Predicting the Success of Bank Telemarketing

Capstone 1 Project

Author: Jin No
Date: March 2019
Github for presentation: https://github.com/Jwn758/SpringboardProjects/tree/master/CapstoneProject1

Table of Contents:

1. Introduction

Banks and insurance companies are all trying to look for more and more ways to reduce their operating expenses. Some banks have even gone so far to remove traditional “brick-and-mortar” locations and/or run their operations entirely online (e.g., Ally Financial, Robinhood). One large driver of a banks’ expense ratios includes marketing costs, and banks are constantly trying to find more ways to optimize their operational efficiencies.

While browsing the business section of Google News, I've seen more and more headlines repeating the same thing: "Financial Institution ABC is implementing AI & Machine learning models which are expected to increase our operational efficiencies by reducing costs by X%". After seeing this phrase repeated over and over again in quarterly and annual investor reports, one must wonder...is this really the case?? Do companies just love using buzzwords to keep investors happy? Do companies really expect to see double digit % reduction in expenses by implementing a machine learning model? What does this really mean?

I wanted to gain a better understanding of what all these headlines actually meant. For my Capstone I Project, I've obtained a dataset from a Portugease bank institution that contains 41188 rows and 21 columns. Each row represents a different marketing campaign (i.e. a phone call made by the bank to get a customer to purchase a term deposit), with the first 20 columns being different characteristics of the campaign and the last column indicating whether or not the customer purchased a term deposit. Although we can't actually measure how much this bank can save without additional data, what this exercise will tell us how accurately a bank can predict a successfull marketing campaign.

Summary:

- Problem: Predicting the likelihood of a bank customer purchasing a term deposit (i.e. CDs) using a dataset obtained from UCI's Marching Learning Repository
- Dataset: The dataset can be found here at http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

1.1 Data Source and Importation

The dataset was obtained from UCI's Machine Learning Repository and is related with direct marketing campaigns of a Portuguese banking institution. 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 not ('no') subscribed.

There are four datasets:

- bank-additional-full.csv with all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010), very close to the data analyzed in [Moro et al., 2014]
- bank-additional.csv with 10% of the examples (4119), randomly selected from 1), and 20 inputs.
- bank-full.csv with all examples and 17 inputs, ordered by date (older version of this dataset with less inputs).
- bank.csv with 10% of the examples and 17 inputs, randomly selected from 3 (older version of this dataset with less inputs).

The smallest datasets are provided to test more computationally demanding machine learning algorithms (e.g., SVM). The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).
Given the description of each dataset, we will go with "bank-additional-full.csv".

1.2 Data Cleaning

Taking a look at the variables available for each marketing campaign:

Out[3]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome	emp.var.rate	co
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1	
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1	
2	37	services	married	high.school	no	yes	no	telephone	may	mon	...	1	999	0	nonexistent	1.1	
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	...	1	999	0	nonexistent	1.1	
4	56	services	married	high.school	no	no	yes	telephone	may	mon	...	1	999	0	nonexistent	1.1	

5 rows × 21 columns

Inspecting quality of dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
age               41188 non-null int64
job               41188 non-null object
marital           41188 non-null object
education         41188 non-null object
default           41188 non-null object
housing           41188 non-null object
loan              41188 non-null object
contact           41188 non-null object
month             41188 non-null object
day_of_week       41188 non-null object
duration          41188 non-null int64
campaign          41188 non-null int64
pdays             41188 non-null int64
previous          41188 non-null int64
poutcome          41188 non-null object
emp.var.rate      41188 non-null float64
cons.price.idx    41188 non-null float64
cons.conf.idx     41188 non-null float64
euribor3m         41188 non-null float64
nr.employed       41188 non-null float64
y                 41188 non-null object
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

There are 20 variables that can affect the outcome Y. The 20 variables are of either type object, int, or float, with a majority of the variables being an object. There seems to be no missing values.

A description of each variable can be found below:

Attribute Information:

Input variables:
bank client data:
1 - age (numeric)
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5 - default: has credit in default? (categorical: 'no','yes','unknown')
6 - housing: has housing loan? (categorical: 'no','yes','unknown')
7 - loan: has personal loan? (categorical: 'no','yes','unknown')
related with the last contact of the current campaign:
8 - contact: contact communication type (categorical: 'cellular','telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
other attributes:
12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
social and economic context attributes
16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target):
21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Right away, I hypothesized some variables as having little to no impact on Outcome Y. Furthermore, even though we didn't find any NaN values, it looks like we should do some high-level cleaning (e.g. 999 for "pdays" column) and discard the "duration" variable for the time being given that the description of the variable says this "should only be includued for benchmark purposes and should be discarded if the intention is to have a realistic predictive model".

Out[5]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	campaign	pdays	previous	poutcome	emp.var.rate	cons.p
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	1	999	0	nonexistent	1.1	
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	1	999	0	nonexistent	1.1	
2	37	services	married	high.school	no	yes	no	telephone	may	mon	1	999	0	nonexistent	1.1	
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	1	999	0	nonexistent	1.1	
4	56	services	married	high.school	no	no	yes	telephone	may	mon	1	999	0	nonexistent	1.1	

In addition to each individual variable, I thought some variables could be grouped into different types or groups and I hypothesize certian groups of variables to be more relevant in predicting the outcome variable Y. To get a better understanding of how each group of variables might affect outcome Y, I split up the 19 (excludes duration) variables into 4 groups:

1. Demographic Characteristics (age, job, marital, education)
2. Socioeconomic Variables (default, housing, loan)
3. Marketing Campaign Variables (contact, month, day_of_week, campaign, pdays, previous, poutcome)
4. Economic Indicators (emp.var.rate, cons.conf.idx, euribor3m, nr.employed)

For convenience, I've also divided the variables into categorical and numerical variables.

I've also added a column encoding the output variable into 0 for "no" and 1 for "yes"

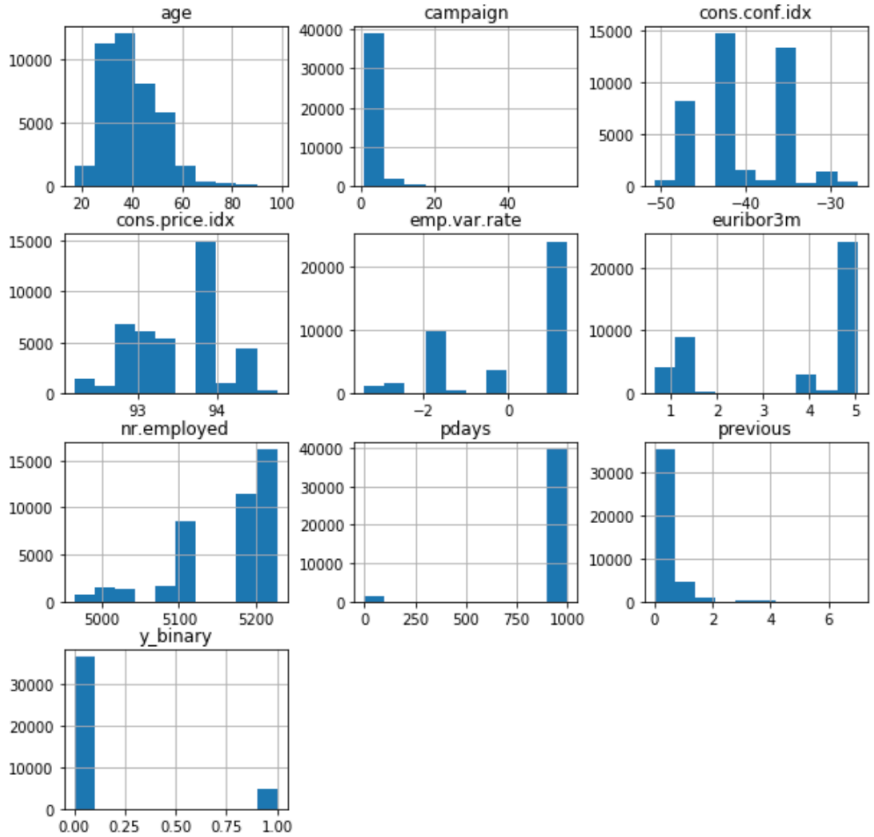
Out[8]:

	y	y_binary
41183	yes	1
41184	no	0
41185	no	0
41186	yes	1
41187	no	0

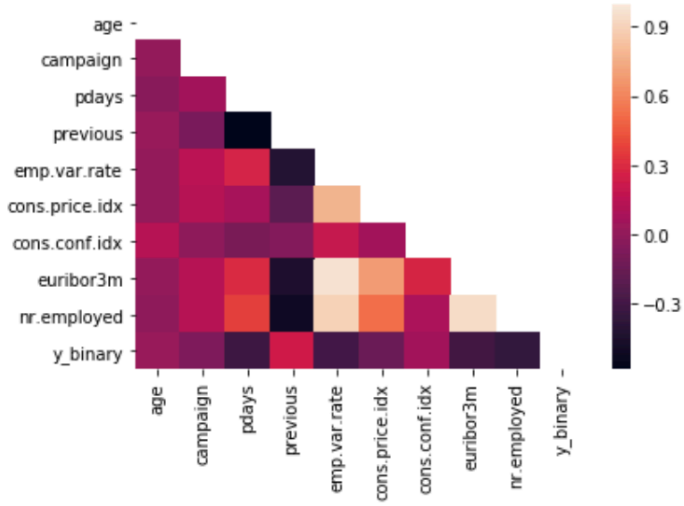
2. Exploratory Data Analysis

We will explore the effects of Demographic Characteristics, Financial Indicators, Marketing Campaign Characteristics, and Economic Indicators on Outcome Y - did the bank customer purchase a term deposit?

Before we get into any machine learning models, I wanted to explore the relationship between our variables and outcome variable Y. Right away, I wanted to see how our numerical variables correlate with the outcome variable and each other. To do this, we will perform a simple correlation matrix but we should first investigate the actual values for each numerical variable:

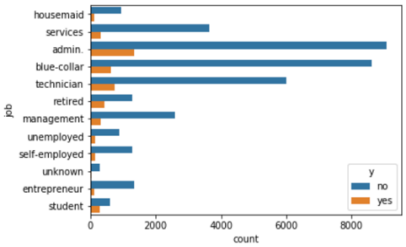


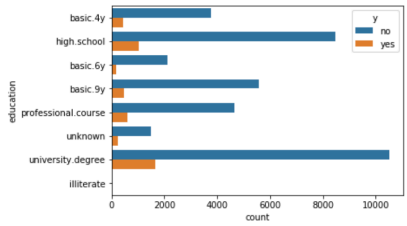
With the above exhibits, we can note that some of our numerical variables have extreme outliers. For example, most customers had a value of 999 for "pdays". We noted this earlier when looking through the descriptions of each variable. This is something we should note in future analyses.

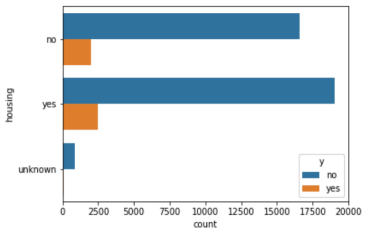


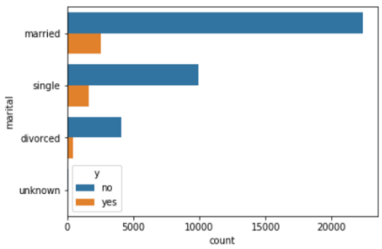
Interestingly enough, there seems to be a high positive correlation amongst our Economic Indicator variables (e.g. emp.var.rate and euribo3m) but no strong correlations between our Outcome variable Y and all others. We'll look into this further later.

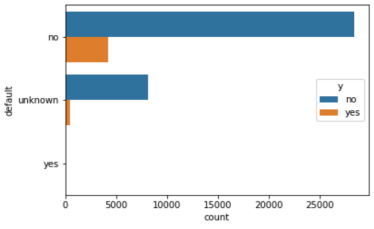
Additionally, to get a better sense of our categorical variables, I wanted to visualize each variable by the outcome variable:

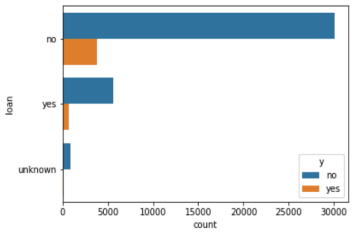


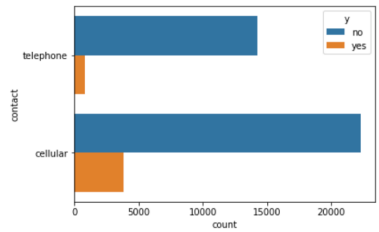


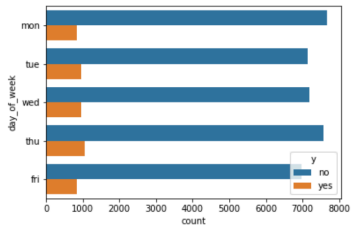


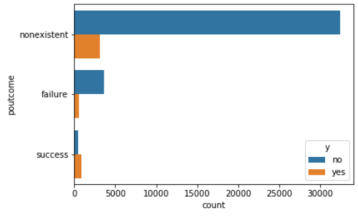






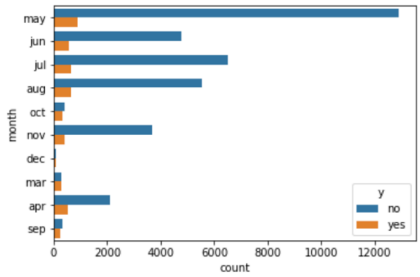






Although we cannot make any conclusions about a categorical variable being more highly correlated with yes or no, what is useful is how each categorical variable is distributed by the outcome variable.

For example, when we look at just month (shown again below), we can note a couple of interesting observations:

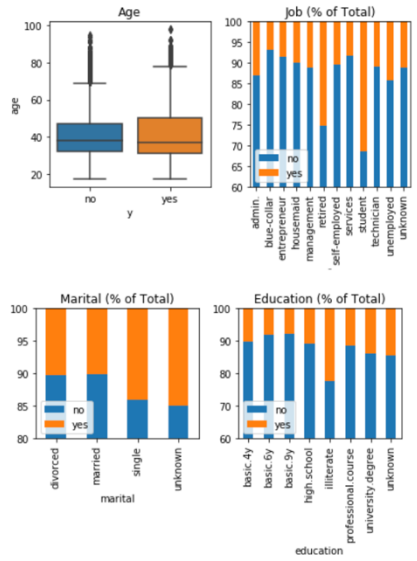


•May was the most active month for marketing campaigns
•Both "Yes" and "No" occured most frequently in May (This is not to say that people were more likely to say yes or no in May than other months)
•March was the month with the most even split between "Yes" and "No"

Through these preliminary exhibits, we also noted a good amount of "unknown" values in certain categorical variables such as "education" and "default". We may want to make a note of this and consider taking them out if they interfere with our machine learning model when we preprocess our feature variables.

2.1 Demographic Characteristics

Includes Age, Job, Marital Status, Education



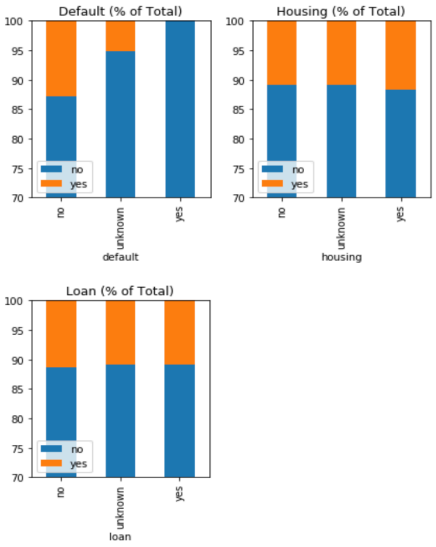
After looking at our demographic variables, we can see a trend that people who bought a term deposit are more likely to have the following characteristics:

• Age: Older
• Job: Retired or Student
• Marital: Single
• Education: Illtirate

It doesn't seem right that illerature customers were more likely to purchase a term deposit. Perhaps there is a second variable going on...are illiterate customers mainly older and is it the case that education might not matter? We'll look into this further later on with statistical testing.

2.2 Socioeconomic Variables

Includes housing, loan, default

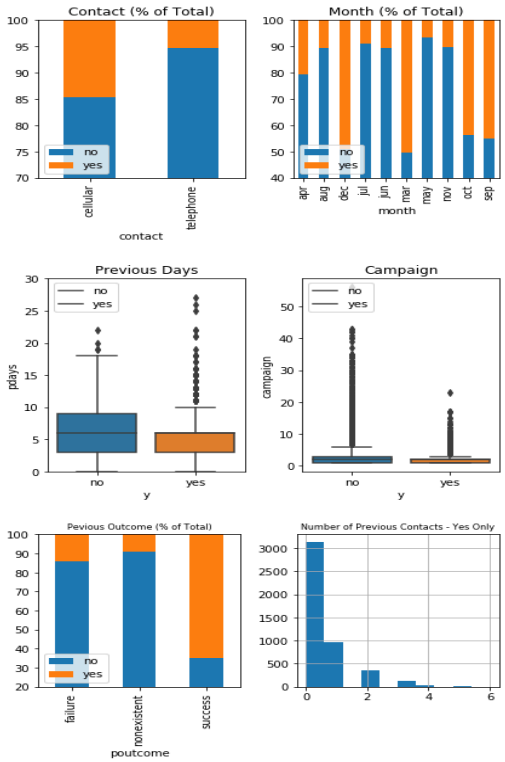


After looking at our financial indicator variables, we can note the following trends:

•Default: People who default do not purchase term deposits at all. People who do not default are more likely to purchase a term deposit than those who do default
•Housing: No difference between no housing, yes housing, and unknown housing
•Loan: No difference in no loan, yes, loan, and unknown loan

2.3 Marketing Campaign Variables

Includes contact, month, day_of_week, campaign, pdays, previous, and poutcome

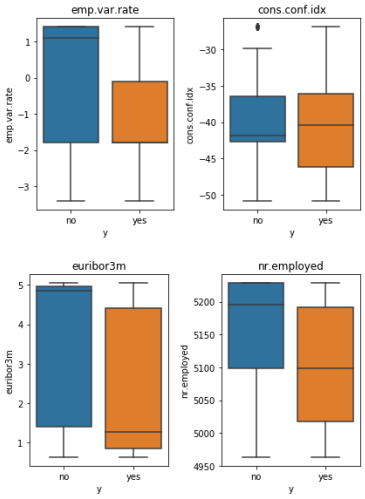


We can note the following about previous marketing campaigns:

•Contact: Customers more likely to purchase term deposit with cellphone than telephone
•Month: Top 3 months were March, June, and September
•Duration: Longer campaigns were more effective
•Campaign: No difference
•Previous Days: Most people who purchased a term deposit purchased it when the previous failed contact was < 5 days
•Previous Contact: Highest success seen in 3-4 previous contacts
•Previous Outcome: To no surprise, those that previously purchased a term deposit were most likely to purchase another term deposit rather than those who didn't or are unknown

2.4 Economic Indicators

Includes emp.var.rate, cons.conf.idx, euribor3m and nr.employed



These economic indicators will be more relevant once we get to our predictive modelling, so we'll ignore these for now.

3. Statistical Testing

We've observed the following interesting trends:

- Are students or retired people more likely to purchase a term deposit?
- Are single people more likely to purchase a term deposit?
- Are there no differences in people who own a home and don't own a home?
- Are cellphones truly more effective than telephones?
- Is there a significant difference between months?

For the purposes of this capstone project, these trends will be explored in more detail with statistical testing:

- Are students or retired people more likely to purchase a term deposit?
- Are single people more likely to purchase a term deposit?

3.1 Effect of Ocupation on the Success of Marketing Campaigns

Proportion Z-Test

Students	Results
Null Hypothesis: Are students less likely to purchase a term deposit?	**Z-test Statistic** of 18.4 with
Test Used: Proportion Z-test with alpha of 0.01	**p-value** of <0.0001
Test Statistic: Z-test Statistic	

Retired Customers	Results
Null Hypothesis: Are retired people less likely to purchase a term deposit?	**Z-test Statistic** of 18.4 with
Test Used: Proportion Z-test with alpha of 0.01	**p-value** of <0.0001
Test Statistic: Z-test Statistic	

3.2 Effect of Marital Status on the Success of Marketing Campaigns

Single Customers	Results
Null Hypothesis: Are single people less likely to purchase a term deposit?	**Z-test Statistic** of 18.4 with
Test Used: Proportion Z-test with alpha of 0.01	**p-value** of <0.0001
Test Statistic: Z-test Statistic	

Since the z-test z-scores were 18.4 for Students and 17.6 for Retired customers with extremely low p-values, we can interpret this as the callback proportions being more than 17 standard deviations away from the mean of 0 difference. We should reject the null hypothesis. In other words, we can confidently say that there is statistical difference in marketing success rates for single, students, and retired customers.

4. Predictive Modeling

4.1 Data Preprocessing

Before we perform any predictive modeling, we'll process our variables using the sklearn package. Before we encode our feature variables, we should find a way to deal with our unknown values in our categorical feature variables

```
admin.           10422
blue-collar       9254
technician        6743
services          3969
management        2924
retired           1720
entrepreneur      1456
self-employed     1421
housemaid         1060
unemployed        1014
student            875
unknown            330
Name: job, dtype: int64
NAs for job : 330
Most frequently occuring value is: admin.


married     24928
single      11568
divorced     4612
unknown        80
Name: marital, dtype: int64
NAs for marital : 80
```

```
Most frequently occuring value is: married


university.degree      12168
high.school             9515
basic.9y                6045
professional.course     5243
basic.4y                4176
basic.6y                2292
unknown                 1731
illiterate                18
Name: education, dtype: int64
NAs for education : 1731
Most frequently occuring value is: university.degree


no         32588
unknown     8597
yes            3
Name: default, dtype: int64
NAs for default : 8597
Most frequently occuring value is: no


yes        21576
no         18622
unknown      990
Name: housing, dtype: int64
NAs for housing : 990
Most frequently occuring value is: yes


no         33950
yes         6248
unknown      990
Name: loan, dtype: int64
NAs for loan : 990
Most frequently occuring value is: no


cellular     26144
telephone    15044
Name: contact, dtype: int64
NAs for contact : 0
```

```
Most frequently occuring value is: cellular


may    13769
jul     7174
aug     6178
jun     5318
nov     4101
apr     2632
oct      718
sep      570
mar      546
dec      182
Name: month, dtype: int64
NAs for month : 0
Most frequently occuring value is: may


thu    8623
mon    8514
wed    8134
tue    8090
fri    7827
Name: day_of_week, dtype: int64
NAs for day_of_week : 0
Most frequently occuring value is: thu


nonexistent    35563
failure         4252
success         1373
Name: poutcome, dtype: int64
NAs for poutcome : 0
Most frequently occuring value is: nonexistent
```

Now that we've imputed unknown values in our categorical features with the most frequently occuring value, we can proceed with encoding them. Let's check to make sure we have no more unknowns:

```
admin.           10752
blue-collar       9254
technician        6743
services          3969
management        2924
retired           1720
```

```
entrepreneur      1456
self-employed     1421
housemaid         1060
unemployed        1014
student            875
Name: job, dtype: int64
married     25008
single      11568
divorced     4612
Name: marital, dtype: int64
university.degree      13899
high.school             9515
basic.9y                6045
professional.course     5243
basic.4y                4176
basic.6y                2292
illiterate                18
Name: education, dtype: int64
no     41185
yes        3
Name: default, dtype: int64
yes    22566
no     18622
Name: housing, dtype: int64
no     34940
yes     6248
Name: loan, dtype: int64
cellular     26144
telephone    15044
Name: contact, dtype: int64
may    13769
jul     7174
aug     6178
jun     5318
nov     4101
apr     2632
oct      718
sep      570
mar      546
dec      182
Name: month, dtype: int64
thu    8623
mon    8514
wed    8134
```

```
tue    8090
fri    7827
Name: day_of_week, dtype: int64
nonexistent    35563
failure         4252
success         1373
Name: poutcome, dtype: int64
```

Now we can encode our feature variables using sklearn's LabelEncoder function. Before we do any sort of predictive modeling, we also split our dataframe into a feature array and target array.

4.2 Training and Testing Datasets

Then, we will split our dataset into training and testing datasets.

```
Size of x training set: (28831, 19)
Size of y testing set: (12357,)
```

4.3 Training Model

We will try different classification specific classifiers to train our model. The following classifiers will be used:

- KKN
- Logistic Regression
- Random Forest

KNN

Let's first try K-nearest neighbors and evaluate based on accuracy score:

```
0.898761835396941
```

Logistic Regression

```
0.8986809096058914
```

Random Forest

```
0.8922068463219228
```

We notice that all 3 of our classifiers performed pretty well. However, we should be careful with using right away we accuracy score as a useful metric to evaluate our models because of the imbalance of "Yes" and "No" in our Outcome variable. This is a classic class imbalance problem. As such, we will use other evaluation metrics.

4.4 Evaluating Models

Since we experience a class imbalance problem, we will use more nuanced evaluation metrics. To evaluate the performance of our models, we will use AUROC scores:

KNN

```
Final AUROC score: 0.7696
```

Logistic Regression

```
Final AUROC score: 0.7579
```

Random Forest

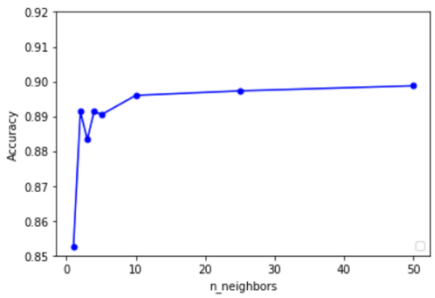
```
Final AUROC score: 0.7520
```

Using AUROC scores, KKN classifer performed the best. In the next section, we will try different hyperparameters for each model before we come to a conclusion about the best performing model.

4.5 Model Tuning

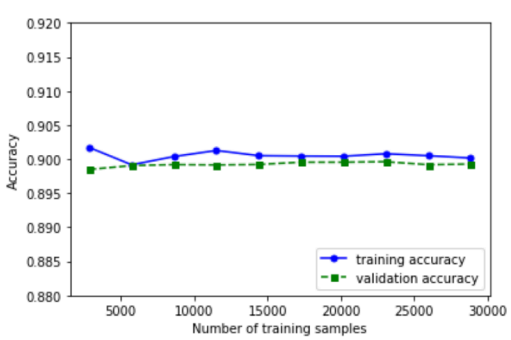
KNN - Hyperparameter Tuning and Evaluation

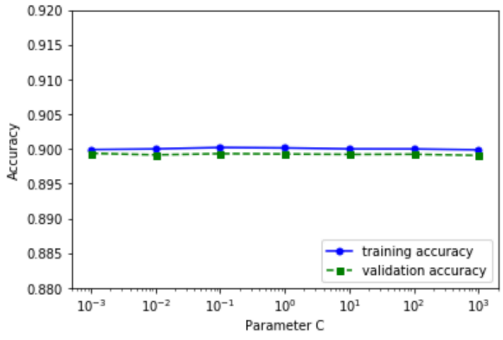
```
x = 1
x = 2
x = 3
x = 4
x = 5
x = 10
x = 25
x = 50
```




```
Final AUROC score: 0.7696
```

Logistic Regression - Hyperparameter Tuning and Evaluation



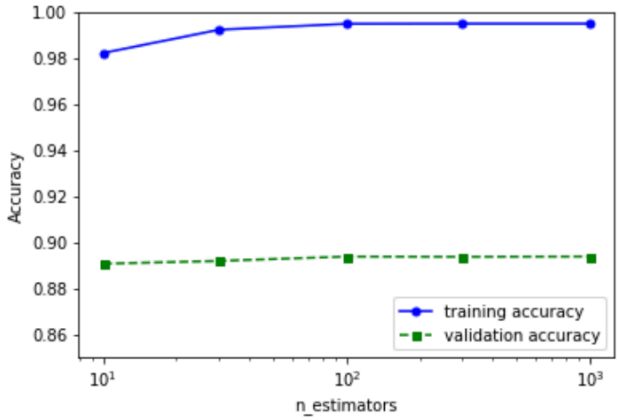



```
[0.89937687 0.89916646 0.89932832 0.89927976 0.8992312  0.89924739
 0.89908554]
```

Looks like we get the best score with c = 0.1

```
Final AUROC score: 0.7579
```

Random Forest - Hyperparameter Tuning and Evaluation



Looks like we get the best score with n_estimators = 100

```
Final AUROC score: 0.7783
```

4.6 Model Selection

We should pick the random forest classifier model with n_estimators of 100 as it performed the best, when evaluating our models based off AUROC scores.

5. Conclusion

5.1 Results

With our RFC classifier, we are able to predict whether or not a customer will purchase a term deposit ~78% of the time. This is fairly good as it beats a random predictor guessing yes or no, as the AUROC score for that would be 0.5.

The bank can implement this predictor and focus their marketing efforts on the 78% of the customers that they can accurately predict the outcome for. According to EMI, a marketing agency specialized in banking, the top 40 banks spent nearly 14b in 2017, with 17 of those banks reporting double-digit growth since the prior year. For sake of simplicity, if we assume each customer for each bank is allocated the same marketing spend and that banks do not currently use any machine learning techniques or strategic prioritization, the top 40 banks could've saved nearly ~$3b. Now I can see why machine learning applications in business operations are getting all the buzz. Although it may have seemed crazy to me before that data scientists at F500 companies go crazy trying to get that extra 0.01% accuracy, it makes sense to me now how that 0.01% increase in accuracy can make all the difference.

5.2 Final Thoughts

Although 0.78 AUROC is fairly good, its not great. If I had more time, it would be useful to focus on the following to see if any improvements to my classifiers are made:

- Feature Engineering:
  - Getting rid of outliers in each variable in my feature array
  - Trying different combinations of one-hot encoder and label encoder of categorical variables to improve the "order" (e.g., perhaps a more defined ranking system of different education or occupation values)
  - More advanced methods of imputing unknown values, instead of just using the mode
- Model Tuning:
  - Trying more hyperparameters for each classifier
  - Trying more classifiers