Supervised Learning for detecting potential clients

[EX1] Let's identify the type of the variables (integer, float, chart...) and the size of the dataset and the file. Which are the variables with more nulls? And with no nulls? Why 'City' variables is considered as object type?

Variable	Type	Number of nulls	Number of non-nulls
City	object	0	13335
Customer Flag	integer	0	13335
Revenue	float	4746	8589
Sector	float	100	13235
Legal_Form_Code	float	106	13229
CNT_EMPLOYEE	integer	0	13335
CNT_CB_DENSITY	float	3070	10265
CNT_CB_MOB_DENSITY	float	3070	10265
CNT_CB_FN_DENSITY	float	3070	10265
Mobile_potential	float	0	13335

Data size: 10 columns and 13335 rows File size: 1 MB

The variable with more nulls is Revenue, which has 4746 nulls, and variables City, Customer Flag, CNT_EMPLOYEE, and Mobile_potential are all filled, they do not have nulls.

City variable is considered as an object type because in pandas it is the default dtype to store strings and represent non-numerical variables. This dtype also accepts a mix of some other python variables.

[EX3] Create a customer_dtand noncustomer_dtdatasets based on the Customer_Flagvariable (Customer_Flag=1 for customers and 0 for noncustomers). Build a boxplot for the Revenue, CNT_EMPLOYEE, Mobile_potential and CNT_CB_Density numeric variables for both datasets. Which are the main differences between customer_dtand noncustomer_dt datasets comparing these variables. Which is the dataset with CNT_EMPLOYEE higher? Which datasets have more outliers in Revenues? Which is the Q1, median (=Q2) and Q3 for Revenues and Mobile_potential?

COSTUMER

- Revenue: Q1 = 1047500.0 Q2 = 2200000.0 Q3 = 4195000.0
- Mobile_potential: Q1 = 1621.055686 Q2 = 1948.437661 Q3 = 2116.474074
- CNT_EMPLOYEE: Q1 = 14 Q2 = 21 Q3 = 31
- CNT_CB_DENSITY: Q1 = 71.5 Q2 = 203 Q3 = 519

NON COSTUMER

- Revenue: Q1 = 902986.0 Q2 =1750000.0 Q3 =3501123.5
- Mobile_potential: Q1 =1513.383597 Q2 = 1797.054278 Q3 = 2035.082840
- CNT_EMPLOYEE: Q1 = 12 Q2 = 18 Q3 = 27
- CNT_CB_DENSITY: Q1 = 37 Q2 = 113 Q3 = 407.25

Although both datasets have the same maximum value of CNT_EMPLOYEE (50) the customer data set tends to have higher values: the three quartiles are higher in customer, which is the highest difference in Q3. Also observe that in the noncustomer, the highest value is interpreted as an outlier while in the customer dataset not (this is not a strong proof but gives some intuition).
Following the 1.5 IQR rule we got 95 outliers for the customer outliers vs 569 outliers in the non-customer data set. [See boxplots below]

[EX4] Remove the values higher than 5*IQR for each variable: Revenue, CNT_EMPLOYEE, Mobile_potential, CNT_CB_DENSITY, CNT_CB_MOB_DENSITYand CNT_CB_FN_DENSITY. Compare with the boxplot of the previous exercise [EX3] and explain the differences.

As we can observe, we have removed some of the outliers, hence, now we can better visualize the boxplot of the different variables. [See boxplots below]

[EX5] Calculate the ratio of the values of City for customer_dt and noncustomer_dt datasets. Compare the ratio of each category of each dataset.

For each city its ratio is the proportion of samples that belongs to that city over all samples. Comparing both datasets, we observe the ratios are quite different.

Ratio of City values for customers		Ratio of City values for noncustomers	
München	0.025060	Köln	0.016139
Köln	0.020286	Stuttgart	0.010139
Chemnitz	0.019093	Bremen	0.010139
Frankfurt	0.014320	Dortmund	0.009518
Stuttgart	0.013126	Düsseldorf	0.008897
...		...	
Kernen	0.001193	Nußdorf	0.000207
Nordkirchen	0.001193	Altlußheim	0.000207
Grasellenbach	0.001193	Glaubitz	0.000207
Thum	0.001193	Erwitte	0.000207
Reichenberg	0.001193	Saulheim	0.000207

[EX6] Calculate the length of X_train and X_test datasets. Is it aligned with the test_size value selected in the split?
X_train has 4536 rows and 9 columns, and X_test has 1135 rows and 9 columns. The test_size is 25% of the data selected in the split, so indeed the sizes are aligned.

[EX7] Draw the histograms of y_train and y_test. Is the dataset balanced (similar number of rows for each class or Target) or imbalanced? How do you think it could affect the quality of the classifier?

We have more data labeled as 0 in the target and they keep the same proportion in y_train and y_test. Therefore, the datasets are imbalanced. This will cause the probability of belonging to class 0 to be much higher than class 1 since if the algorithm had any doubt between a class the most probable class will be 0, leading to a biased classification. [See histograms below]

[EX8] Train the SVM classifier of the Sklearn library. Evaluate the following metrics for kernel="rbf":

Which is precision, recall and accuracy of the algorithm?

- 0: precision = 0.83 recall: 1.00
- 1: precision = 1.00 recall: 0.01
- accuracy: 0.83

Which is the confusion matrix?

	predicted 0s	predicted 1s
actual 0s	945	0
actual 1s	189	1

Is the algorithm classifying both classes (Target=0 and Target=1) in a similar way? Why?

No, observing the confusion matrix we have that 1585(945+189) samples have been classified as 0's, whereas we only have one 1. Notice that the real number of 0's is correct but the number of 1's is 190. This is due to the unbalance in the set: it learned to almost always predict 0 so the accuracy is great, and also the precision of both classes since the predictions are mostly correct. The precision of 0's is high since we have more 0's than 1's so even predicting almost all 1's as 0's the precision rate is still good. Since we only predict 1's in extreme cases all predicted 1's are correct so we have a precision of 1. Nevertheless the recall of 0's is almost 0 which means that it only captures 1% of the 1's.

Compare the precision and recall of the training dataset (i.e.X_train and y_train) vs the test dataset (i.e. X_test and y_test). Which is better? Why? Which is the correct measure to estimate the performance for other unseen datasets? Why?
Evaluating the model in the training set we got:

- 0: precision = 0.86 recall: 1 1: precision = 1 recall: 0.01
- Accuracy: 0.86

The model is not good even for predicting trained samples, so we would say that both predictions are equally wrong but slightly better in the train set.
We cannot evaluate the performance with the training dataset since we have already used it to optimize the algorithm so it will always provide better results (if the difference is high the model could be overfitted). Since the results using the training dataset are quite like the ones using the test dataset our model is not overfitted.

In an ideal case we would look for perfect classifications, but it depends a bit on what we are trying to predict. In our practice case we want to predict targets and since it is expensive to send people to talk with them, we want to be as sure as possible that they are good targets. Although all ratios are somehow dependent, we would focus on a high precision (of 1s) but giving 5000 samples to only predict one target is not efficient (low recall) so we probably could gain a lot of recall with sacrificing precision and make a big improvement in the model.

[EX9] Train the Decision Tree algorithm from Sklearn library. Evaluate the following metrics:

Which is precision, recall and accuracy of the algorithm?

- 0: precision = 0.86 recall = 0.86
- 1: precision = 0.31 recall = 0.31
- accuracy = 0.56137

Which is the confusion matrix?

	predicted 0s	predicted 1s
actual 0s	812	133
actual 1s	129	61

Is the Decision Tree algorithm working better than SVM? Why?

In order to compare the algorithms, we want to capture the maximum number of new customers so we are interested in the recall of 0s(non-customers), also we have to take care of the cost of sending sales managers, so the precision of 0s and the recall of 1s are important too.
Notice that the Decision Tree model has higher precision of non-customers whereas the recall of non-customers has decreased. Nevertheless, the recall of 1's is much higher now which compensates the lower recall of 0s. Therefore, Decision Tree is the best option for our problem.

[EX10] Train SVM and Decision Tree algorithms using a KFold cross-validation with k=5 and calculate the mean and standard deviation of the accuracy. Plot a boxplot of the accuracy for every model. Which is the model with better mean value of the accuracy? Which is the algorithm with less deviation on the accuracy?

Accuracy SVM: mean = 0.8589 std dev = 0.0125
Accuracy Decision trees: mean = 0.7804 std dev = 0.0088
SVM has a better accuracy mean and Decision trees have less deviation on the accuracy since its standard deviation is smaller. [See boxplot below]

[EX12] Train an SVC and Decision Tree algorithm with the new final_dataset. Evaluate the recall, precision, and confusion matrix of all models.

SVC

- 0: precision=0.52 recall=0.70
- 1: precision=0.52 recall=0.34
- Accuracy: 0.52

- Confusion Matrix:

	predicted 0s	predicted 1s
actual 0s	196	85
actual 1s	180	93

Decision tree

- 0: precision=0.61 recall=0.54
- 1: precision=0.58 recall=0.65
- Accuracy: 0.57

- Confusion Matrix:

	predicted 0s	predicted 1s
actual 0s	151	130
actual 1s	96	177

Which has better accuracy?
Decision trees have more accuracy than Support Vector Classification.

Which is the model with better recall? And precision?
The SVC model has better recall classifying 0's whereas Decision tree model has better recall classifying 1's. Nevertheless, the SVC model has a small value of recall for class 1, this implies that sales managers visit actual customers wasting our resources. Moreover, Decision tree model has a more balanced recall, in conclusion Decision Tree has better recall.
Comparing the precision of both models it is clear that Decision Tree has higher precision in both classes.
If we compare these models with the previous Decision Tree algorithm, we observe that the precision of DT is much higher than the actual models and the recall of 1s is higher than the SVC recall. Nevertheless, we are also interested in having a small recall in 0s so the actual DT will have better recall.

Which model do you recommend to classify both classes? Justify your answer.

Overall, since precision is an important factor and also having higher recall in our problem, we consider that the actual Decision tree model will be more efficient.

[EX13] Build a voting ensemble formed by a SVM and Decision Tree and train it with the balanced training dataset. Calculate the precision, recall and confusion matrix of the new classifier. Is it better than any of the previous baseline models? Justify your answer.

- Accuracy: 0.51
- 0: precision=0.50 recall=0.82
- 1: precision=0.55 recall=0.21
- Confusion matrix:

	predicted 0s	predicted 1s
actual 0s	135	30
actual 1s	135	36

The recall of 1s is too small. Compared with the Decision Tree model, since the ensemble model has lower precision and recall in 1s (apart from lower accuracy) we still consider the Decision Tree model as the better option.
Between the ensemble and the SVM, both have a similar accuracy and precision of 0s, but the recall of 1s is considerably better in the SVC than in the ensemble and therefore we would keep the SVM.

[EX14] Build a Bagging ensemble based on Random Forest. Random Forest is considered a bagging ensemble formed by Decision Trees algorithms. Train the Random Forest with the balanced training dataset, i.e., X_train and y_train. Calculate the precision, recall and confusion matrix of the new classifier. Is it better than any of the previous baseline models? Justify your answer.

- Accuracy: 0.61
- 0: precision=0.62 recall=0.58
- 1: precision=0.62 recall=0.65
- Confusion matrix:

	predicted 0s	predicted 1s
actual 0s	96	69
actual 1s	60	111

Those results are clearly the better ones obtained. We have balanced data and in general a higher accuracy, precision and recall than in any other model.

[EX15] Plot the histograms of the probabilities resulting from the prediction of the Random Forest model for class 0 and class 1. [See histogram below]
The probabilities of being customer or non-customer are too overlapped. The probabilities of 1's are slightly shifted to the right in comparison with the probabilities of 0's, which makes that the default cut-off of 0.5.

[EX16] Build a Boosting ensemble based on Gradient Tree Boosting (GBT). There are several boosting algorithms such as Adaboost, etc. Train the GBT with the balanced training dataset, i.e., X_train and y_train. Calculate the precision, recall and confusion matrix of the new classifier. Is it better than any of the previous baseline models? Justify your answer.

- Accuracy: 0.61
- 0: precision=0.62 recall=0.52
- 1: precision=0.60 recall=0.69
- Confusion matrix:

	predicted 0s	predicted 1s
actual 0s	83	79
actual 1s	53	118

This model is better than the SVC and DT. First, the overall accuracy is better. Also, it has a high precision for 0's and higher recall for 1's. It would be comparable to the bagging ensemble model we did before.

[EX17] Plot the histograms of the probabilities resulting from the prediction of the GBT model for class 0 and class 1. Compare it with histogram of Random Forest. Which one classifies better from your point of view? Why?

The GBT model seems to separate in a more efficient way the probabilities for classes 0 and 1. A possible cut-off for GBT will be a value around 0.5 since we have the peak probability of 1 at ~0.65. Random Forest model has more overlapped probabilities and the peaks of the probabilities overlap too, which makes it harder to believe the prediction in some cases since it estimates for example at 0.69(more or less) more wrong targets than correct. These plots are like a visual interpretation of the confusion matrix. [See histogram below]

[EX18] Execute the prediction for the selected model. Adjust the cutoff value to optimize the classifier if you consider necessary. How many non customers are you going to send to the sales managers to sell our products to them?

To optimize the selected model the cut-off value must be between 0.5 or slightly larger than 0.5. With threshold 0.5 we have obtained a suitable precision value (0.62) and a higher recall also (0.52 for 0s and 0.69 for 1s). The result seems to be balanced. Using cut-off 0.5, we would send managers to capture 139 new customers in order to sell our products but only 86 of them are non customers.

[EX19] Order the features by importance. Which are the top 3 features to discriminate between non-customers and customers?

1. CNT_CB_MOB DENSITY: Number of companies with mobile services.
2. Mobile Potential: An estimation of the total annual expense that a company can do in telco services including IoT.
3. City_coded: The numeric code mapped to each city (this seems quite surprising since apparently there is some hidden pattern in the way the cities are mapped, or could be just randomness)

[EX20] In this project, we have used classification techniques to identify potential customers. We have adjusted the main classification parameters as cutoff, recall and precision according to the final purpose: in our case, identify non customers that could be interested in buying our products. Consider a new campaign focused on accelerating the sales of an existing mobile tariff to our customers. Answer the following questions:

In this case, which is the target variable? Which are target=0 samples? And target=1?

The target will distinguish two targets to train the model, 0 and 1. Target 0 corresponds to customers without mobile tariff and target 1 customers with mobile tariff.

As the marketing campaign is oriented to our customers we will have further information about them in our internal systems. In particular, we could add to the information of the previous section 3 new variables: data and voice consumption and mobile expense. Adding more data to the dataset may imply more computational data and cost. Would you add these 3 new variables to the dataset? Justify your answer.

Yes because a priori this new data is much more related to the tariff that a customer would be interested in. If you don't want to imply more computational cost, maybe it is a good idea to drop the 3 least important features of the model and keep this new data which could provide more information.

Today the mobile tariff is not very popular among our customers. Will the training dataset be balanced or unbalanced? Justify your answer.

Unbalanced. As we have seen in this lab if you have an unbalanced data (suppose many 0's) the model could end up predicting 0 in most of the cases since it will be the prediction with highest accuracy due to the unbalance.

Imagine the following scatter plot (monthly data traffic vs monthly voice traffic):

Describe in terms of monthly data traffic and monthly voice traffic the pattern of target 1 customers.

It seems that customers with Mobile Tariff tend to be the ones with the highest consumption both in Monthly voice and data traffic (not only one, but at both features).

Draw a plane to separate both classes. [See picture below]

According to the previous plane, which are the customers to be phoned to sell the mobile tariff?

The ones above the line.

Could you estimate the precision and recall of the classification?

Confusion matrix:

	predicted 0s	predicted 1s
actual 0s	32	4
actual 1s	2	9

Precision:	0: 0.94	1: 0.69
Recall:	0: 0.89	1: 0.82

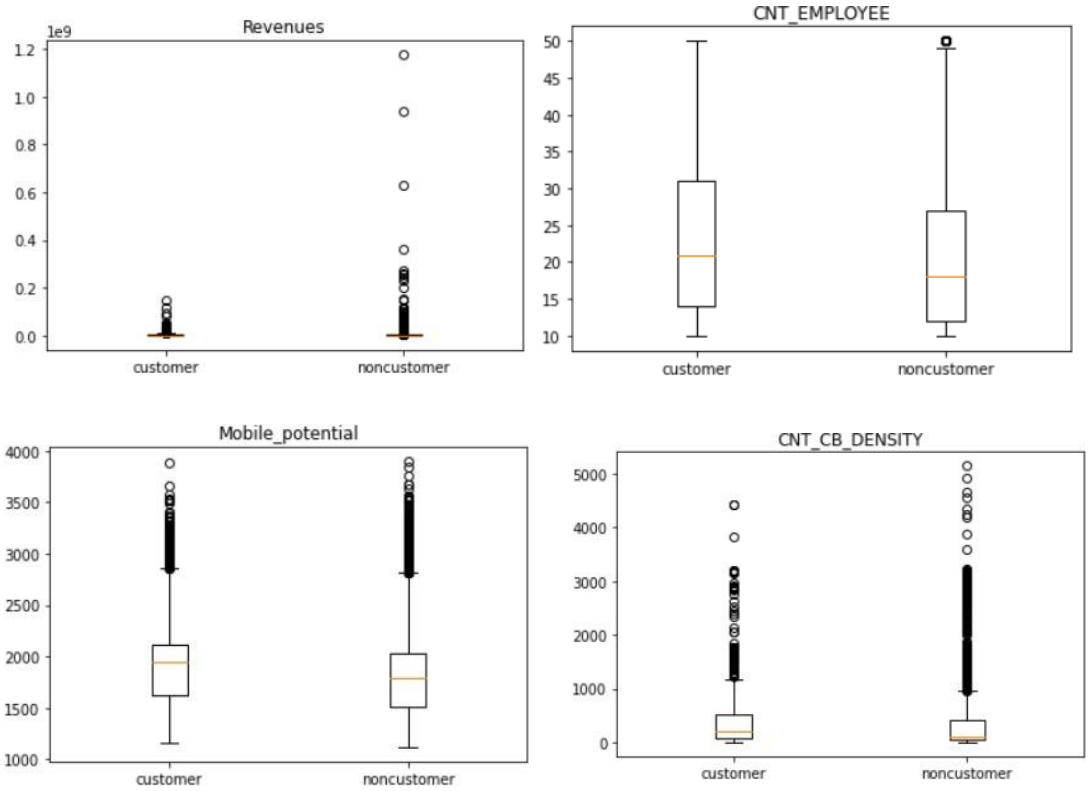
We hereby declare that, except for the code provided by the course instructors, all of our code, report, and figures were produced by ourselves.

Annex

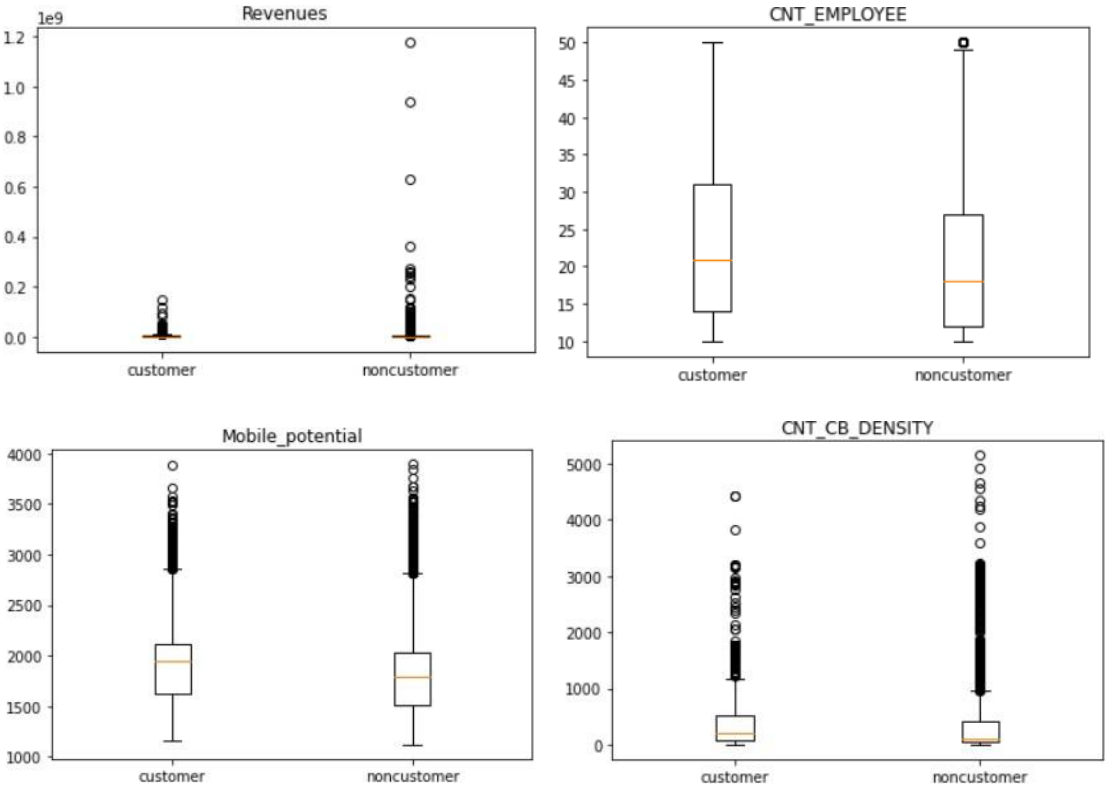
COMPARISON TABLE

		Accuracy	Precision		Recall		Confusion matrix
			0	1	0	1	
Sklearn library	SVM	0.83	0.83	1.00	1.00	0.01	945 0 189 1
	DT	0.56	0.86	0.31	0.86	0.32	812 133 129 61
KFold cross-validation k=5	SVM	mean → 0.86			std dev → 0.0125		
	DT	mean → 0.78			std dev → 0.0088		
Balanced dataset	SVM	0.52	0.52	0.52	0.70	0.34	196 85 180 93
	DT	0.57	0.61	0.58	0.54	0.65	151 130 96 177
Voting ensemble	VC	0.51	0.50	0.55	0.82	0.21	135 30 135 36
Bagging	RT	0.61	0.62	0.62	0.58	0.65	96 69 60 111
Boosting	GTB	0.61	0.62	0.60	0.52	0.69	83 79 53 118

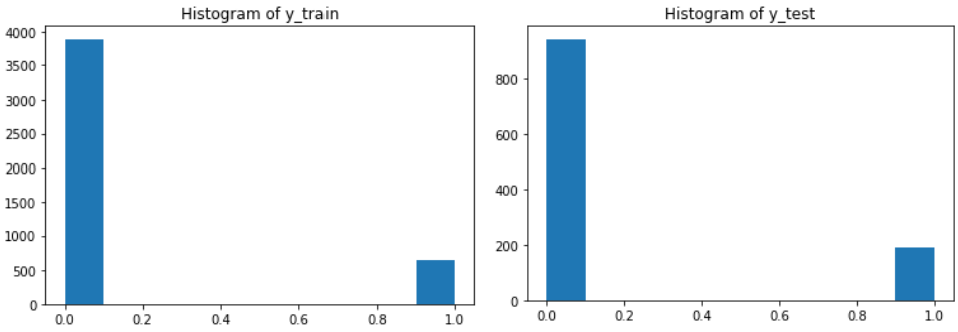
[EX3]


Revenues
1e9
1.2
1.0
0.8
0.6
0.4
0.2
0.0
customer
noncustomer
CNT_EMPLOYEE
50
45
40
35
30
25
20
15
10
customer
noncustomer
Mobile_potential
4000
3500
3000
2500
2000
1500
1000
customer
noncustomer
CNT_CB_DENSITY
5000
4000
3000
2000
1000
0
customer
noncustomer

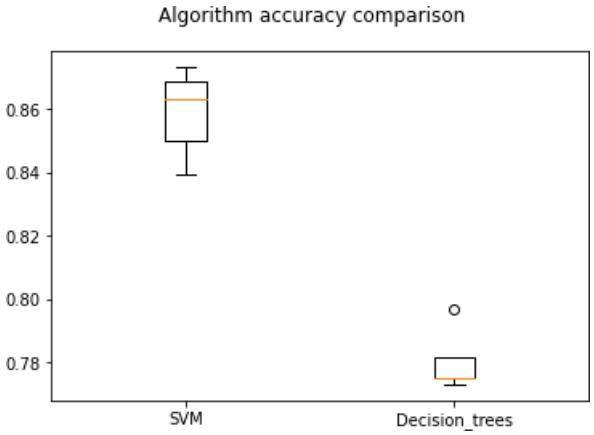
[EX4]


Revenues
1e9
1.2
1.0
0.8
0.6
0.4
0.2
0.0
customer
noncustomer
CNT_EMPLOYEE
50
45
40
35
30
25
20
15
10
customer
noncustomer
Mobile_potential
4000
3500
3000
2500
2000
1500
1000
customer
noncustomer
CNT_CB_DENSITY
5000
4000
3000
2000
1000
0
customer
noncustomer

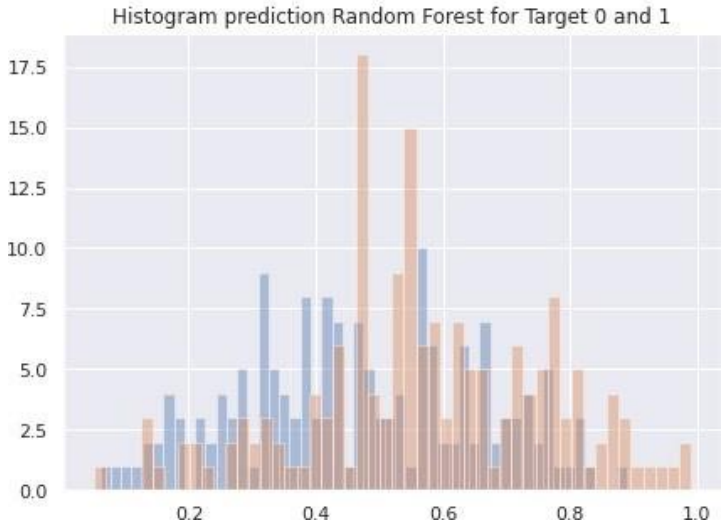
[EX7]



[EX10]

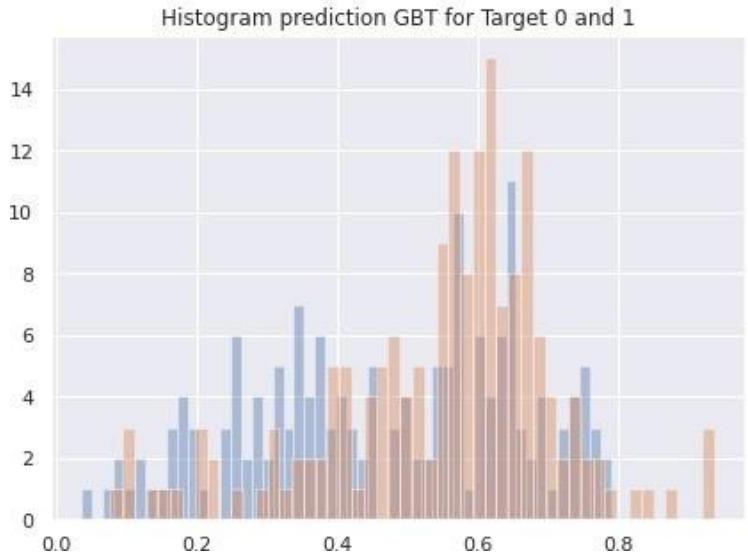


[EX15]



Target 0: Blue
Target 1: Orange

[EX17]



Target 0: Blue
Target 1: Orange

[EX20]