Propensify – A Propensity Model to identify how likely certain target groups customers respond to the marketing campaign

ABSTRACT

In today's world, where the market is highly competitive and volatile, companies have realized that in order to be successful, they need to have strong customer acquisition strategies. These strategies are the driving force for the success of the company in terms of revenue and brand value. the building blocks for the development of these strategies are the data that are available to the company. Most companies today have realized that data-driven decisions are the key to make any informed decision. Customer data is one such asset which is vital in understanding the customer base. The insights from this data can be key in developing strategies for new customer acquisition as well. Companies today drive various marketing campaigns through various channels in order to effectively reach its target audience. Most often than not, they become high-cost strategies. Companies incur loss when these expensive strategies do not yield favourable outcomes. Thus, a cost-effective strategy is the need of the hour for them to be successful. This study aims in demonstrating the fact that data-driven campaigns are highly effective in being cost-efficient and get good return on investment. This project showcases a propensity model to identify potential customers leveraging various data science concepts.

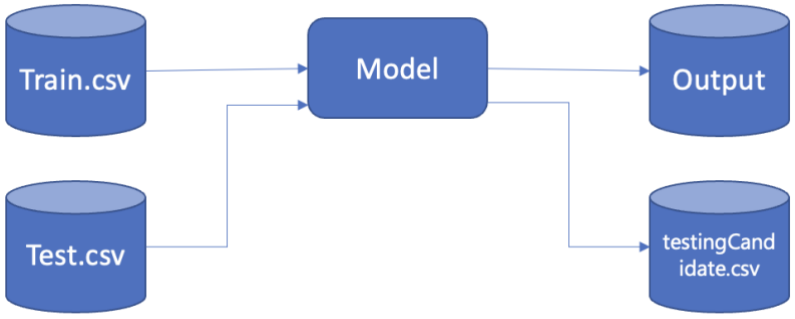
GOAL

The primary goal is to build a propensity model predicts the chance that individuals, leads and customers might opt in for. The model will use various statistical analyses to assess the customer behavior and use Machine Learning to predict the probability of a customer responding to a particular marketing campaign strategy. This project follows a systematic approach of the following concepts:

- Data Analysis – use statistical concepts to understand the data. This includes data preprocessing, exploratory data analysis (EDA) and visual representation of data
- Modeling – Building various ML models to get probabilities of a customer responding to a particular scenario. Apart from model building, this step also includes visual representation of model and model evaluation which helps in choosing the best model for this problem
- Model deployment plan – changing the code into a reproduceable form that can be reused and can be used in a production environment

Architecture

The model has the following flow:



The input given to train the model is ‘train.csv’. Various models will be trained with this data, and we will assess the best model to use for this project. Then the best model will be used to predict the outcome (probability of yes/no) with ‘test.csv’

Data

There are 2 datasets provided. They are:

1. Train.csv which contains historical data of customers who have responded in yes/no.
2. Test.csv which contains a list of potential customers to whom to market. We will be predicting whether the customer will say yes/no on this data.

The ‘Train.csv’ file contains the following columns:

Name	Description
custAge	The age of the customer (in years)
profession	Type of job
marital	Marital status
schooling	Education level
default	Has a previous defaulted account?
housing	Has a housing loan?
loan	Has a personal loan?
contact	Preferred contact type
month	Last contact month
day_of_wee	Last contact day of the week
campaign	Number of times the customer was contacted
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)
previous	Number of contacts performed before this campaign and for this client
poutcome	Outcome of the previous marketing campaign
emp.var.rate	Employment variation rate - quarterly indicator
cons.price.id	Consumer price index - monthly indicator
cons.conf.idx	Consumer confidence index - monthly indicator
euribor3m	Euribor 3 month rate - daily indicator
nr.employed	Number of employees - quarterly indicator
pmonths	Number of months that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
pastEmail	Number of previous emails sent to this client
responded	Did the customer respond to the marketing campaign and purchase a policy?

The column ‘responded’ will be used as the target variable to train the model. The corresponding value in the ‘test.csv’ will be ‘propensity’, which will be final predicted output.

From here on in this report, ‘train.csv’ will be called as ‘input data’

Data Analysis

The columns and its corresponding inputs are:

```
RangeIndex: 8240 entries, 0 to 8239
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   custAge         6224 non-null   float64
 1   profession      8238 non-null   object
 2   marital         8238 non-null   object
 3   schooling       5832 non-null   object
 4   default         8238 non-null   object
 5   housing         8238 non-null   object
 6   loan            8238 non-null   object
 7   contact         8238 non-null   object
 8   month           8238 non-null   object
 9   day_of_week     7451 non-null   object
 10  campaign        8238 non-null   float64
 11  pdays           8238 non-null   float64
 12  previous        8238 non-null   float64
 13  poutcome        8238 non-null   object
 14  emp.var.rate    8238 non-null   float64
 15  cons.price.idx  8238 non-null   float64
 16  cons.conf.idx   8238 non-null   float64
 17  euribor3m       8238 non-null   float64
 18  nr.employed     8238 non-null   float64
 19  pmonths         8238 non-null   float64
 20  pastEmail       8238 non-null   float64
 21  responded       8238 non-null   object
 22  profit          930 non-null    float64
 23  id              8238 non-null   float64
```

As seen above, there are a lot of null values in various columns. We will be handling through methods like removing, imputing, etc.

Data Cleaning

We will list the cleaning process done in a systematic step by step way:

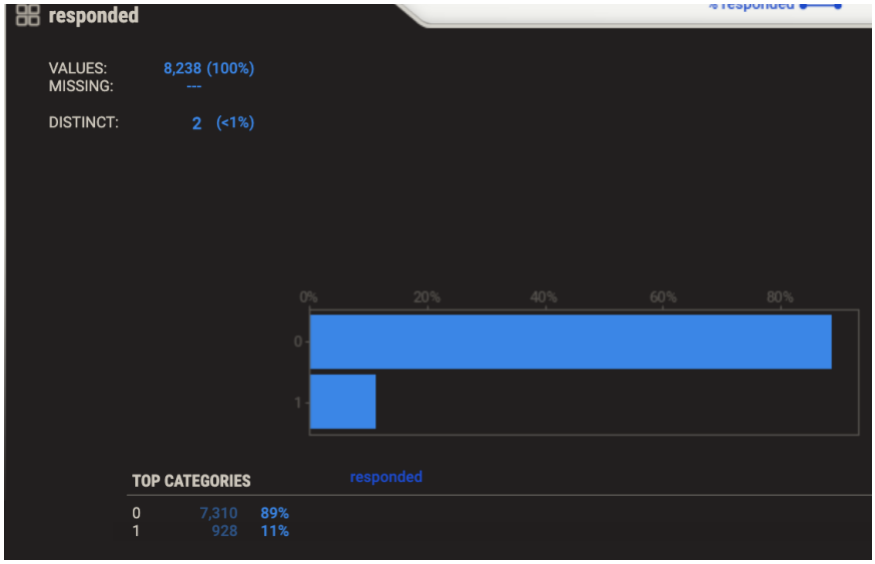
1. We will be removing the columns ‘id’ and ‘profit’ as they are extra columns than what is given in the list of columns to consider
2. To impute missing data in key columns, we have used the following methods:
    a. For ‘custAge’, we have taken the median value and imputed to the null values
    b. For ‘schooling’, we have taken the mode of the column
    c. For ‘day_of_week’, we are imputing the value ‘unknown’ for null values
3. In the ‘pdays’ column, there were 999 values for instances where the customers were called for the first time. There are a significant no of records with that value. So, we replace it with ‘-1’ just so that it wouldn’t be considered as an outlier

Exploratory Data Analysis

We have used the ‘sweetviz’ package to visualize the data along with its features.

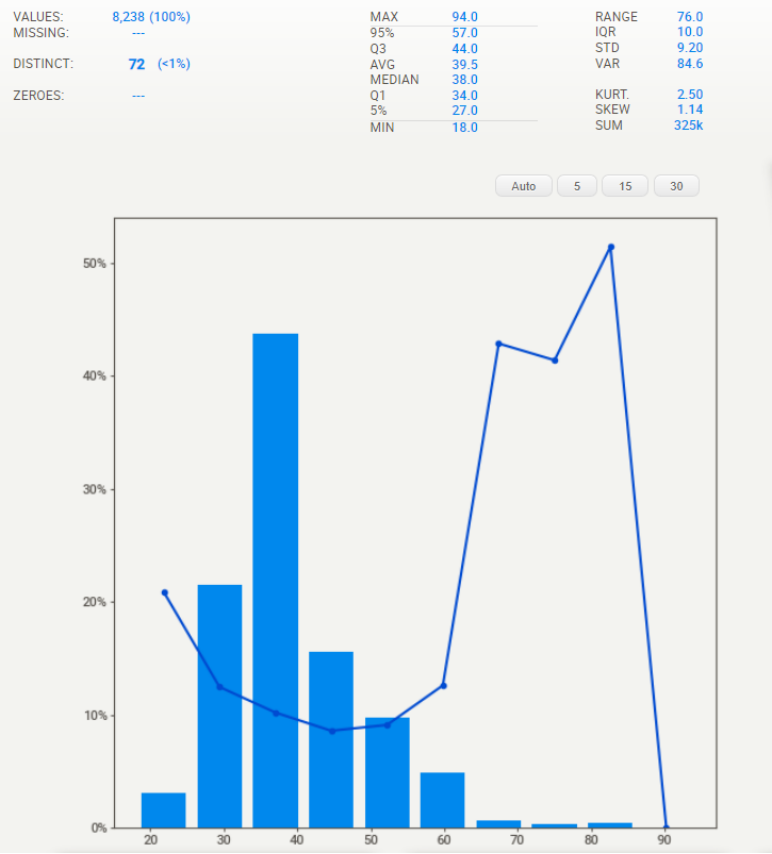
Let us analyze key columns:

1. responded



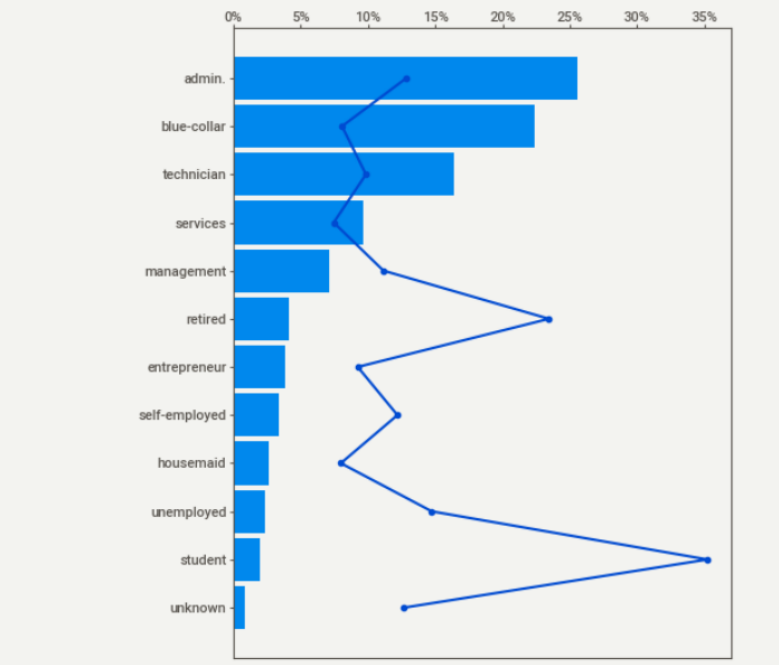
From the above picture, we can see that majority of the customers have responded ‘0’. It is to be noted that 0:’no’ and 1:’yes’

2. custAge



We can clearly see that there has been good variation in the target variable across all age groups. There aren't any outliers in this column.

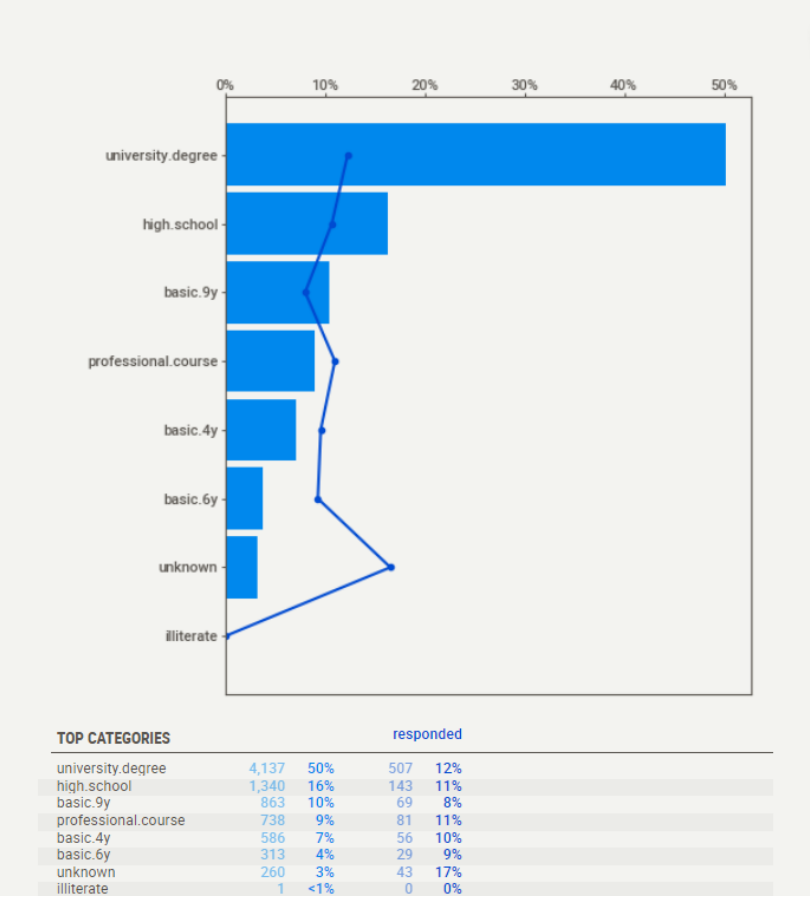
3. Profession



TOP CATEGORIES			responded	
admin.	2,102	26%	270	13%
blue-collar	1,847	22%	149	8%
technician	1,351	16%	133	10%
services	792	10%	59	7%
management	583	7%	65	11%
retired	337	4%	79	23%
entrepreneur	314	4%	29	9%
self-employed	279	3%	34	12%
housemaid	213	3%	17	8%
unemployed	190	2%	28	15%
student	159	2%	56	35%
unknown	71	<1%	9	13%

Here, students react highly to the campaign and retired people also have opted for insurance based on the data

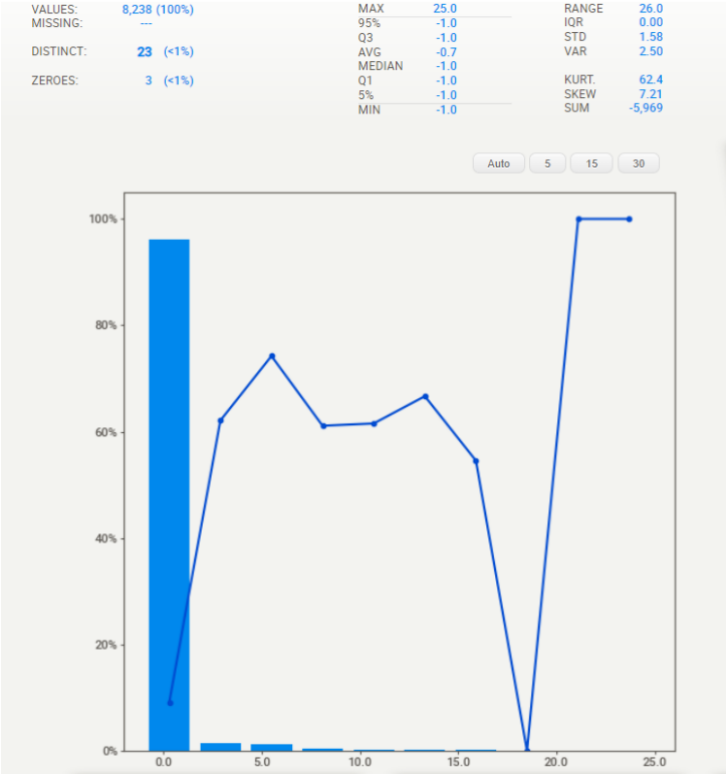
4. Schooling



TOP CATEGORIES			responded	
university.degree	4,137	50%	507	12%
high.school	1,340	16%	143	11%
basic.9y	863	10%	69	8%
professional.course	738	9%	81	11%
basic.4y	586	7%	56	10%
basic.6y	313	4%	29	9%
unknown	260	3%	43	17%
illiterate	1	<1%	0	0%

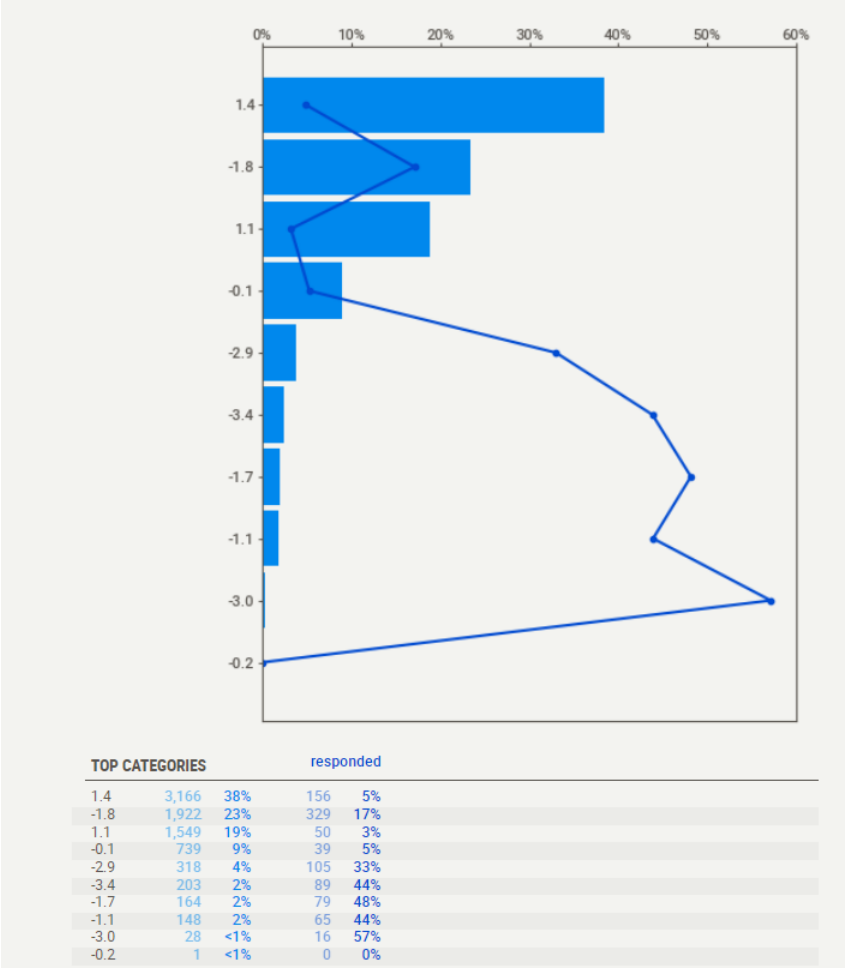
With respect to schooling, people who have a degree and who have prof course have high customer presence

5. Pdays



Since we had replaced 999 as -1, we can now see that the min. value has been changed from 0 to -1. We can see an inverse relationship between the days interval and response. The more the value, relatively higher are the chances that a customer would opt in

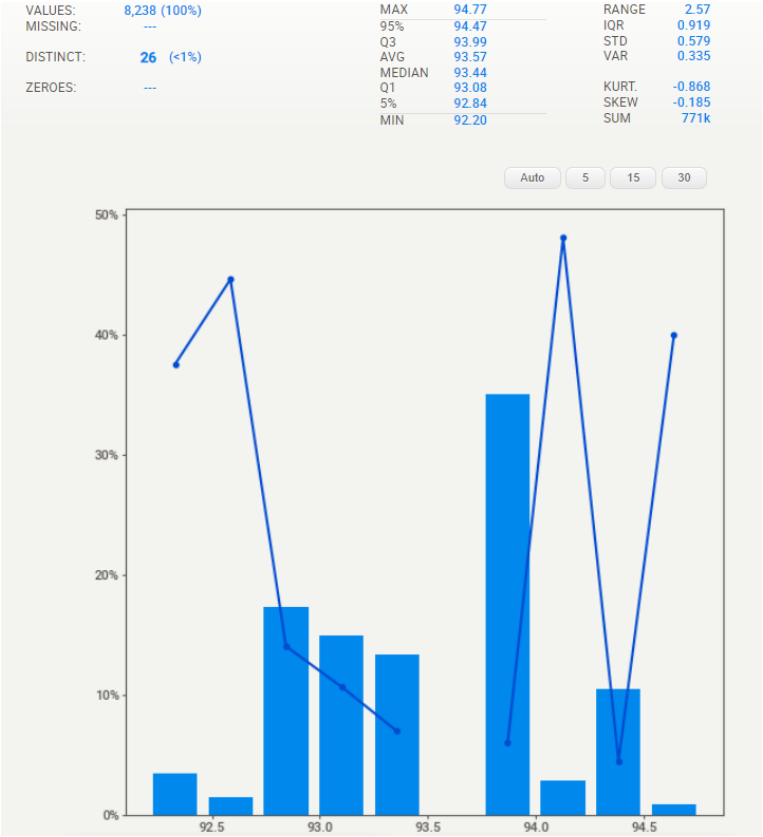
6. emp.var.rate



TOP CATEGORIES			responded	
1.4	3,166	38%	156	5%
-1.8	1,922	23%	329	17%
1.1	1,549	19%	50	3%
-0.1	739	9%	39	5%
-2.9	318	4%	105	33%
-3.4	203	2%	89	44%
-1.7	164	2%	79	48%
-1.1	148	2%	65	44%
-3.0	28	<1%	16	57%
-0.2	1	<1%	0	0%

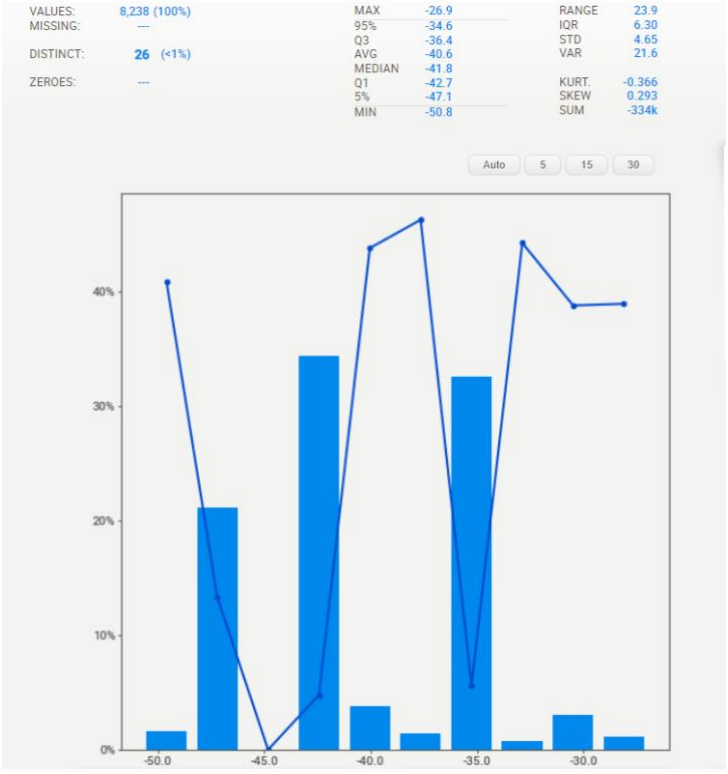
The lesser the rate, the more the chance of customer saying yes. We can see that negative values have more % of chances

7. cons.price.idx



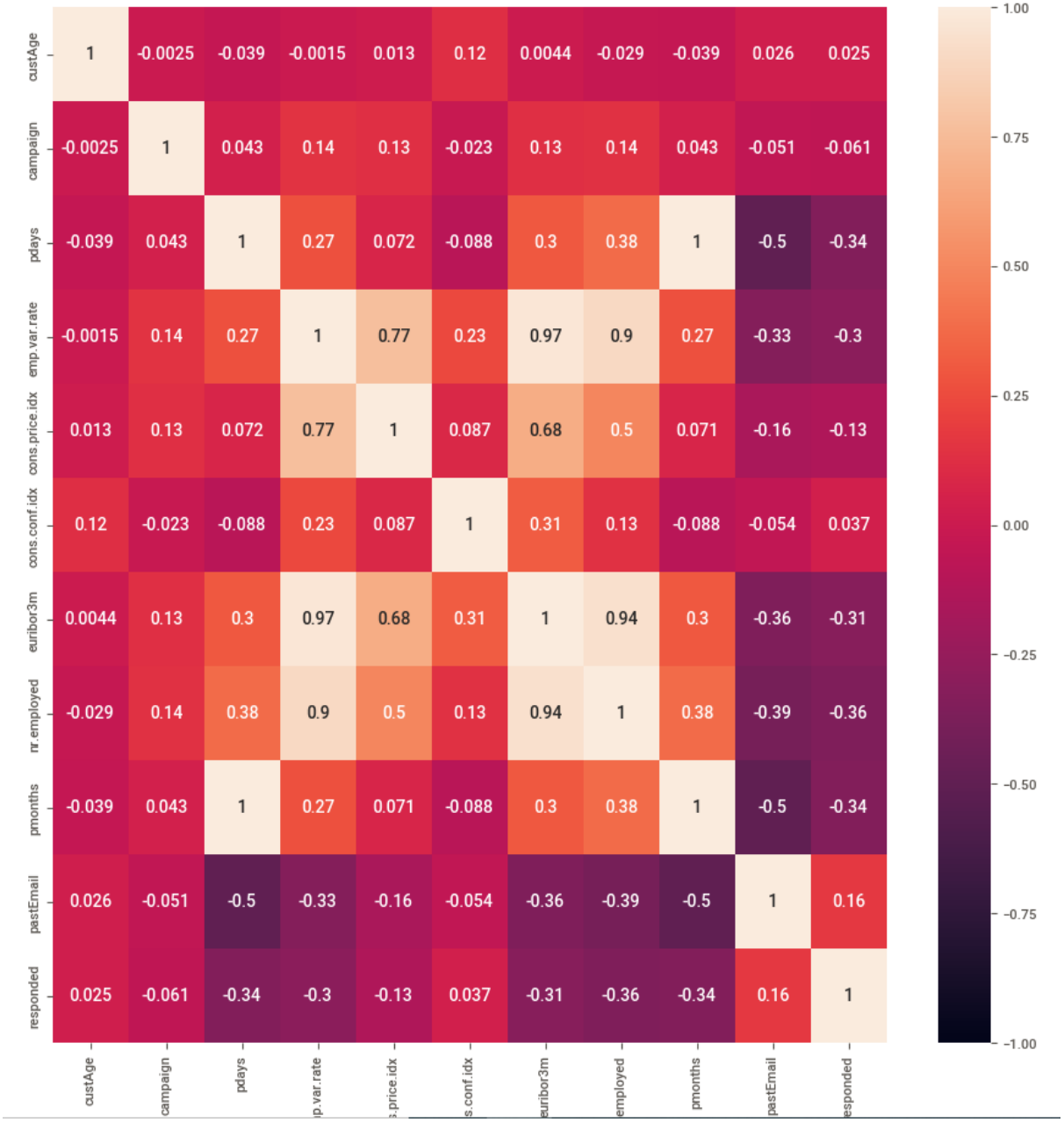
We can see significant changes in the response variable with respect to the consumer price index. This shows that it's an important indicator.

8. cons.conf.idx



We can see significant changes in the response variable with respect to the consumer confidence index. This shows that it's an important indicator.

Correlation Analysis



we will be removing 'pmonths' column because in essence, both 'pdays' and 'pmonths' are very similar and highly correlated. The former has a high variability with the output variable, so we are keeping it.

Feature Engineering

In this section, we will see all the steps done to prepare the data for modeling

Encoding

In order to transform the categorical data to a model interpretable data type, we are converting it to a numerical field. We are using One-hot encoding method for this project.

The columns transformed are:

1. schooling
2. profession
3. marital
4. default
5. contact
6. poutcome

Scaling

The numerical fields in the dataset are not standardized. For instance, custAge, pdays and cons.price.idx are in tens but nr.employed are in thousands. So, modeling with this will give spurious results. Hence, we use Scaling to standardize the data. Standard Scaler algorithm is used on this dataset.

Modeling

The propensity model is a classification problem. So, will be using various classification models to train our dataset. The models considered are:

1. Logistic Regression
2. Decision Tree
3. Random Forest
4. XGBoost

We will be selecting the best model based on the accuracy metrics in the model evaluation phase.

Sampling

For class imbalance, we are adding stratified sampling method to the train-test split so that relatively equal representation is present in both.

Logistic Regression

We will be having a 80:20 split for train:test.

Hyper parameter tuning

We will use the Grid Search method.

```
LogisticRegression
LogisticRegression(class_weight='balanced', max_iter=1000, solver='newton-cg')
```

We fit the Logic Regression model with the given data with ‘responded’ as the Target variable.

Model Evaluation

We are using a combination of metrics to evaluate the model.

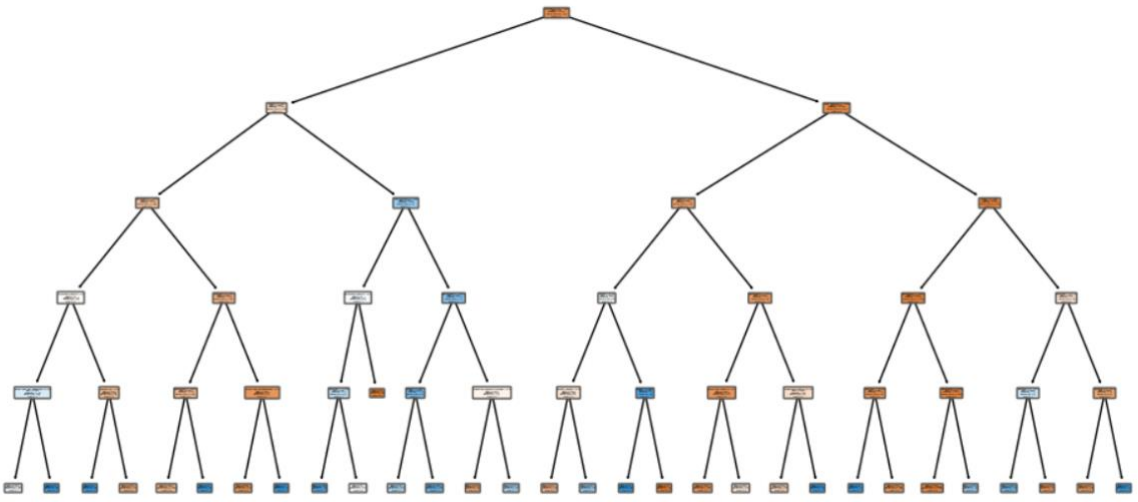
f1 score: 0.43170320404721757

roc-auc score: 0.7486687848432696

accuracy score: 0.7942961165048543

Decision Tree

We transformed the data as required and fit the data with Decision Tree Classifier



We found the optimum results with max_depth = 5.

Model Evaluation

We are using a combination of metrics to evaluate the model.

f1 score: 0.3924528301886792

roc-auc score: 0.7844553050027212

accuracy score: 0.9023058252427184

Random Forest

We fit the data with the Random Forest Classifier method. We found the optimum result with max_depth = 6 and n_estimators = 10.

Model Evaluation

We are using a combination of metrics to evaluate the model.

f1 score: 0.38554216867469876

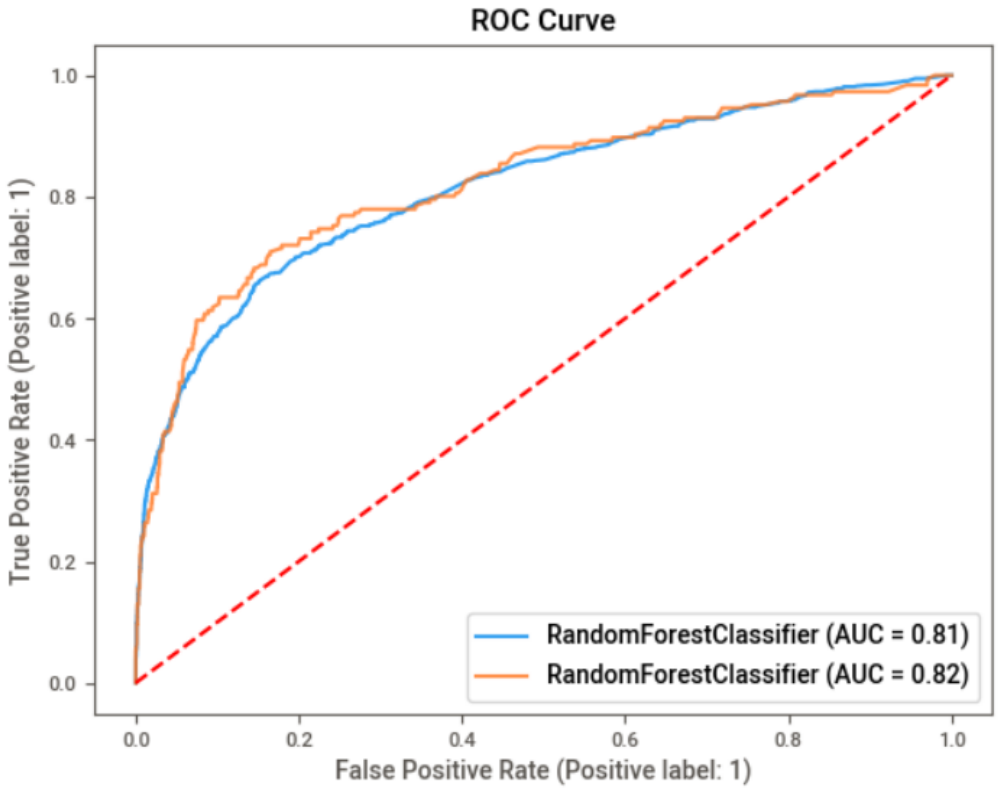
roc-auc score: 0.8236158304281953

accuracy score: 0.9071601941747572

Below is the classification report for the model

	precision	recall	f1-score	support
0	0.91	0.99	0.95	1462
1	0.76	0.26	0.39	186
accuracy			0.91	1648
macro avg	0.84	0.62	0.67	1648
weighted avg	0.90	0.91	0.89	1648

The ROC Curve of the model is shown below:



Feature Importance

	Varname	Imp
1	pdays	0.235929
5	euribor3m	0.192267
3	cons.price.idx	0.106355
2	emp.var.rate	0.105295
4	cons.conf.idx	0.083470
0	custAge	0.055344
6	nr.employed	0.040140
38	poutcome_failure	0.031986
7	pastEmail	0.029226

The top 10 features that have maximum impact in the target feature are shown above.

XGBoost

As part of boosting tree classifier, we tried the XGBoost model.

Model Evaluation

With cross validation, we did with num_boost_round = 500 and nfold=5. We found the below result

	train-auc-mean	train-auc-std	test-auc-mean	test-auc-std
0	0.790057	0.004446	0.761043	0.023991
1	0.809401	0.004247	0.771927	0.021237
2	0.821034	0.008530	0.771352	0.022022
3	0.827933	0.005954	0.773076	0.019324
4	0.833436	0.006803	0.772805	0.021613
...	...	...	...	...
495	0.996463	0.000327	0.714790	0.013982
496	0.996465	0.000350	0.714723	0.013850
497	0.996484	0.000332	0.714570	0.013852
498	0.996508	0.000352	0.714541	0.013845
499	0.996519	0.000346	0.714651	0.013674

f1 score: 0.38554216867469876

roc-auc score: 0.7679971463454098

accuracy score: 0.9047330097087378

Findings

From the above results from the model evaluation metrics, we can see that the Random Forest Classification is the best model based on the data given to us.

Hance, we will be using this model to calculate the propensity score from the potential customers data (‘test.csv’).

We used the same model fitting parameters as trained with the input data and create a new column in the customers data which gives the probability score. In this project, we assign the value ‘1’ for those that have a score or 0.5 and higher and ‘0’ for others. 1 denotes that the customer is highly likely to buy our product and 0 is for an unlikely scenario.

Model Deployment Plan

Amazon **SageMaker Batch Transform** is a **serverless**, **scalable**, and **cost-effective** solution for running batch inferences on **large datasets**. It allows users to perform bulk inferences on their data in the form of a CSV or JSON file, by running inference on a trained SageMaker model. It runs on Amazon EC2 instances, making use of parallel processing to perform inferences quickly and efficiently. The results of the batch inference are stored in an Amazon S3 bucket, providing an easy way to access the results of the inferences.

Here, in our project, we are assigned the task of building a propensity model to identify potential customers to develop campaign strategies for pitching insurance products. Batch Transform is a suitable deployment plan as it performs bulk inferences on the data making use of parallel processing. It is more efficient than real-time interface in this case. It saves cost.

Summary

In this study, we have demonstrated that a data-driven solution can be used to find potential customers for the company. Our study extensively uses the customer data that is with them to find insights and use that data as base to predict whether a given scenario might result in a likely result or otherwise. A strategy developed from this study, can be highly cost efficient as it can find scenarios where the customer might be unlikely to buy their product and we can effectively cut down those campaigns. With such a study, companies can effectively develop strategies and can increase their revenue