevent the greediness of the Decision Tree algorithm. In addition to 10 different results generated by previous models, this method brought 4 new results to ultimate result table.

Table 2 The grid for Decision Tree.

Parameter	Levels		
criterion	gini	entropy	
max_depth	4	10	30
min_samples_leaf	10	20	30

The last method was a powerful method called Gradient Boosting (GB). Despite its power, the computational complexity makes the GB harder to implement because it takes longer than the other models that we used. To save time, we did not consider all the hyperparameters and focused only on the learning rate. We tested one low (0.1) and one high level (1.0) learning rates by 5 folds cross-validation. The other hyperparameters are not tested, and they are used with their default values.

Results

The best hyperparameter combinations for the models are chosen by considering AUC-ROC for each dataset. ROCs of the chosen models are represented in Figure 13 separately for each dataset.

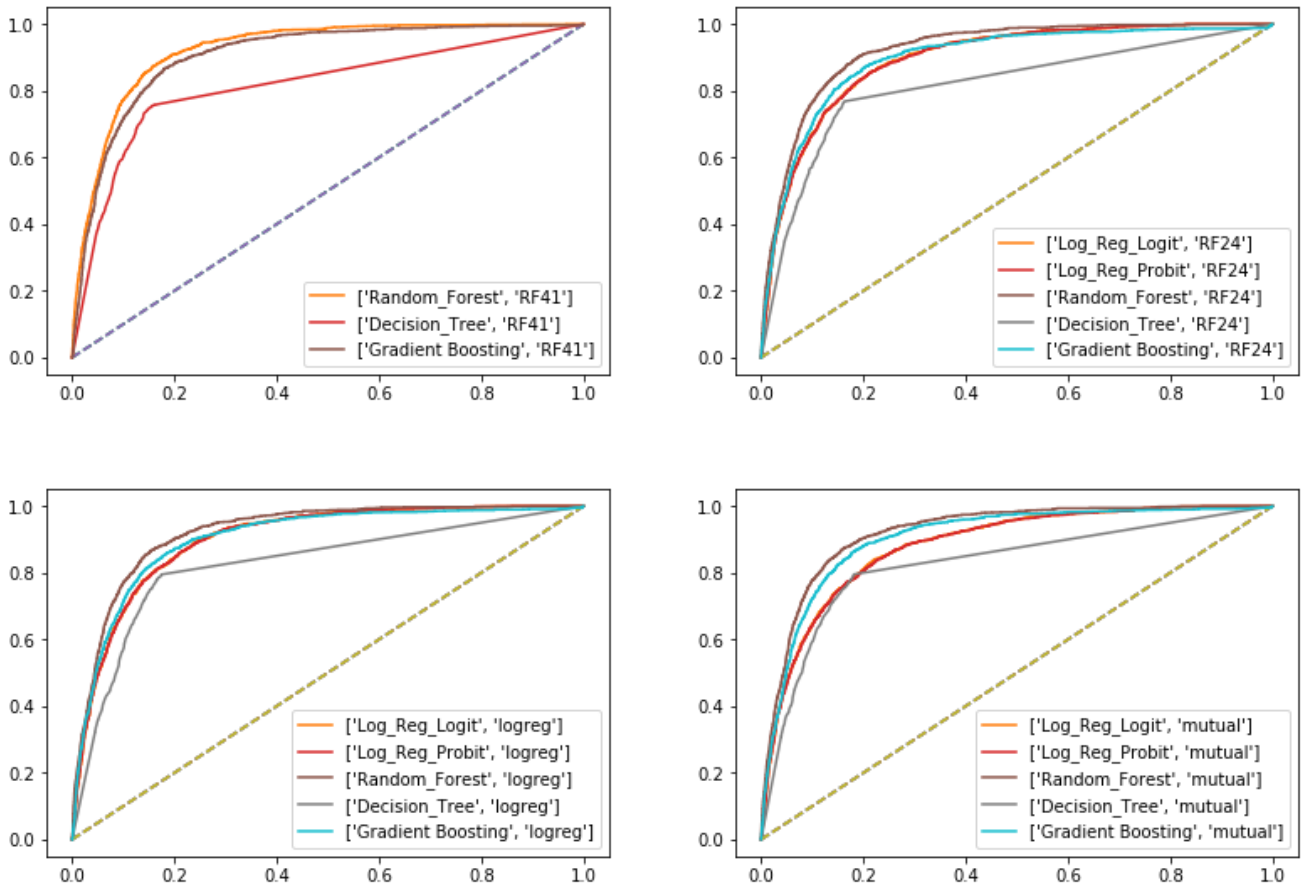

Figure 13 ROCs of the models for each dataset.

According to Figure 13, it seems like the Random Forest algorithm outperforms the other algorithms for each dataset in terms of AUC-ROC. However, since detecting which dataset results in the best performance is hard from the figure, we need to look at the result table sorted by AUC scores. The result table is shown in Figure 14.

	Method	Dataset	AUC	Accuracy
0	Random Forest	RF41	0.926906	0.839840
10	Random Forest	logreg	0.926651	0.840381
5	Random Forest	RF24	0.925120	0.838050
15	Random Forest	mutual	0.924275	0.837801
12	Gradient Boosting	logreg	0.909852	0.834804
7	Gradient Boosting	RF24	0.905878	0.835928
8	Logistic Regression Logit	logreg	0.904151	0.824399
9	Logistic Regression Probit	logreg	0.903787	0.821319
17	Gradient Boosting	mutual	0.901340	0.827395
2	Gradient Boosting	RF41	0.900574	0.833847
3	Logistic Regression Logit	RF24	0.898137	0.815575
4	Logistic Regression Probit	RF24	0.897698	0.812578
13	Logistic Regression Logit	mutual	0.885257	0.805586
14	Logistic Regression Probit	mutual	0.884378	0.803754
16	Decision_Tree	mutual	0.834179	0.763964
11	Decision_Tree	logreg	0.831638	0.769333
6	Decision_Tree	RF24	0.825189	0.771997
1	Decision_Tree	RF41	0.816007	0.764838

Figure 14 Result table sorted by AUC.

The result table shows us that the best results are obtained by fitting a Random Forest model to the RF41 data which includes 41 of the features. We know that the small differences among the first five models in the table above means that the order of the models can be changed easily with different seeds. However,

Table 3 Chosen Parameters of Best Performing Random Forest Model.

Parameter	Levels		
n_estimators	**200**	350	500
max_features	auto	**sqrt**	
max_depth	10	**30**	50
min_samples_split	2	**5**	10
min_samples_leaf	1	2	**4**
bootstrap	**True**	False	

since all of them are satisfactory, choosing the one at the top is acceptable. The chosen hyperparameters of the best model are given in Table 3 in green cells.

Apart from being the best prediction model in terms of AUC-ROC, Random Forest is also performed adequately in terms of accuracy. The accuracy of the Random Forest algorithm is about 83%. The confusion matrix is shown in Figure 15. Other metrics that are extracted from confusion matrix are as follows.

```
precision : [0.80379606 0.88537437]
recall    : [0.89894281 0.78057105]
fscore    : [0.8487111  0.82967616]
```

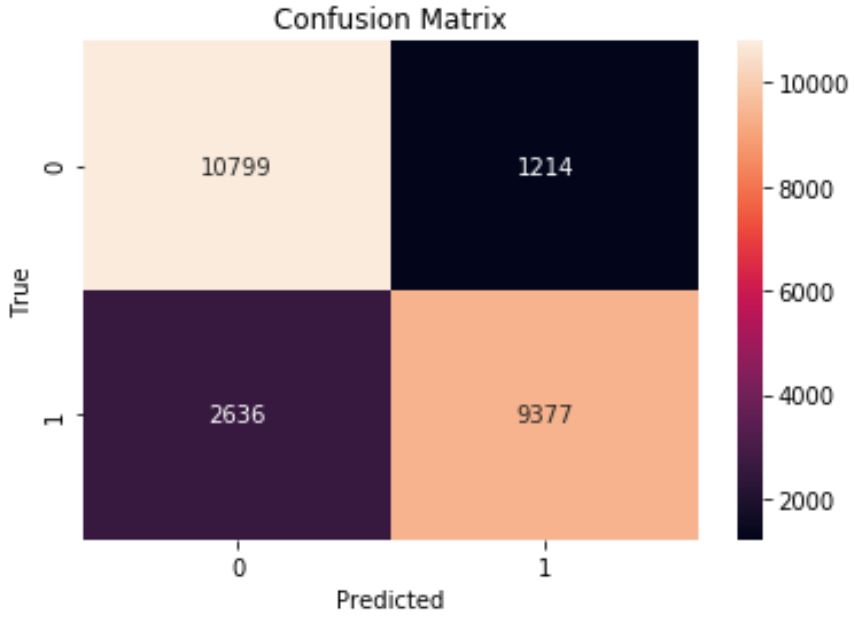

Figure 15 Confusion Matrix of the Random Forest model.

Conclusion

Predictions of whether a customer will purchase a term deposit or not can be made with approximately 83% accuracy by using 41 features of the customers. These features are as follows.

```
Index(['duration', 'day', 'age', 'balance', 'campaign', 'pdays',
       'poutcome_success', 'poutcome_failure', 'month_num', 'housing_no',
       'housing_yes', 'previous', 'month_may', 'month_oct', 'month_apr',
       'job_technician', 'marital_married', 'job_blue-collar', 'month_mar',
       'job_admin.', 'marital_single', 'education_tertiary', 'job_management',
       'education_secondary', 'month_aug', 'month_sep', 'loan_yes',
       'job_retired', 'job_services', 'education_primary', 'loan_no',
       'month_jul', 'marital_divorced', 'month_jun', 'month_feb', 'month_nov',
       'contact_telephone', 'contact_cellular', 'job_self-employed',
       'job_entrepreneur', 'job_unemployed', 'job_student'],
      dtype='object')
```

Although the RF predictions are satisfactory for the time being, when the customer calls increase with the new calls, preferences of prediction models can be changed, and simpler models such as Logistic Regression may be preferable.

With the help of this prediction model, success rates of salespeople can be improved because it helps them to target the right people, and eventually lead the bank to be more profitable and efficient in terms of revenue over time.

References

https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

https://medium.com/@kesarimohan87/model-selection-using-cross-validation-and-gridsearchcv-8756aac1e9d7

https://www.programcreek.com/python/example/82501/sklearn.preprocessing.MinMaxScaler

https://ipfs-sec.stackexchange.cloudflare-ipfs.com/datascience/A/question/24405.html

Examples on https://scikit-learn.org/