BUAN 6341.002 Applied machine Learning Project

Customer Targeting Model

by

Brijesh Dungrani
Jeevesh Dhingra,
Madhur Mehta
Rashi Jain,
Shrey Patel.

Abstract

The purpose of this paper is to target top customers of a super server company for marketing campaign and identify a shortlist of B2B customers who have likelihood to buy products after being subjected to marketing campaign

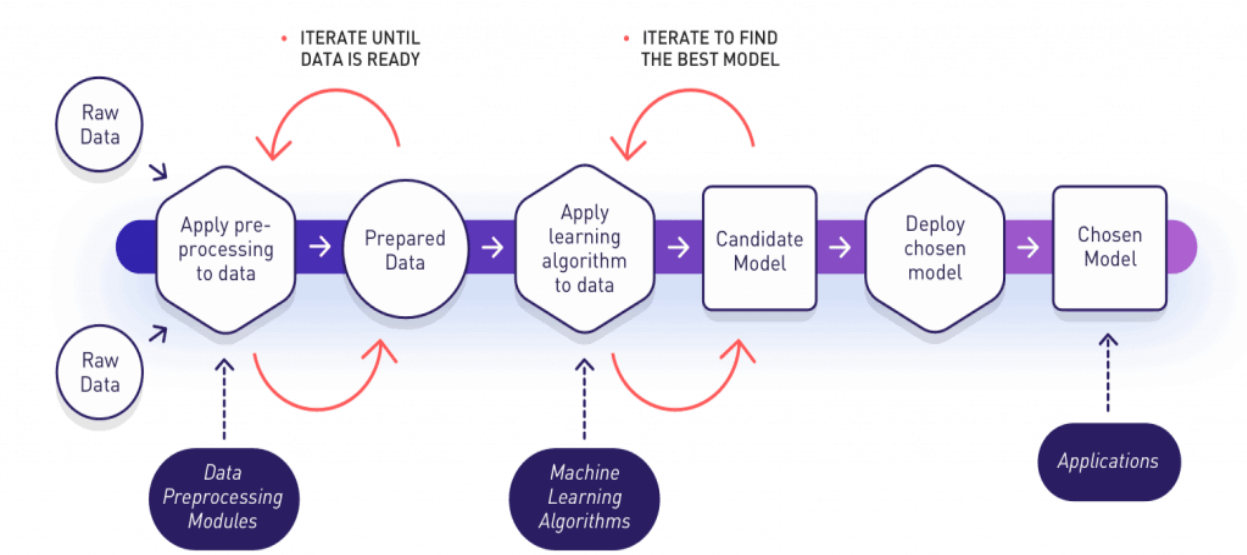
Segmentation strategy helps in capitalizing limited budget for only those customers who have a high likelihood for purchase.

The dataset contains labelled data of customers for that we are using supervised learning methods for model training. We are using classification algorithm Gradient Boosting to train the model. This approach is implemented by using gradient boosting, a supervised machine learning algorithm.

Introduction

Machine learning focuses on the development of computer programs that can access data and use it learn and improve from experience.

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E



Machine Learning Workflow

Now a days marketing has become an integral part of a company's activities for looking ways to promote their new products and services focused on customers. Every company has a limited budget for marketing campaign. A better strategy to effectively use the budget is by targeting customers who have a higher probability of buying a product or service. In the dataset we used, there are 46332 records and 34 features

Methods

Tools Used: Python, Jupyter Notebook, MS Excel

ML Algorithms: Gradient Boosting Classifier.

The reason of using machine learning is to build a predictive model to produce the better customer prediction for marketing campaign. As there is imbalance in data, we have used random under sampling to balance it. Out of 34 features we have used 28 features for model training.

Data Overview

The dataset consists of 46332 records and 34 features.

Broadly, the features tell us about the:

- Contact Attributes (Details about the person in contact)
- Digital Interaction (Communication between client and companies)
- Firmographics (Additional information about the companies)
- IT Spend (Past deals)
- Past Purchases (Whether the companies bought from us)

Featurs:

Variable Category	Variable Name	Description
ID	account_id	Account ID: identifier for a B2B customer
Dependent Variable	purchase_event	whether the account made a purchase overall from "the company" or not
Contact Attributes	decision_maker	whether there is a decision-making contact associated with the account in the known contact database or not
Contact Attributes	number_contacts	the number of contacts associated with the account in the known contact database
Contact Attributes	persona_executive	whether there is an executive contact associated with the account in the known contact database or not
Contact Attributes	persona_manager	whether there is a a manager contact associated with the account in the known contact database or not
Contact Attributes	persona_tech	whether there is a technical contact associated with the account in the known contact database or not
Digital Interaction	email_acitivity	the overall index of email activities
Digital Interaction	event_attendance	whether the account attended some events hosted by the company or not
Digital Interaction	chat_activity	whether the account had some chat activities overall with the company or not
Digital Interaction	web_acitivity_cloud	whether the account had some web acitivities related to the cloud products or not
Digital Interaction	web_activity_networking	whether the account had some web acitivities related to the networking products or not
Digital Interaction	paid_media_activity	whether the account had some paid media activities or not
Firmographics	decision_headquarter	whether the account is a decision headquarter or not
Firmographics	decision_power	whether the account has a high or low decision power
Firmographics	industry_vertical_retail	whether the account belongs to the retail industry or not

Firmographics	industry_vertical_healthcare	whether the account belongs to the healthcare industry or not
Firmographics	industry_vertical_finance	whether the account belongs to the finance industry or not
Firmographics	industry_vertical_infrastructure	whether the account belongs to the infrastructure industry or not
Firmographics	number_employees_1000_10K	whether the account has the number of employees between 1000 and 10K
Firmographics	number_employees_greater_than_50K	whether the account has the number of employees greater than 50K
Firmographics	number_employees_10K_50K	whether the account has the number of employees between 10K and 50K
Firmographics	number_employess_less_than 1000	whether the account has the number of employees less than 1000
IT Spend	it_spend_networking	the projected budget of the account to spend on the networking products
IT Spend	it_spend_cloud	the projected budget of the account to spend on the cloud products
IT Spend	it_spend_others	the project budget of the account to spend on the other IT products
Past Purchases	competitor_1_cloud	whether the account purchased a cloud product from Competitor 1 or not
Past Purchases	competitor_1_networking	whether the account purchased a networking product from Competitor 1 or not
Past Purchases	competitor_3_networking	whether the account purchased a networking product from Competitor 3 or not
Past Purchases	competitor_2_networking	whether the account purchased a networking product from Competitor 2 or not
Past Purchases	competitor_2_cloud	whether the account purchased a cloud product from Competitor 2 or not

```
#Looking at data types
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46332 entries, 0 to 46331
Data columns (total 34 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   account_id                            46332 non-null  int64  
 1   number_contacts                       46332 non-null  int64  
 2   it_spend_networking                   46332 non-null  float64
 3   decision_maker                        46332 non-null  int64  
 4   email_acitivity                       46332 non-null  int64  
 5   it_spend_cloud                        46332 non-null  int64  
 6   it_spend_others                       46332 non-null  int64  
 7   past_purchase_networking              46332 non-null  int64  
 8   number_employees_10K_50K              46332 non-null  int64  
 9   web_acitivity_cloud                   46332 non-null  int64  
 10  industry_vertical_infrastructure      46332 non-null  int64  
 11  decision_headquarter                  46332 non-null  int64  
 12  number_employees_1000_10K             46332 non-null  int64  
 13  industry_vertical_finance             46332 non-null  int64  
 14  competitor_2_networking               46332 non-null  int64  
 15  number_employess_less_than 1000       46332 non-null  int64  
 16  number_employees_greater_than_50K     46332 non-null  int64  
 17  decision_power                        46332 non-null  int64  
 18  chat_activity                         46332 non-null  int64  
 19  web_activity_networking               46332 non-null  int64  
 20  competitor_3_cloud                    46332 non-null  int64  
 21  past_purchase_cloud                   46332 non-null  int64  
 22  event_attendance                      46332 non-null  int64  
 23  competitor_2_cloud                    46332 non-null  int64  
 24  competitor_1_cloud                    46332 non-null  int64  
 25  competitor_1_networking               46332 non-null  int64  
 26  competitor_3_networking               46332 non-null  int64  
 27  industry_vertical_retail              46332 non-null  int64  
 28  paid_media_activity                   46332 non-null  int64  
 29  industry_vertical_healthcare          46332 non-null  int64  
 30  persona_executive                     46332 non-null  int64  
 31  persona_manager                       46332 non-null  int64  
 32  purchase_event                        46332 non-null  int64  
 33  persona_tech                          45863 non-null  float64
dtypes: float64(2), int64(32)
memory usage: 12.0 MB
```

Data challenges

1. Missing Values:

- 469 missing values were found in persona_tech predictor.
- In the number of employees and industry vertical information features, nearly 80% of the records are empty.

Handling Missing Values

- We found that there are 469 null values from column 'persona tech'. We imputed them using knn imputer.
- For the other 2 features, the percentage of missing values are quite high and did not provide much information hence, these features were also dropped

2. Outliers:

An outlier is a data point that is noticeably different from the rest. They represent errors in measurement, bad data collection, or simply show variables not considered when collecting the data.

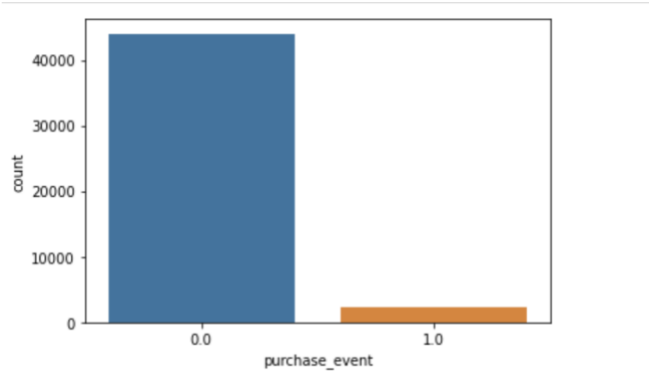
Handling Outliers

Outliers comprise of a very minute percentage of data, and we can see that these are extreme values hence it is better to get rid of them.

3. Imbalanced dataset:

Imbalanced data refers to those types of datasets where the target class has an uneven distribution of observations, i.e., one class label has a very high number of observations and the other has a very low number of observations.

We observed that 96% of the records belong to non-responders whereas only 4 % of the records belong to the responders.



Handling Imbalanced dataset:
An effective way to handle imbalanced data is to oversampling or undersampling. Let's start by defining those two new terms:

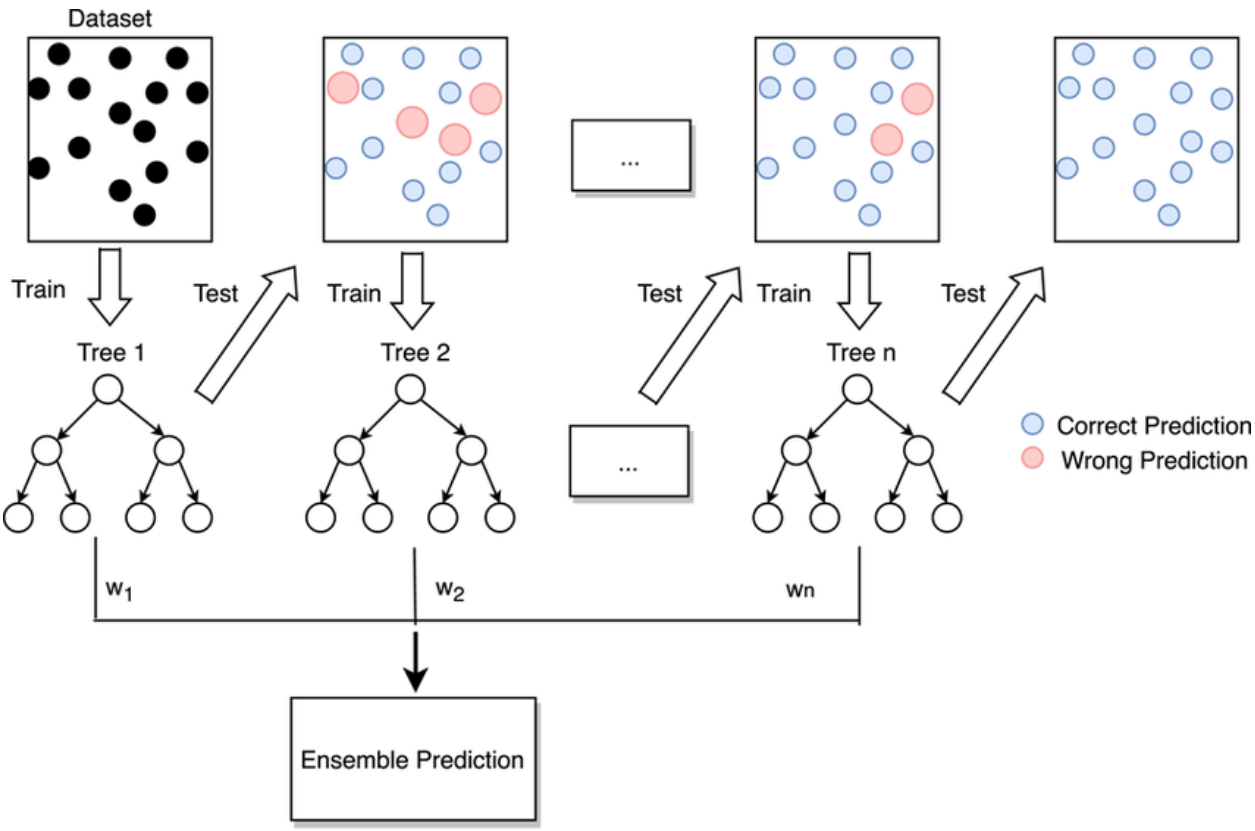
- **Random Oversampling**: Randomly duplicate examples in the minority class.
- **Random Undersampling**: Randomly delete examples in the majority class.

Algorithms and techniques

We used PyCaret library for this project.

PyCaret is an open-source, low-code machine learning library in Python that automates machine learning workflows. It is an end-to-end machine learning and model management tool that speeds up the experiment cycle exponentially and makes you more productive. In comparison with the other open-source machine learning libraries, PyCaret is an alternate low-code library that can be used to replace hundreds of lines of code with few words only. This makes experiments exponentially fast and efficient.

We decided on using Gradient Boosting Classifier for this project. GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n_classes_ regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced.



Flow diagram of GB

```
from sklearn.ensemble import GradientBoostingClassifier
clf = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0,
    max_depth=1, random_state=0).fit(X_train_new, Y_train)
clf.score(X_test_new, Y_test)
```

```
0.9569064748201439
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn import metrics
from numpy import mean
from imblearn.ensemble import BalancedBaggingClassifier
from imblearn.ensemble import EasyEnsembleClassifier
```

```
# init setup
from pycaret.classification import *
clf1 = setup(data = df_upd, target = 'purchase_event')

# compare models
best = compare_models()
```

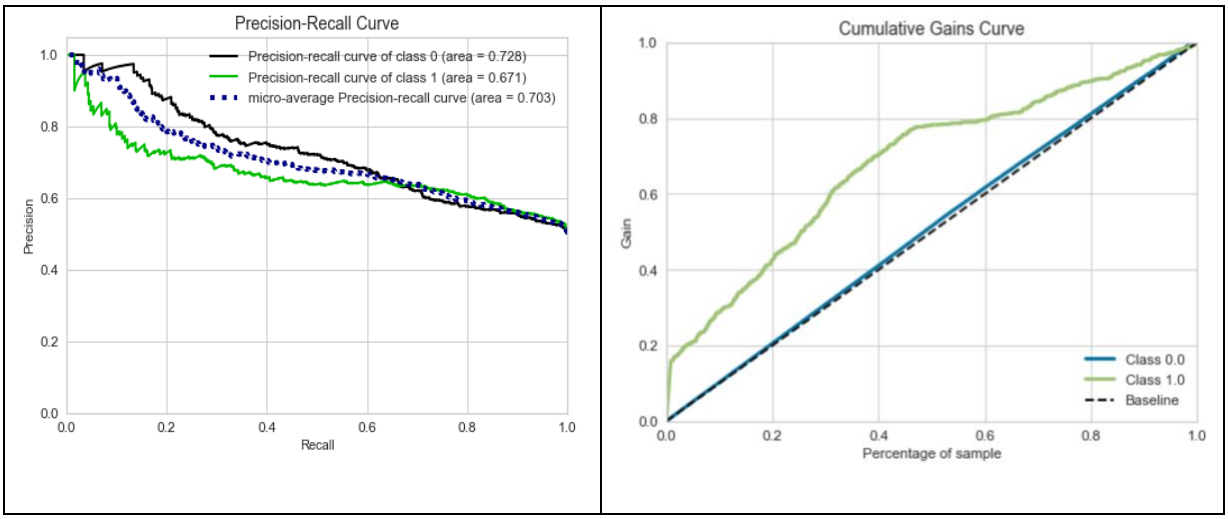
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ada	Ada Boost Classifier	0.9574	0.7106	0.1749	0.9914	0.2968	0.2858	0.4064	0.8090
lightgbm	Light Gradient Boosting Machine	0.9574	0.7412	0.1749	0.9868	0.2966	0.2856	0.4055	0.4630
gbc	Gradient Boosting Classifier	0.9571	0.7353	0.1754	0.9554	0.2960	0.2846	0.3990	2.4720
rf	Random Forest Classifier	0.9530	0.7004	0.1934	0.6486	0.2973	0.2802	0.3368	1.8880
et	Extra Trees Classifier	0.9517	0.6903	0.2198	0.5827	0.3188	0.2990	0.3380	0.9950
nb	Naive Bayes	0.9485	0.5163	0.0000	0.0000	0.0000	0.0000	0.0000	0.0350
ridge	Ridge Classifier	0.9485	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0360
qda	Quadratic Discriminant Analysis	0.9485	0.6194	0.0000	0.0000	0.0000	0.0000	0.0000	0.0540
lda	Linear Discriminant Analysis	0.9485	0.5611	0.0000	0.0000	0.0000	0.0000	0.0000	0.0420
dummy	Dummy Classifier	0.9485	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0260
knn	K Neighbors Classifier	0.9451	0.5839	0.0180	0.1745	0.0325	0.0232	0.0406	0.2590
lr	Logistic Regression	0.9416	0.5130	0.0108	0.0794	0.0188	0.0051	0.0091	1.1580
dt	Decision Tree Classifier	0.9197	0.6114	0.2677	0.2448	0.2555	0.2132	0.2135	0.1830
svm	SVM - Linear Kernel	0.9027	0.0000	0.0599	0.0114	0.0187	0.0022	0.0039	0.3750

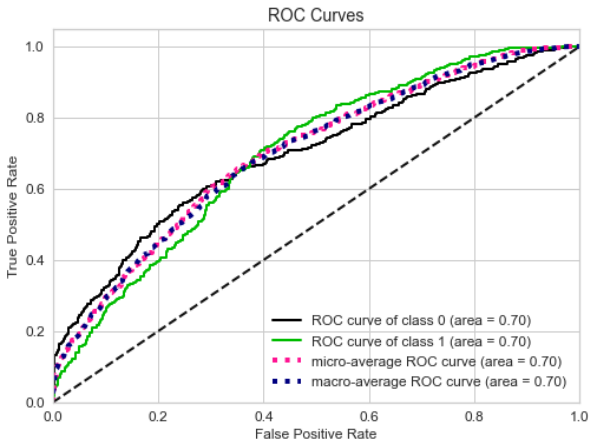
Results

The model has a roc_auc score of 70 and f1score of 67%.

66% of the buyers have a technical contact associated with the account in the known contact database.

73% of buyers are decision headquarters.





99% of the buyers are accounts with low decision power. Area under precision recall curve should be as high as possible as it is a trade-off between both quantities. Gain chart shows that how many times positive class will be selected more than the negative class in top samples.

As we know that AUC is the probability that the model ranks a random positive example more highly than a random negative example, it should be as high as possible, in our case it is 0.7.

Conclusions

- **Missing Values:**
    - 469 missing values were found in **persona_tech** column.As these rows account for only 1% of the data they were dropped.
    - For 80% of the records the **number of employees** information is missing and for 84% of the records **industry vertical information** is missing. As the percentages are quite high and we did not have enough information to impute these values , these columns were also dropped.
- **Outliers:** There were 3 potential outliers each from **number_contacts,it_spend_networking and it_spend_others columns**.These outliers were dropped from data to make model more robust.
- **Imbalance :** The ratio of total records for 2 classes are 18:1 which causes imbalance in dataset,so we have used under sampling by taking a sample from majority class that has comparable number of records as in the minority class.
- **Recommendations.**
    - 91% of buyers didn't show up for any events hosted by the company
    - 96% of buyers didn't engage in any chat activities with the company.

Based on above points, there is a need to improve digital interactions with existing customers.

References

- https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
- https://scikit-plot.readthedocs.io/en/stable/metrics.html
- https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/
- https://pycaret.gitbook.io/docs/