Lead Scoring Assignment

Presentation

Table of Contents

Problem Statement

An education company called X Education wants to improve their lead conversion rate with the help of machine learning model to identify "Hot Leads".

If they successfully identify this set of leads, the lead conversion rate should go up as the sales team will now be focusing more on the potential leads.
c

Analysis approach summary

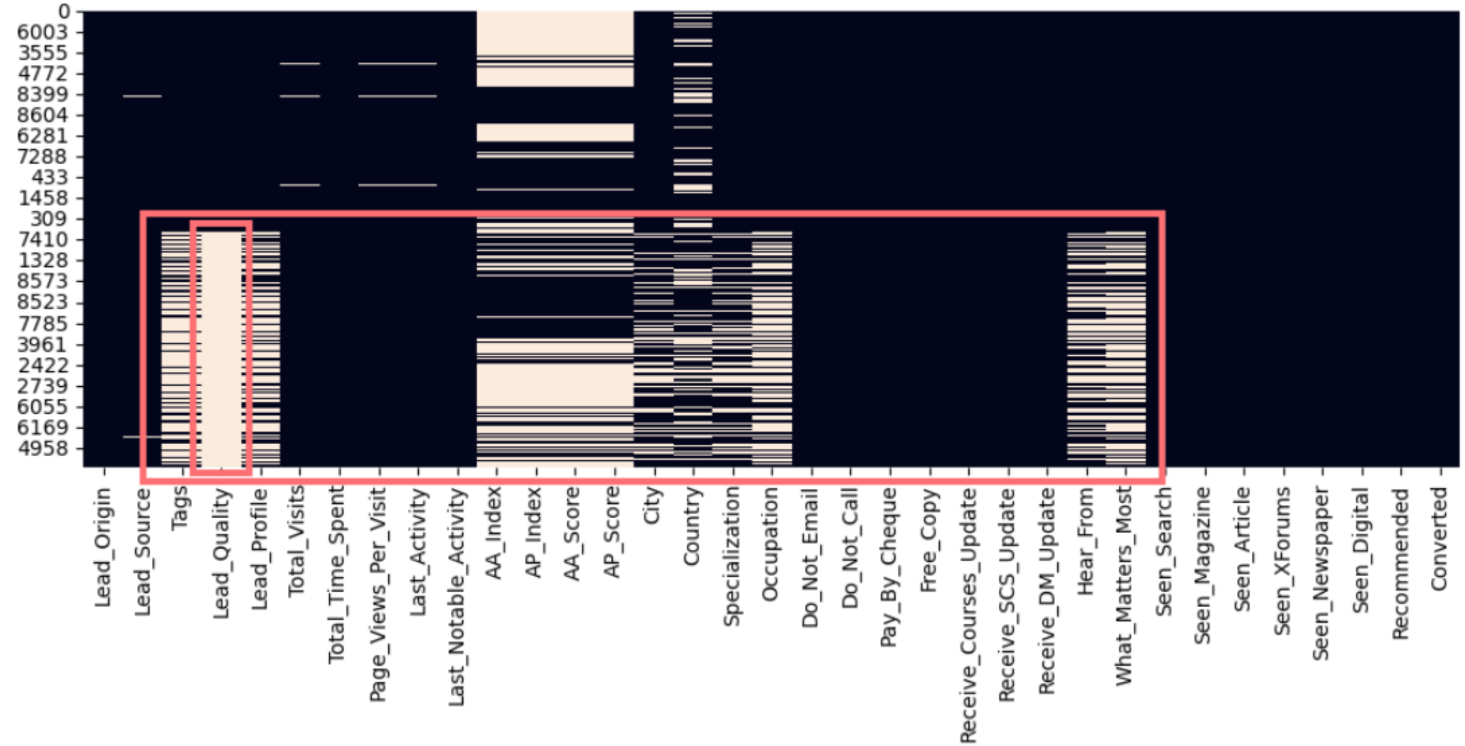
1. DATA WRANGLING
2. EDA
3. DATA PREPARATION
4. FEATURE SELECTION
5. MODEL BUILDING AND FINETUNING
6. MODEL EVALUATION

Data wrangling

Missing values

- 'Select' was replaced with NaN
- Features with high percentage of missing values were dropped
- Missing values were imputed with appropriate values where applicable

Visualize locations of missing values



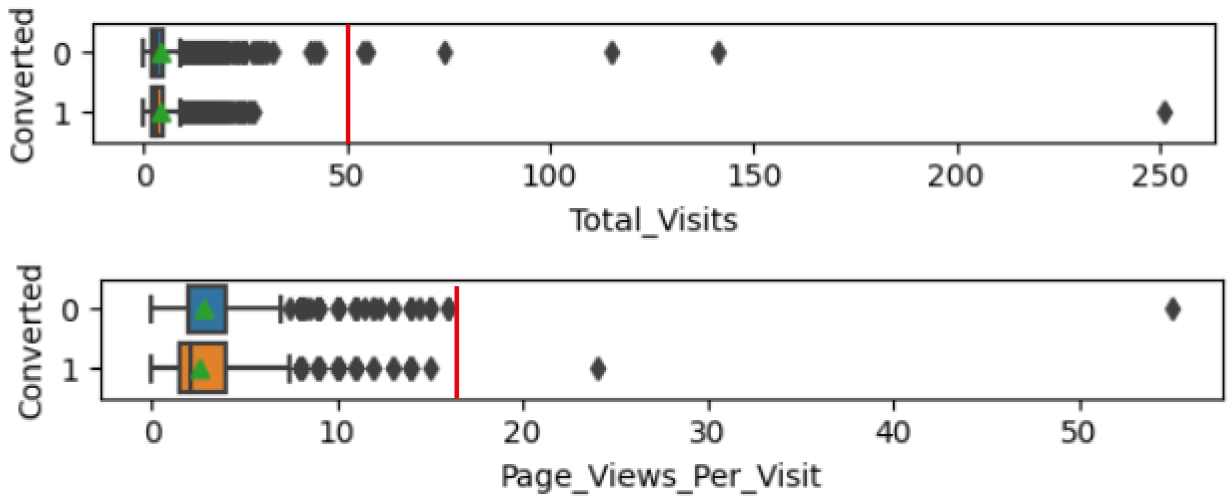
Missing values count

```python
# missing values
df.isna().sum()[df.isna().sum()>0]
```

```
Lead_Source               33
Tags                    2402
Lead_Quality            3705
Lead_Profile            1970
Total_Visits             137
Page_Views_Per_Visit     137
Last_Activity            103
AA_Index                3515
AP_Index                3515
AA_Score                3515
AP_Score                3515
City                     682
Country                 1180
Specialization           699
Occupation              1951
Hear_From               1468
What_Matters_Most       1970
```

Outliers

- Outliers that exceeded a certain limit were capped at a reasonable value.



Other sanity check

- Categorical features that have little variation were removed.
- The category names were spellchecked, shortened or renamed.
- Infrequent classes were grouped into 1 category.

Before

```
df.Lead_Source.value_counts()
```

```
Google               2868
Direct Traffic       2543
Organic Search       1154
Olark Chat            673
Reference             410
Referral Sites        125
Welingak Website       73
Facebook               52
XNA                    33
bing                    6
google                  5
Click2call              4
Press_Release           2
Social Media            2
Live Chat               2
youtubechannel          1
testone                 1
Pay per Click Ads       1
welearnblog_Home        1
WeLearn                 1
blog                    1
NC_EDM                  1
```

After

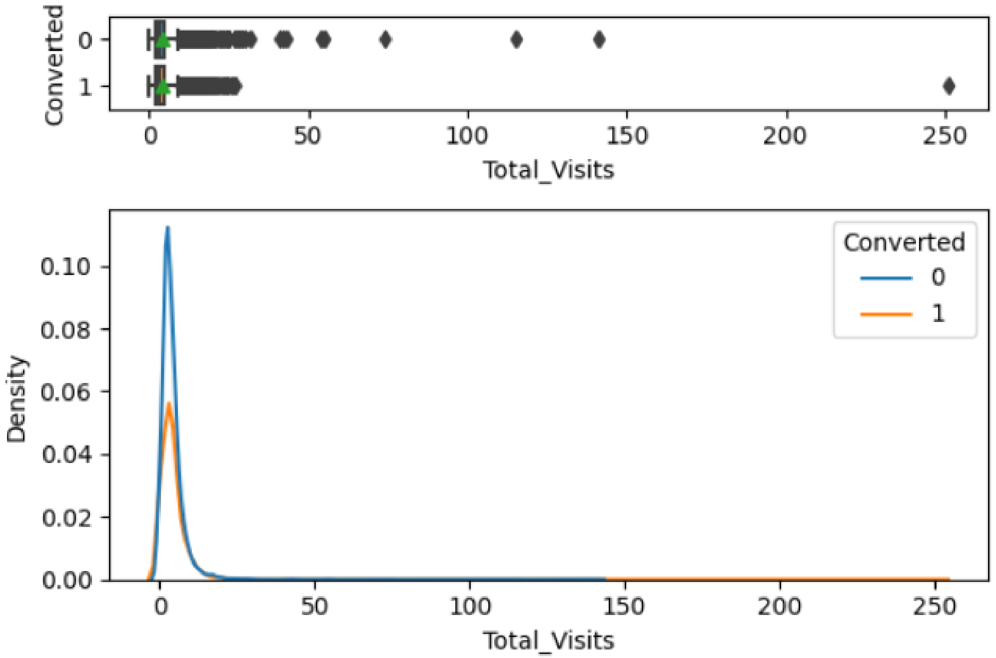
```
df.Lead_Source.value_counts()
```

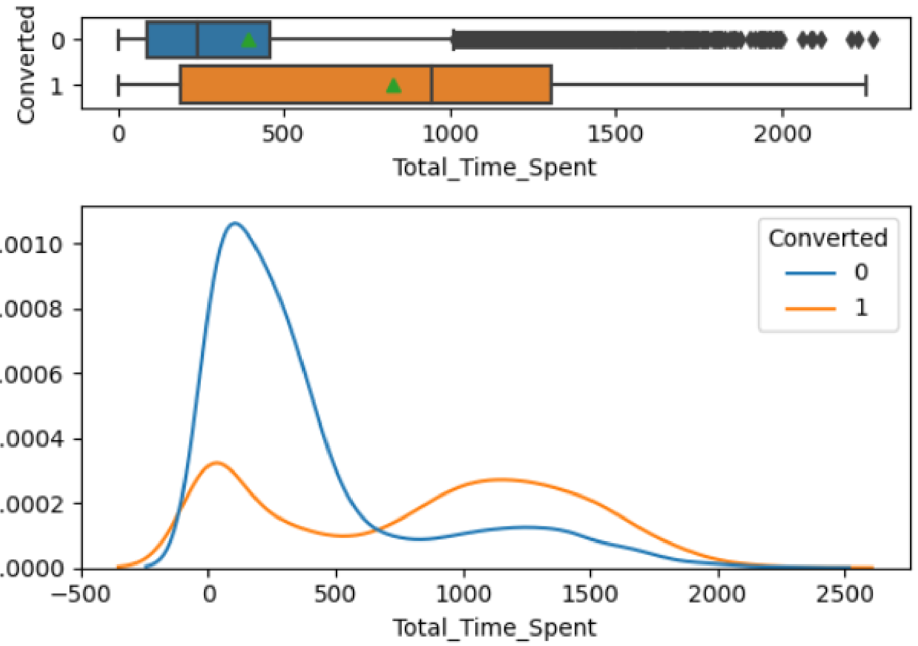
```
Google               2873
Direct Traffic       2543
Organic Search       1154
Olark Chat            673
Reference             410
Referral Sites        125
Welingak Website       73
Social Media           54
XNA                    33
Others                 21
```

EDA

EDA main points

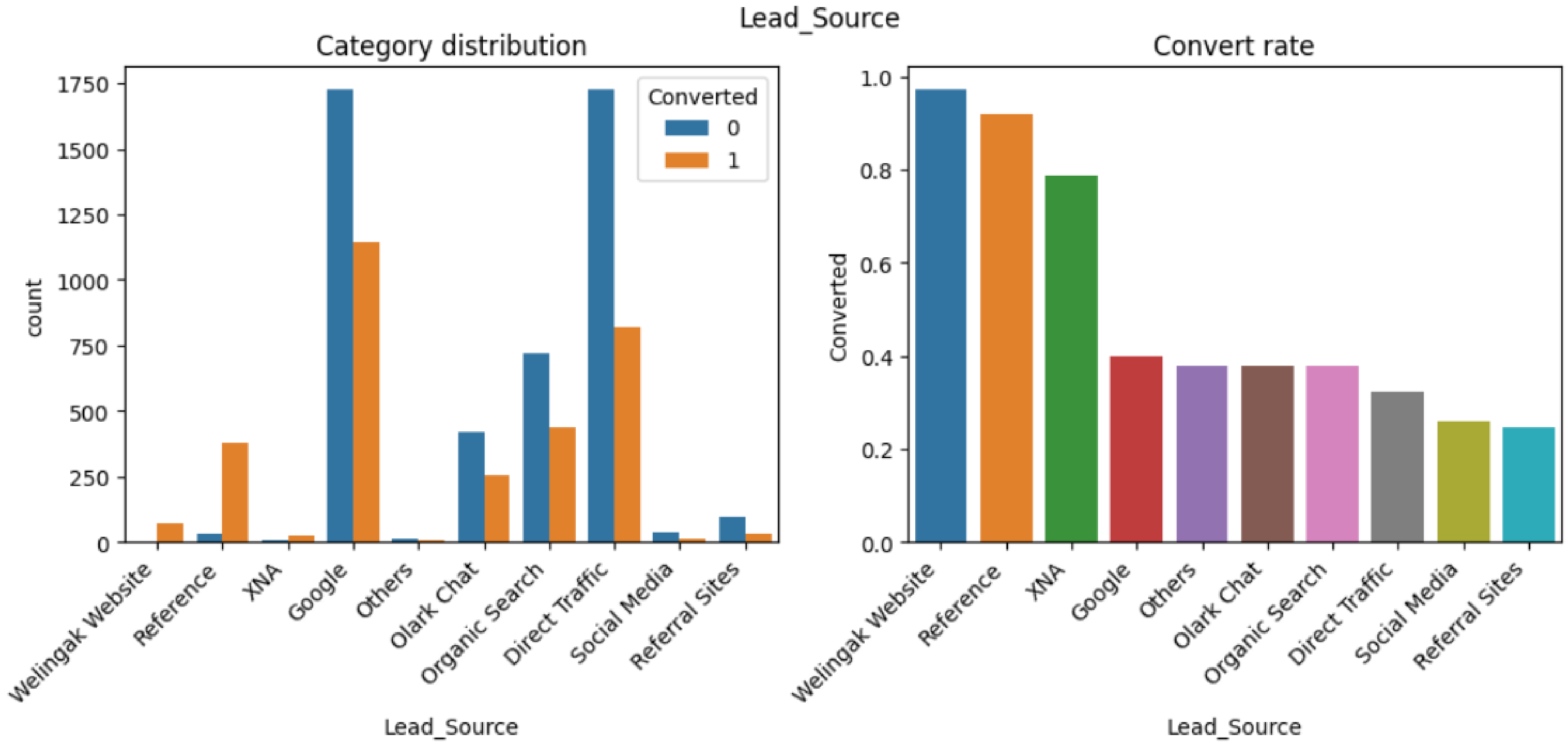
- Most leads have 5 or less visits. Hot leads have higher total time spent.





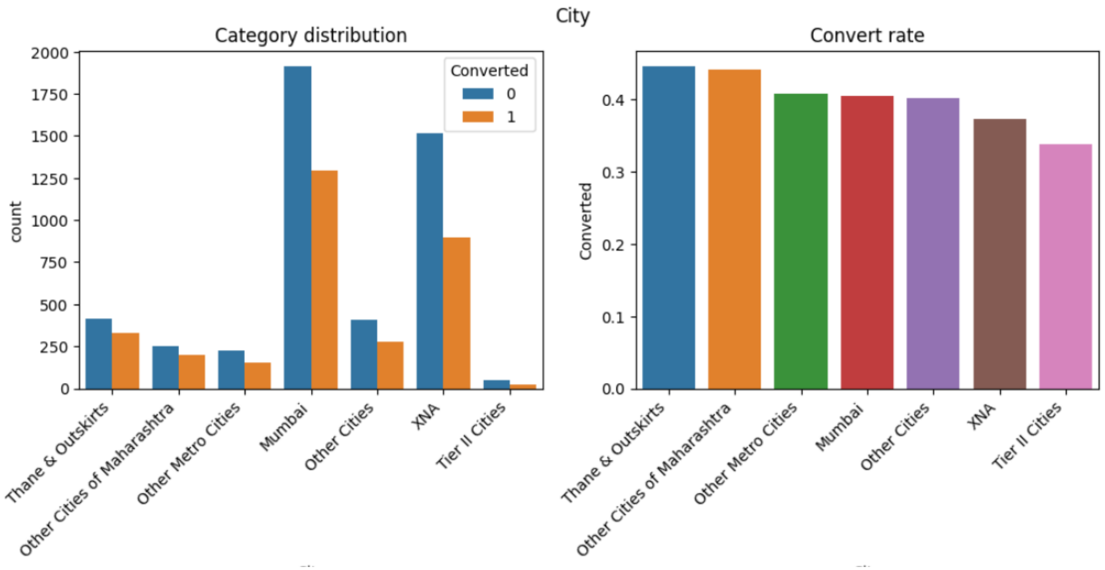
EDA main points

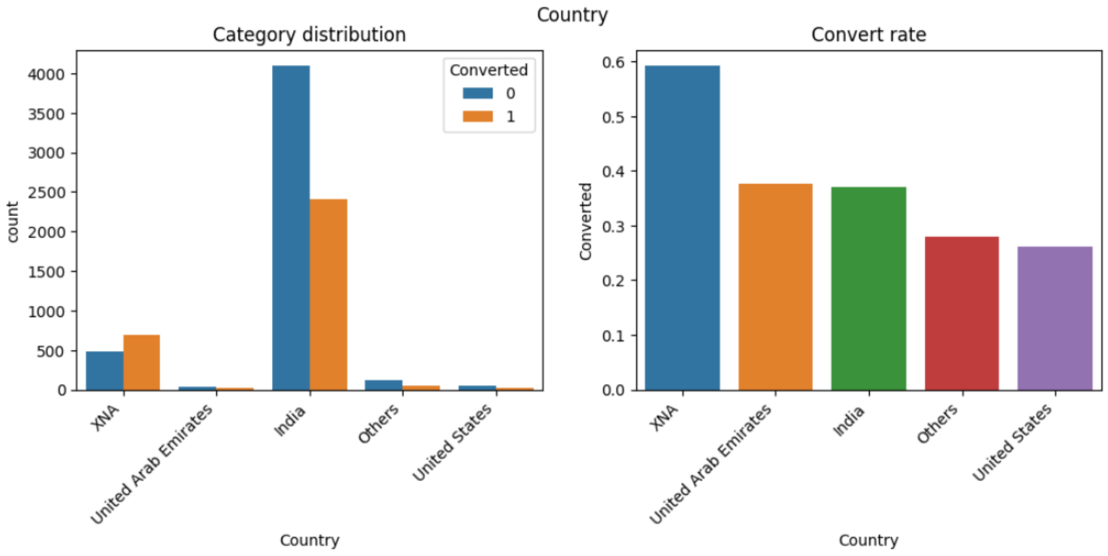
- Most of the leads are from Google and Direct Traffic yet they have a slightly below average convert rate.
- Convert rate for leads from Welingka and Reference are the highest while Social Media and Referral Sites are lowest.



EDA main points

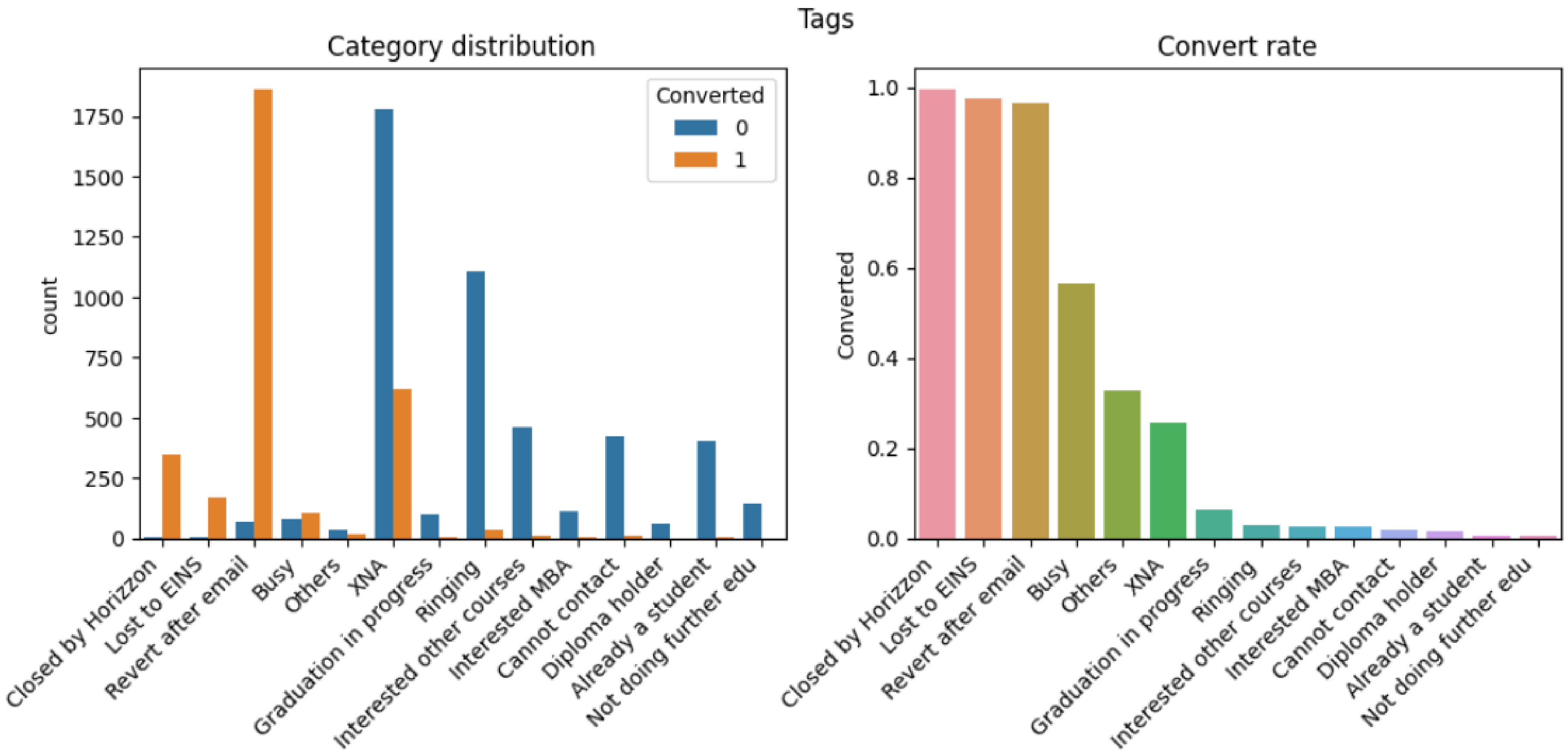
- There's not much different in convert rate among different categories in City, Country.





EDA main points

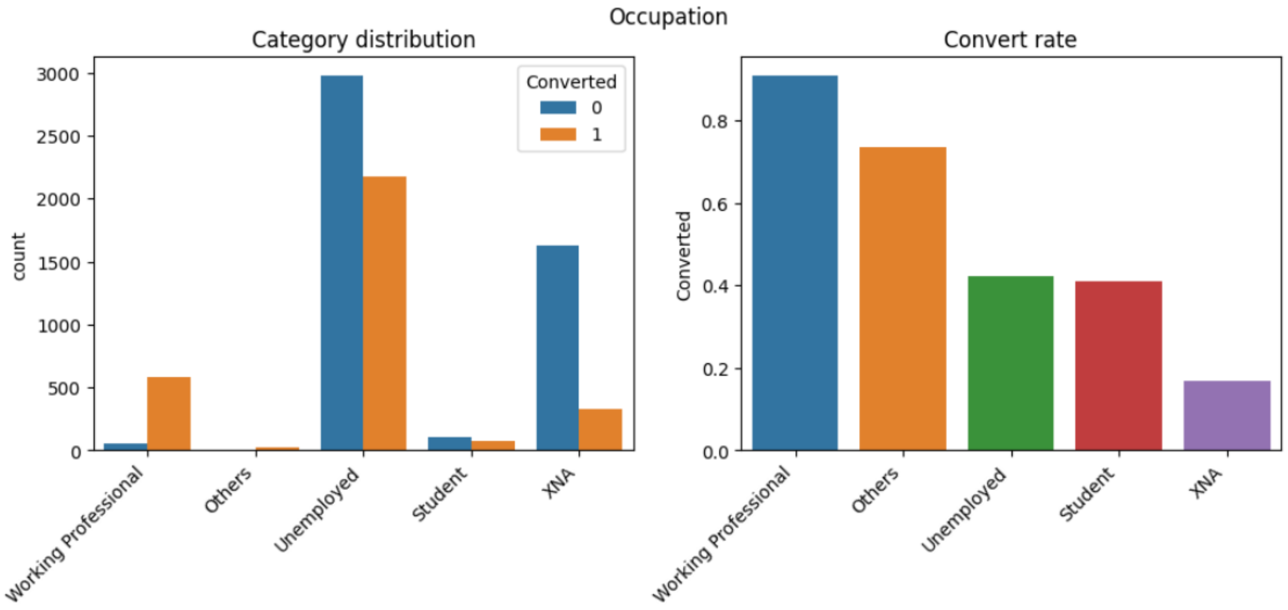
- Some tags like 'Revert after email' have near 100% convert rate. Looks like the email interaction effectively educated the leads about the product or service being offered, addressed any questions or concerns they may have, and provided them with the confidence/motivation to purchase.



EDA main points

- Occupation: Working professional has the highest convert rate but makes up a very small portion. The largest portion is Unemployment.
- Leads in Health care, Banking/Investment and Marketing,… has the highest conversion rate.

Occupation



Specialization

Data preparation

Convert categorical features

- Convert categorical variables to weight of evidence (WOE)
- Categorical features with high information values are dummies encoded

Example of replacing category with WOE:

```
df[['Tags']].head()
```

	Tags
0	Interested other courses
1	Ringing
2	Revert after email
3	Ringing
4	Revert after email

```
convert_woe('Tags')
```

Tags	Total	Good	Bad	WOE
Closed by Horizzon	347	345	2	4.345120
Revert after email	1924	1859	65	3.689994
Lost to EINS	172	168	4	3.376492
Busy	186	105	81	0.637864
Others	49	16	33	-0.271870
XNA	2402	620	1782	-0.647549
Graduation in progress	108	7	101	-1.934825
Diploma holder	60	1	59	-2.318569
Interested MBA	116	3	113	-2.541202
Interested other courses	473	13	460	-2.951815
Ringing	1143	34	1109	-2.993880
Not doing further edu	143	1	142	-3.152100
Cannot contact	429	8	421	-3.233739
Already a student	407	3	404	-3.785585

```
df[['Tags_WOE']].head()
```

	Tags_WOE
0	-2.951815
1	-2.993880
2	3.689994
3	-2.993880
4	3.689994

Convert categorical features

- Convert categorical variables to weight of evidence (WOE)
- Categorical features with high information values are dummies encoded

Features with high information values are also converted to dummies encoding:

	feature	IV
2	Tags	4.619689
8	Occupation	0.738028
3	Last_Activity	0.558294
4	Last_Notable_Activity	0.458297
1	Lead_Source	0.418798
0	Lead_Origin	0.408993
6	Country	0.112503
7	Specialization	0.072313
5	City	0.009754

Tags_Busy	Tags_Cannot_contact	Tags_Closed_by_Horizzon
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

[...]

Data scaling

- All features are min-max scaled
- Features that are highly correlated are dropped based on information values.

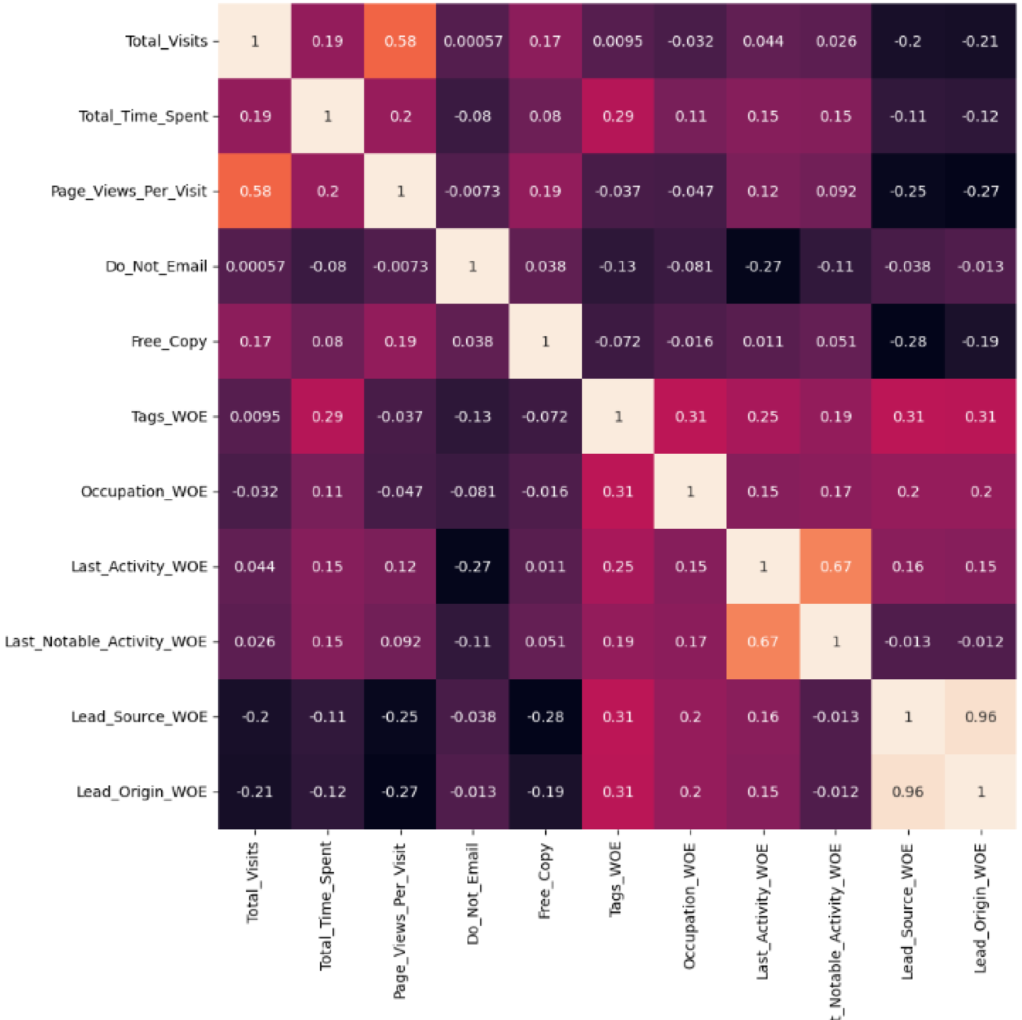
Feature scaling

```
# min max scaling
scaler = MinMaxScaler()

X_train_scaled = X_train.copy()
X_train_scaled[X_train_scaled.iloc[:,1:].columns] = scaler.fit_transform(X_train_scaled.iloc[:,1:])

X_test_scaled = X_test.copy()
X_test_scaled[X_test_scaled.iloc[:,1:].columns] = scaler.transform(X_test_scaled.iloc[:,1:])
```

Correlation heatmap



Model building, tuning and evaluation

Feature selection

- Feature selection are done with RFE and VIF values (<5)
- For the first model, categorical features' WOE are used instead of their dummies.
- After identifying the most impacting categorical variables, their WOE are replaced with dummies for better interpretability
- This is to keep the model relatively simple at first and scale up in complexity as needed.

Max VIF based on n_features selected with RFE

	n_features	max_vif_value
4	9	1.601059
3	8	1.600138
2	7	1.595129
1	6	1.366133
0	5	1.363783

Select the top features with RFE

```
model1_top_features = rfe_top_features(woe_columns, 9)
print(model1_top_features)
```

```
Top 9 features:
['Total_Visits', 'Total_Time_Spent', 'Page_Views_Per_Visit',
'Do_Not_Email', 'Free_Copy', 'Tags_WOE', 'Occupation_WOE', 'La
st_Activity_WOE', 'Lead_Source_WOE']
```

VIF for selected features

	feature	VIF
0	const	10.84
3	Page_Views_Per_Visit	1.60
1	Total_Visits	1.53
6	Tags_WOE	1.37
9	Lead_Source_WOE	1.34
2	Total_Time_Spent	1.21
8	Last_Activity_WOE	1.20
7	Occupation_WOE	1.14
5	Free_Copy	1.11
4	Do_Not_Email	1.09

Model iteration

- Insignificant variables (p-value>0.05) are removed after each iteration

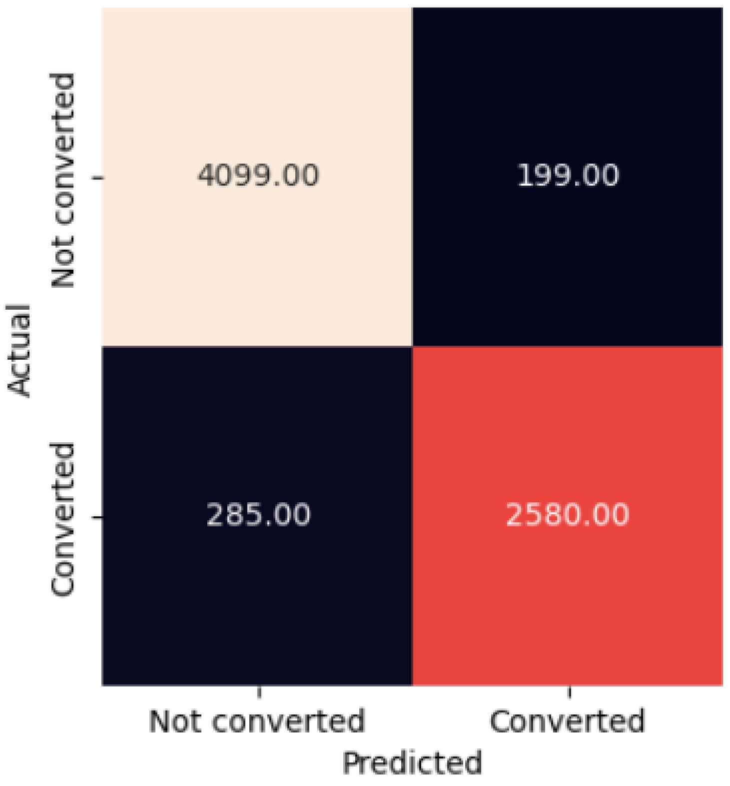
	P>|z|
const	0.000
Total_Visits	0.000
Total_Time_Spent	0.000
Page_Views_Per_Visit	0.000
Do_Not_Email	0.093
Free_Copy	0.000
Tags_WOE	0.000
Occupation_WOE	0.000
Last_Activity_WOE	0.000
Lead_Source_WOE	0.000

	P>|z|
const	0.000
Total_Visits	0.000
Total_Time_Spent	0.000
Page_Views_Per_Visit	0.000
Free_Copy	0.000
Tags_WOE	0.000
Occupation_WOE	0.000
Last_Activity_WOE	0.000
Lead_Source_WOE	0.000

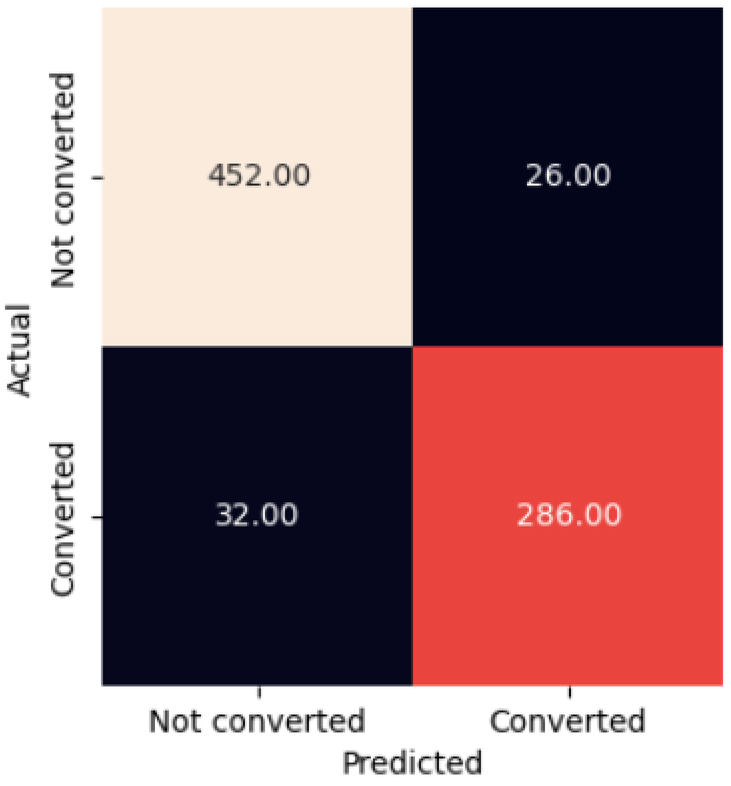
Model evaluation

- Due to business objective, F1 score will be the primary metric instead of accuracy. We want to limit both the false positives (waste of resources) and false negatives (loss of revenue).
- Model performance are evaluated based on the confusion matrix on both train and test data. This is to ensure the model is not overfitted.

Train set



Test set



Train

	precision	recall	f1-score	support
0	0.93	0.95	0.94	4298
1	0.93	0.90	0.91	2865
accuracy			0.93	7163
macro avg	0.93	0.93	0.93	7163
weighted avg	0.93	0.93	0.93	7163

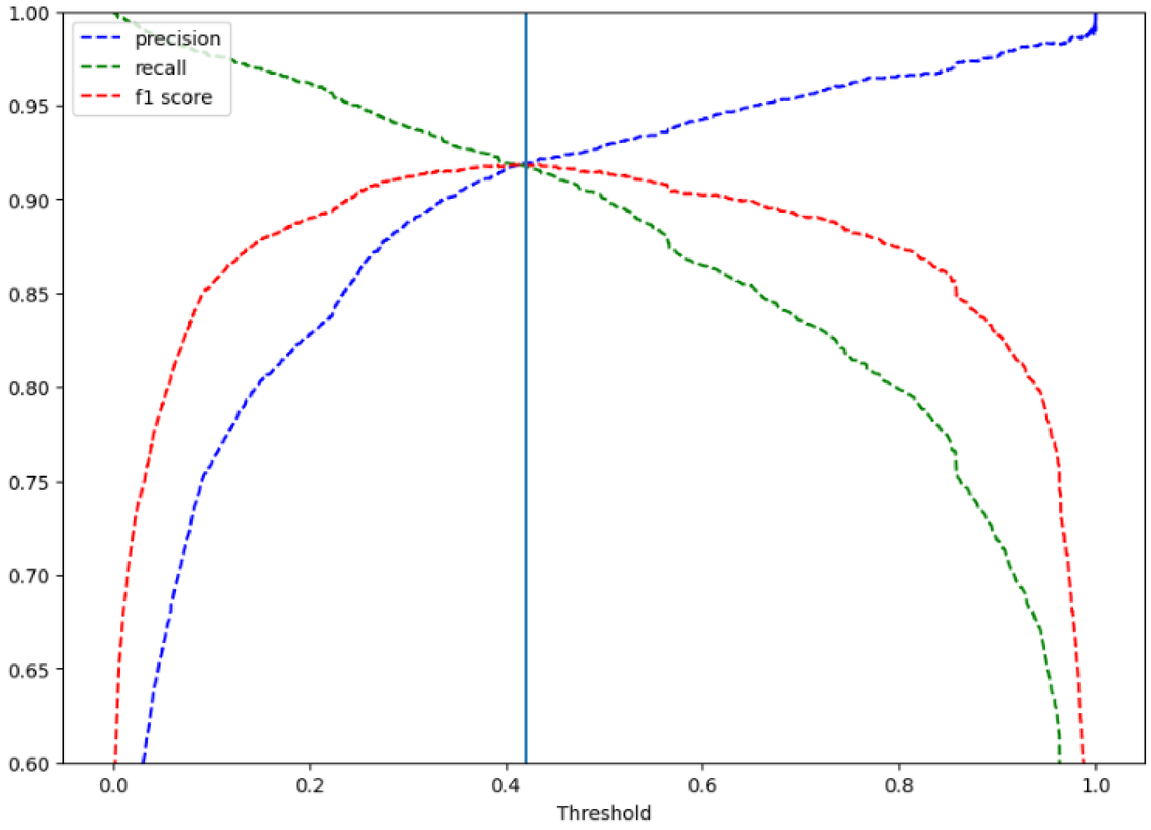
Test

	precision	recall	f1-score	support
0	0.93	0.95	0.94	478
1	0.92	0.90	0.91	318
accuracy			0.93	796
macro avg	0.93	0.92	0.92	796
weighted avg	0.93	0.93	0.93	796

Threshold optimization

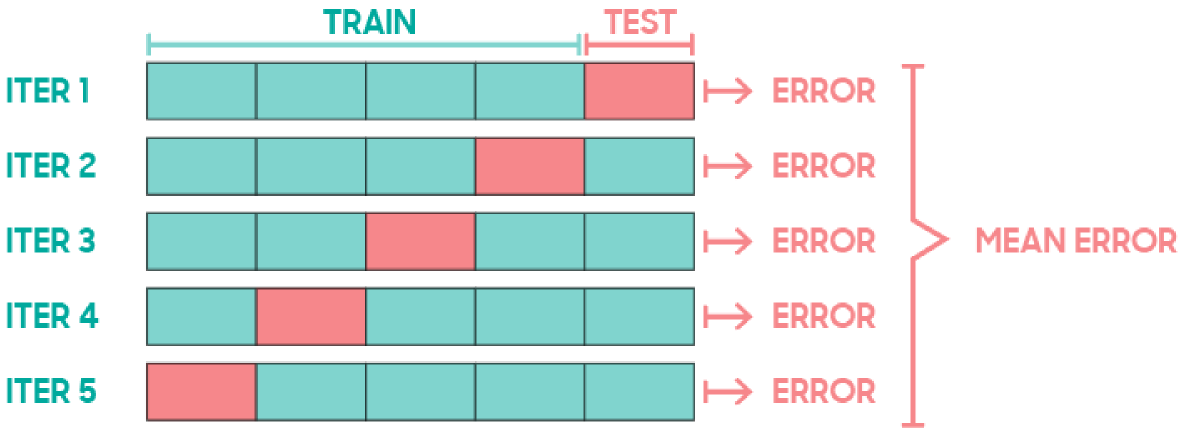
- Plot the precision-recall curve to determine the optimal threshold for prediction

```
pre_rec_curve(model3, model3_top_features, 0.42)
```



K-fold cross validation

- Due to the small size of data, we only have a small test size.
- Kfold cross validation splits the data into different train and test sets each iteration to have a more generalized performance of the model.




```
cv_result
```

	Model	Accuracy	Precision	Recall	F1	AUC
0	model1	0.924237	0.907712	0.902293	0.904994	0.973579
1	model2	0.930645	0.918283	0.907320	0.912769	0.974302
2	model3	0.932278	0.918354	0.911719	0.915024	0.974777

Final model selection

Final model

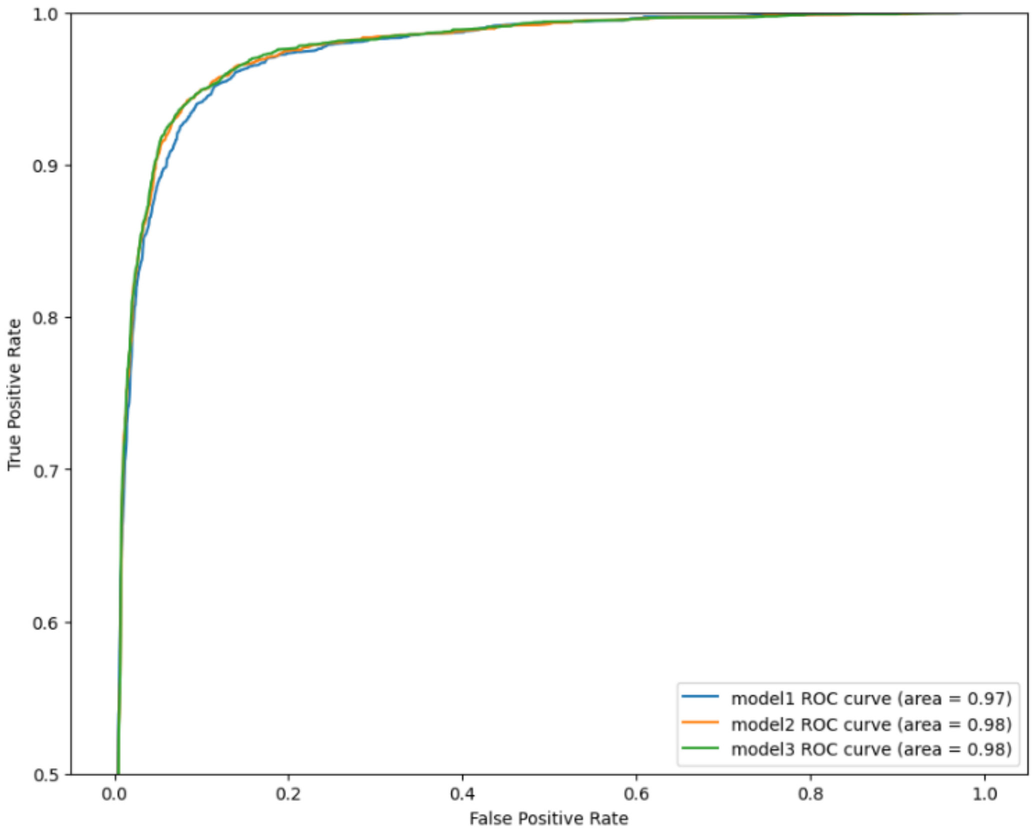
- Final model is selected based on F1 and AUC score, with consideration to interpretability.
- Model 3 has the highest AUC and F1 score, with the highest interpretability in term of features.
- AUC = 0.975, F1 = 0.915, Accuracy = 0.932

Kfold cross validation result

```
cv_result
```

	Model	Accuracy	Precision	Recall	F1	AUC
0	model1	0.924237	0.907712	0.902293	0.904994	0.973579
1	model2	0.930645	0.918283	0.907320	0.912769	0.974302
2	model3	0.932278	0.918354	0.911719	0.915024	0.974777

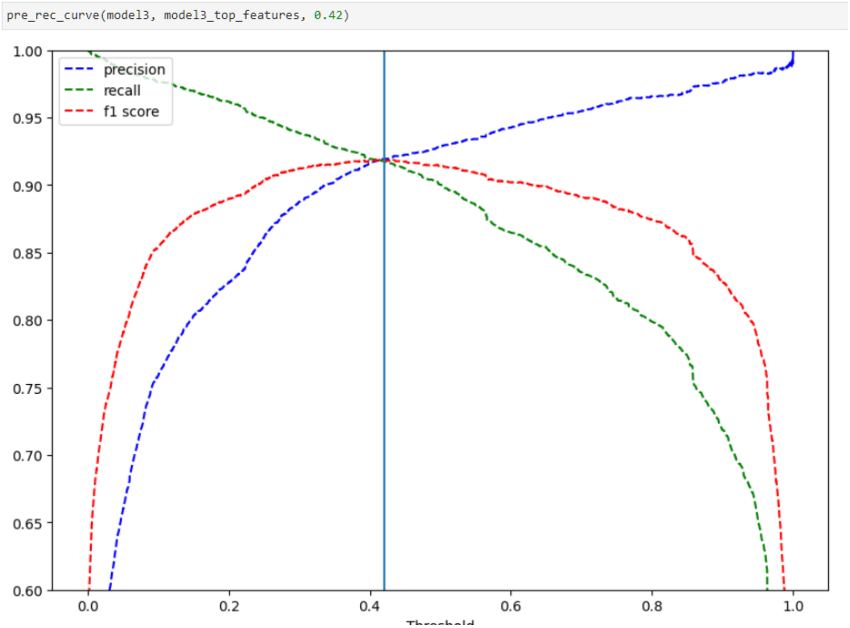
AUC of different models



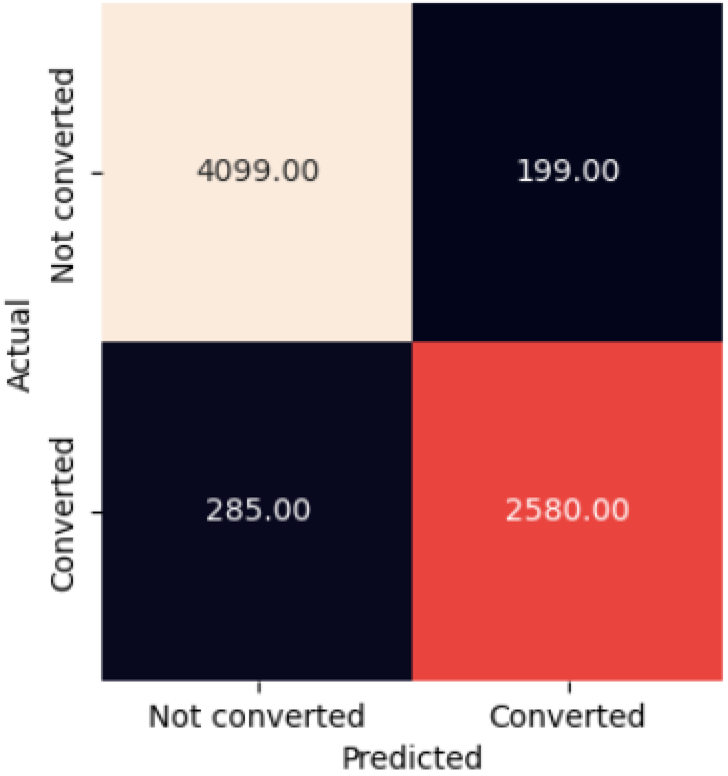
Final model

- Model3 confusion matrix on train and test set at the optimal threshold (0.42)

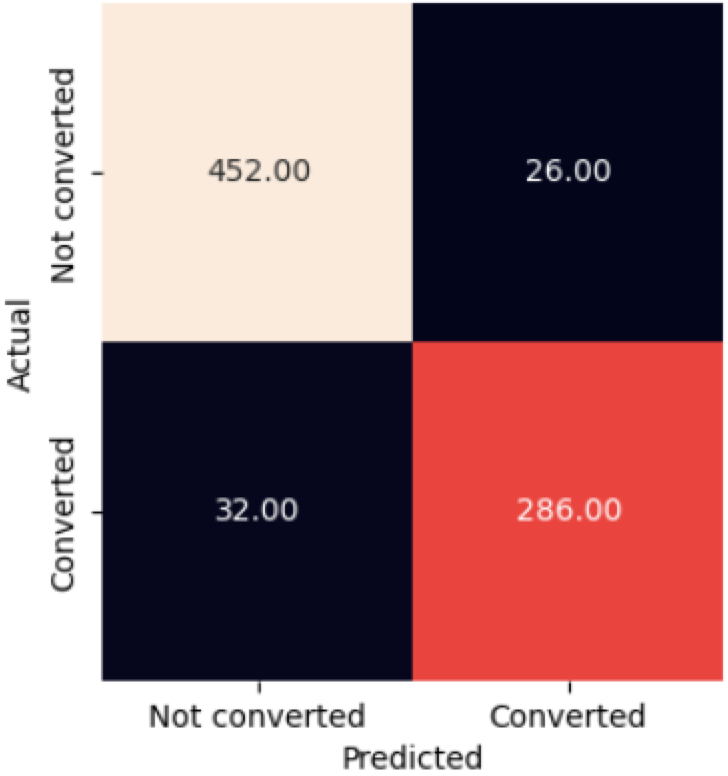
Optimal threshold



Train set



Test set



Features importance

Features that have **positive effects** the chance of conversion

- **Total_Time_Spent** and **Total_Visits** are 2 numerical features that positively affect the chance of conversion. This indicate that the people who are interested tend to visit more often and spend more time on each visit.
- Leads that were tagged as **Revert_after_email**, **Lost_to_EINS**, **Closed_by_Horizzon**,… also have a higher chance of being converted.
- Other features are **Last_Activity_SMS_Sent, Lead_Source_WOE, Occupation_WOE**.

```
model3.params[model3.params>0].sort_values
```

```
Tags_Lost_to_EINS          4.896269
Tags_Closed_by_Horizzon    4.792340
Occupation_WOE             4.526694
Total_Time_Spent           4.093767
Tags_Revert_after_email    3.040651
Total_Visits               3.016675
Lead_Source_WOE            1.800476
Last_Activity_SMS_Sent     1.499414
```

Features importance

Features that have **negative effects** the chance of conversion

- Higher **Page_Views_Per_Visit** negatively correlated with the chance of conversion. Some of the possible explanations could be:
    - unable to find the information they are looking for (course description/fee/requirement/etc...)
    - browsing though different courses/programs and unable to choose one.
- Other tags such as
    - Tags related to the leads' education status (**Diploma_holder, Not_doing_further_edu,...**)
    - Tags related to the leads' interest (**Interested_MBA, Interested_in_other_courses,...)**
    - Tags regarding contactability (**Cannot_contact**)

  can negative correlate with the chance of being converted as well.

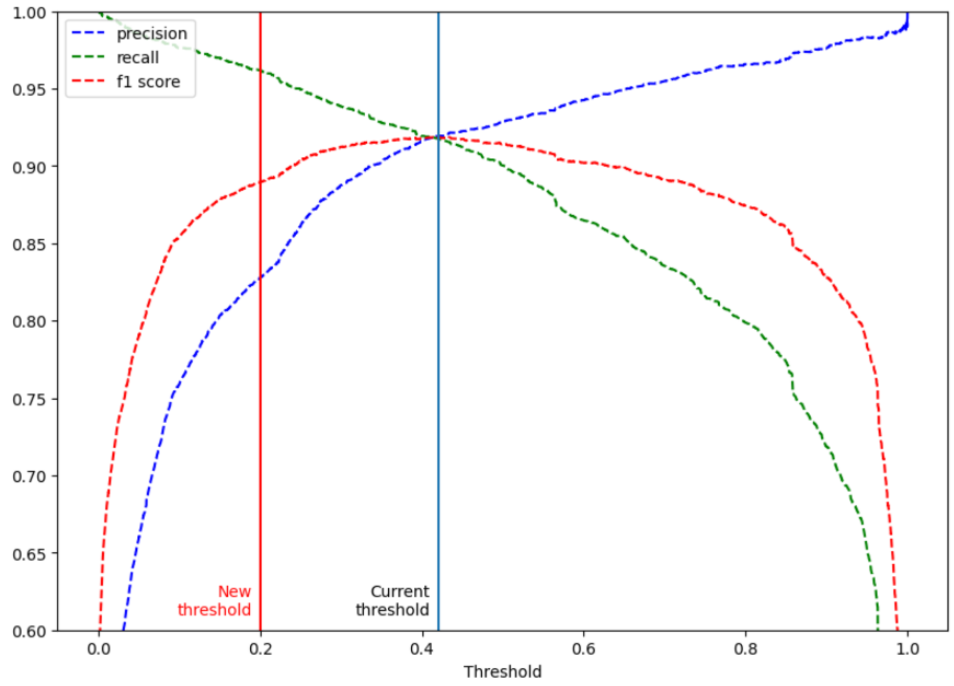
```
model3.params[model3.params<0].sort_values()
```

```
Tags_Diploma_holder                       -4.862708
Tags_Not_doing_further_edu                -4.449880
Tags_Already_a_student                    -4.314084
Tags_Interested_MBA                       -4.309675
Tags_Cannot_contact                       -4.271480
Tags_Ringing                              -4.028486
Tags_Interested_other_courses             -3.448672
const                                     -3.054744
Tags_Graduation_in_progress               -2.126932
Page_Views_Per_Visit                      -1.799430
Last_Activity_Converted_to_Lead           -1.444002
Last_Activity_Email_Bounced               -1.213302
Last_Activity_Form_Submitted_on_Website   -1.145996
Tags_Others                               -1.089809
Last_Activity_Page_Visited_on_Website     -0.803544
Do_Not_Email                              -0.695509
Last_Activity_Olark_Chat_Conversation     -0.611650
Free_Copy                                 -0.353726
```

Question 3

Question:
X Education has a period of 2 months every year during which they hire some interns. The sales team, in particular, has around 10 interns allotted to them. So during this phase, they wish to make the lead conversion more aggressive. So they want almost all of the potential leads (i.e. the customers who have been predicted as 1 by the model) to be converted and hence, want to make phone calls to as much of such people as possible. Suggest a good strategy they should employ at this stage.

```
pre_rec_curve(model3, model3_top_features, threshold = 0.42, new = 0.2)
```

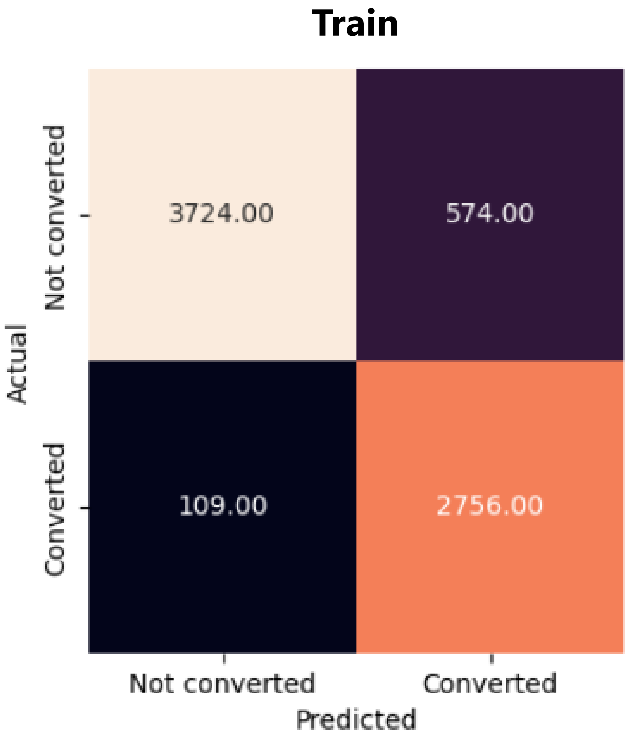


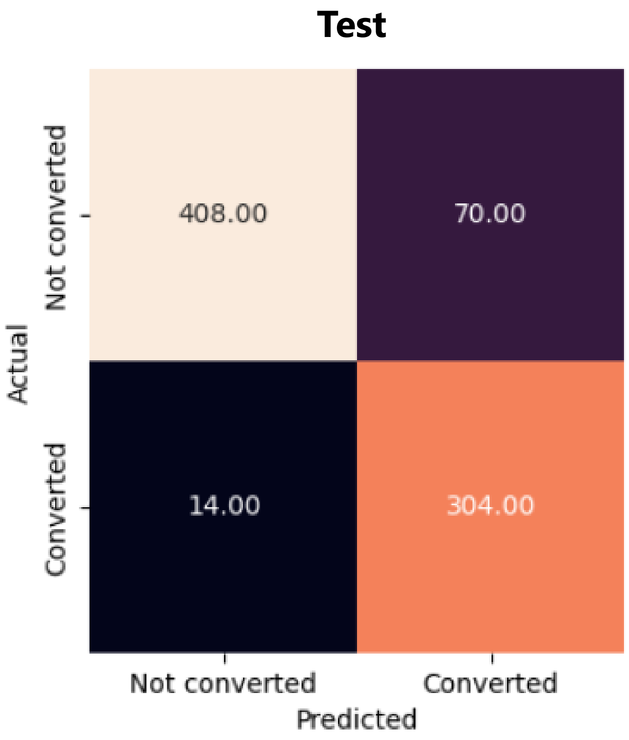
Answer:
The solution is to lower the threshold for 'Convert' prediction. In technical term, this will increase recall at the cost of reduced precision (lower false negative rate but higher false positive rate). In business term, there will be more leads get classified as hot leads for the interns to work on, but the chance of conversion of these leads will be lower.

The exact threshold adjustment should be made according to the capability of the interns. A suggested range can be around 0.2~0.25. Any lower will result in a steep decline in precision without any significant gain in recall.

Question 3

Model 3 performance at adjusted threshold = **0.2**





Train

	precision	recall	f1-score	support
0	0.97	0.87	0.92	4298
1	0.83	0.96	0.89	2865
accuracy			0.90	7163
macro avg	0.90	0.91	0.90	7163
weighted avg	0.91	0.90	0.91	7163

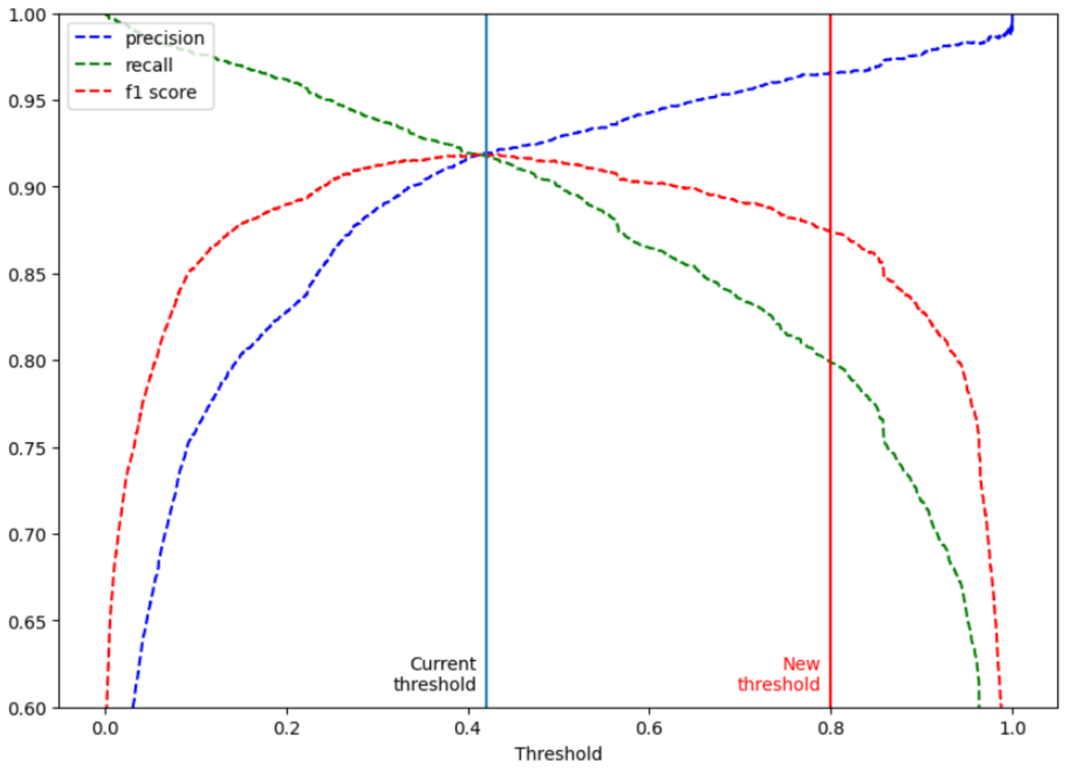
Test

	precision	recall	f1-score	support
0	0.97	0.85	0.91	478
1	0.81	0.96	0.88	318
accuracy			0.89	796
macro avg	0.89	0.90	0.89	796
weighted avg	0.91	0.89	0.90	796

Question 4

Question:
Similarly, at times, the company reaches its target for a quarter before the deadline. During this time, the company wants the sales team to focus on some new work as well. So during this time, the company's aim is to not make phone calls unless it's extremely necessary, i.e. they want to minimize the rate of useless phone calls. Suggest a strategy they should employ at this stage.

```
pre_rec_curve(model3, model3_top_features, threshold = 0.42, new = 0.8)
```

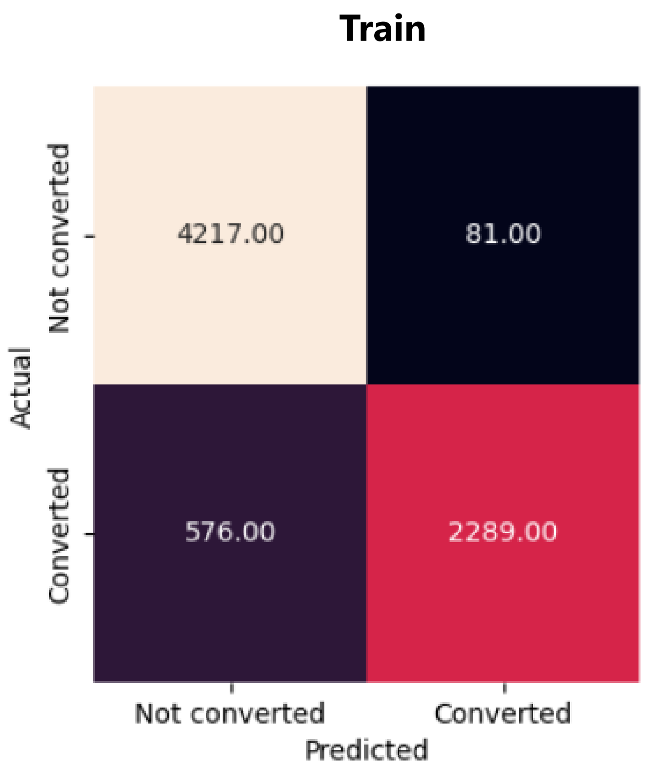


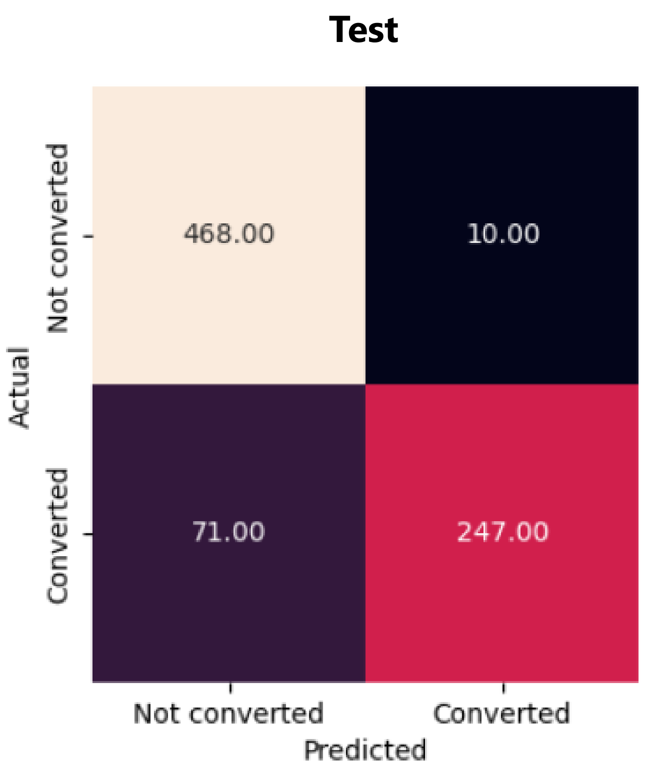
Answer:
The solution is the exact opposite of the previous question, which is to increase the threshold for hot leads. Doing this will reduce the number of leads that classified as hot, but also will reduce the false negative rate. The final result is that the team will have fewer, but more promising leads to contact.

The exact threshold increase still needs to be discussed, but around 0.7~0.8 should be a good starting point. Depends on what the team's definition of "Extremely necessary" is, the threshold can be pushed as high as 0.95~0.99.

Question 4

- Model 3 performance at adjusted threshold = **0.8**
- 20% less leads classified as hot, precision increase to 0.96 (from 0.9)





Train

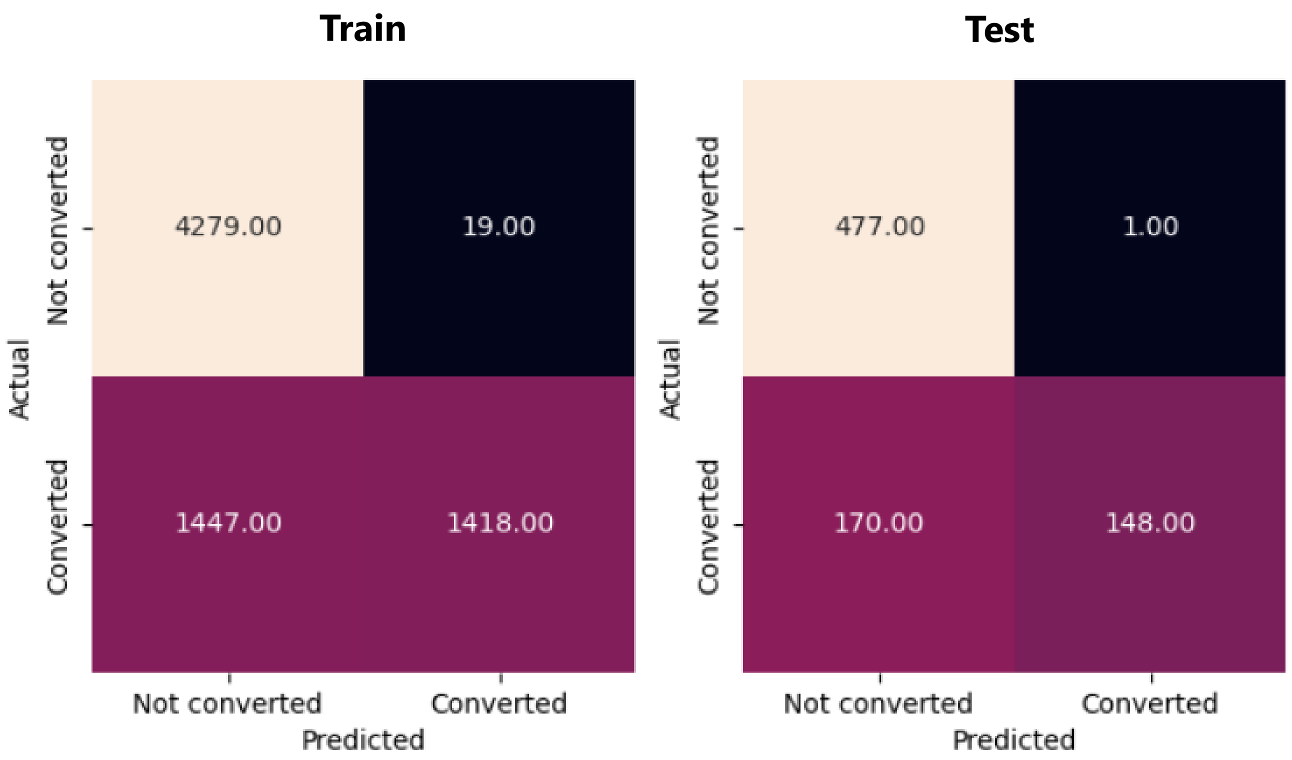
	precision	recall	f1-score	support
0	0.88	0.98	0.93	4298
1	0.97	0.80	0.87	2865
accuracy			0.91	7163
macro avg	0.92	0.89	0.90	7163
weighted avg	0.91	0.91	0.91	7163

Test

	precision	recall	f1-score	support
0	0.87	0.98	0.92	478
1	0.96	0.78	0.86	318
accuracy			0.90	796
macro avg	0.91	0.88	0.89	796
weighted avg	0.91	0.90	0.90	796

Question 4

- Model 3 performance at adjusted threshold = **0.98**
- 50% less leads classified as hot, precision increase to 0.99 (from 0.9)



Train

	precision	recall	f1-score	support
0	0.75	1.00	0.85	4298
1	0.99	0.49	0.66	2865
accuracy			0.80	7163
macro avg	0.87	0.75	0.76	7163
weighted avg	0.84	0.80	0.78	7163

Test

	precision	recall	f1-score	support
0	0.74	1.00	0.85	478
1	0.99	0.47	0.63	318
accuracy			0.79	796
macro avg	0.87	0.73	0.74	796
weighted avg	0.84	0.79	0.76	796

Thank You