Machine Learning for Data Science – Bank Customer Churn Rate

Final Project DS105 | Lee Yi Ping

Background

- Customer Churn, also known as customer turnover, or customer defection, in the loss of clients or customers.

- Bank, insurance companies, streaming services companies and telecom service companies, often use customer churn analysis and customer churn rates as one of their key business metrics because the cost of retaining existing customers is far less than acquiring a new one.

- The bank decided to collect data from 6 months period to evaluate the problem after noticing increase in the number of customers leaving the bank.

Problem Statement

- The project objective is to develop churn prediction model on how likely it's current customers will be exiting the bank (i.e. close their bank account) in near future.

- Our task is a binary classification problem to classify a customer whether he or she will exit or not given how long they have been a customer of the bank, whether they are active member, their age, gender, credit score, estimated salary, number of products and if the customer holds a credit card or not.

Dataset – Bank Customer

- It contains 10,000 rows and 14 attributes
- 10,000 customers were selected randomly among three countries – France, Germany and Spain.
- Well-labelled to explain all its columns and the target variable is a binary variable reflecting the fact whether the customer exit the bank (closed account) or continues to be a customer.

Sample of Dataset

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedS
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	1013
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	1125
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	1139
3	4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	938
4	5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	790
...	...	...	...	...	...	...	...	...	...	...	...	...	
9995	9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	962
9996	9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	1016
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	420
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	928
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	381

10000 rows × 14 columns

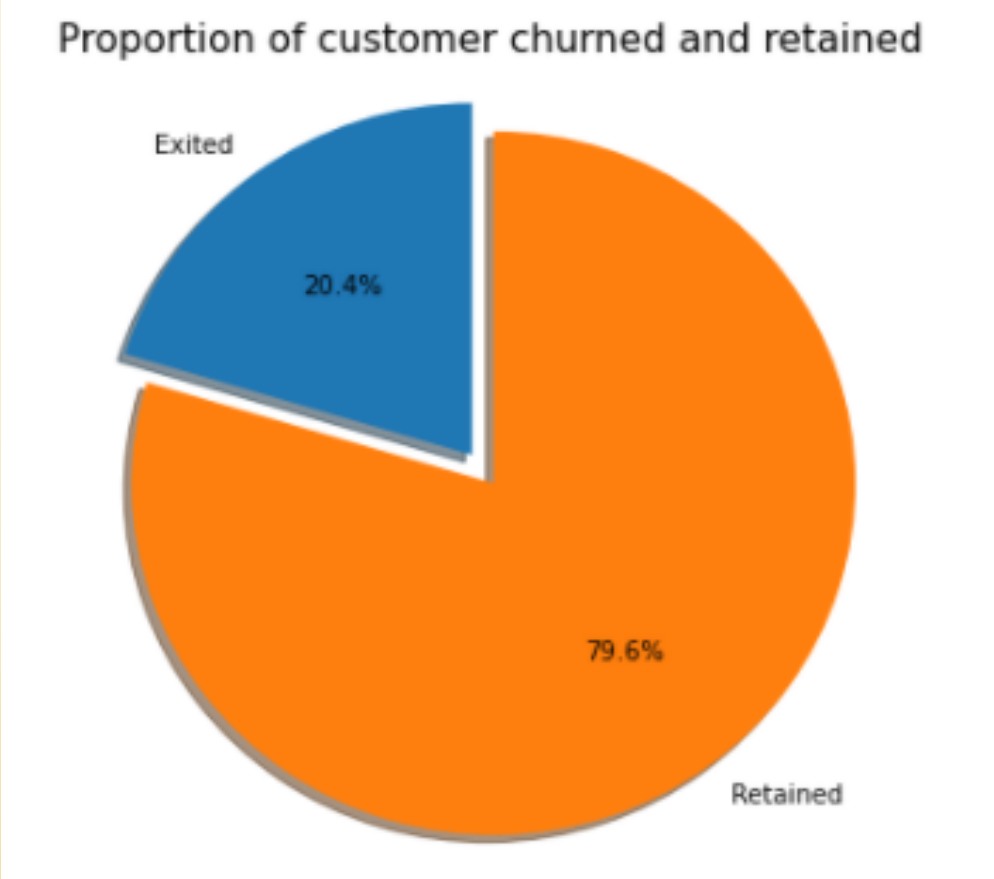
Important attributes from the dataset

- Age
  - Older individuals are more likely to have a good relationship with their bank and are less likely to end their engagement.
- Exited: Churned or not? ("0=No,1=Yes")
  - It is most unlikely that an active member will exit the bank.
- Account balance, number of product
  - Although the customer did not have any balances in their account or does not have any product with the bank when the data was collected, it does not necessary indicate customer churn.

Challenges and Difficulties

- To identify outliers or unbalanced dataset.
  i. How to deal with outliers which to keep or drop to prevent inaccurate results from prediction.
  ii. Check and understand which metrics to handle imbalance data to have better performance.
- Selection of encoding method for categorical feature.
- To decide which features to drop that is not needed for the modelling.
- How to have best hyperparameter tuning results on the selected model to improve the model performance.

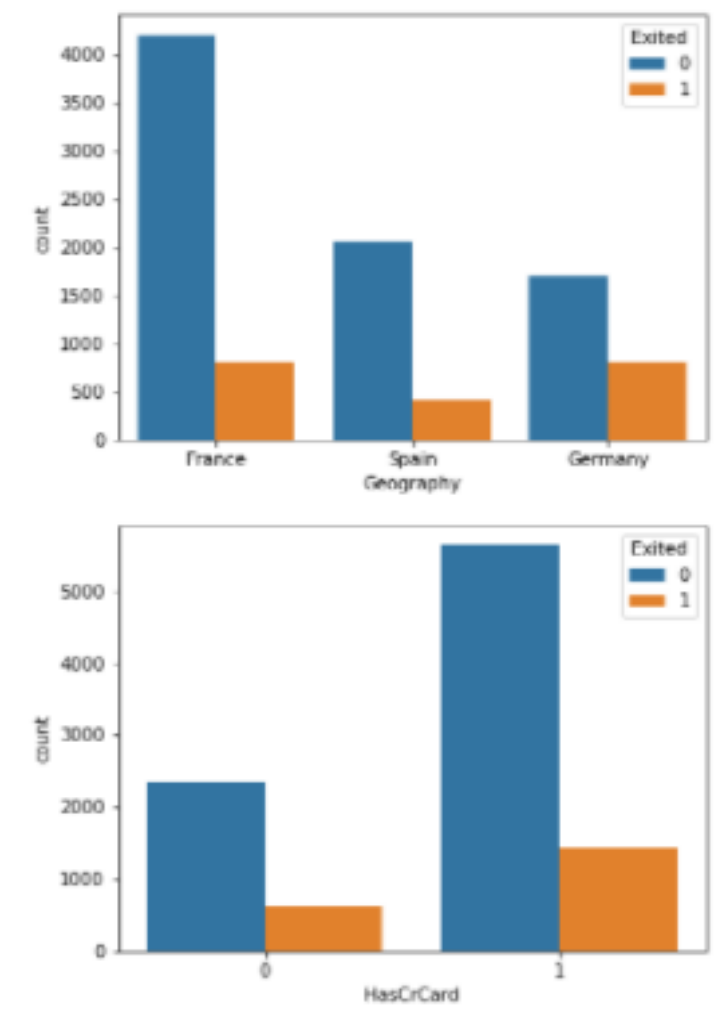
Data Exploration/Insights



- As we can see from the pie chart, there are about 20% of the customer have churned.
- 20.4 % customers churned(leave) from the bank and 79.6% customers retained
- Given 20% is a small number, we need to ensure that the chosen model does predict with great accuracy on this 20% as it is of interest to the bank to identify and keep this bunch as opposed to accurately predicting the customers that are retained.
- Generally, it is common to see imbalanced data in Churn and Fraud datasets.

Data Exploration/Insights

- ***Analysis – Categorical Attributes***

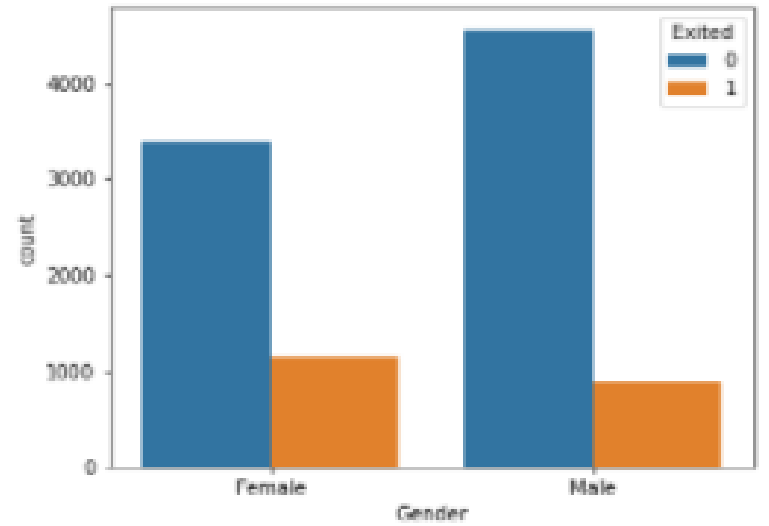


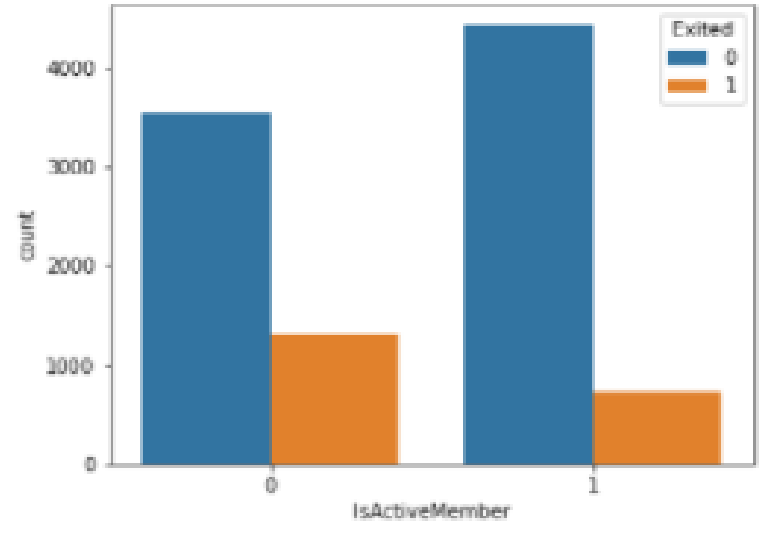
1. Majority customers from France but customers in Germany are more likely to churn than customers in France and Spain.

2. Interestingly, majority of the customers that churned are those with credit card which is 70% of the customers owe a credit card.

Data Exploration/Insights

- ***Analysis – Categorical Attributes***



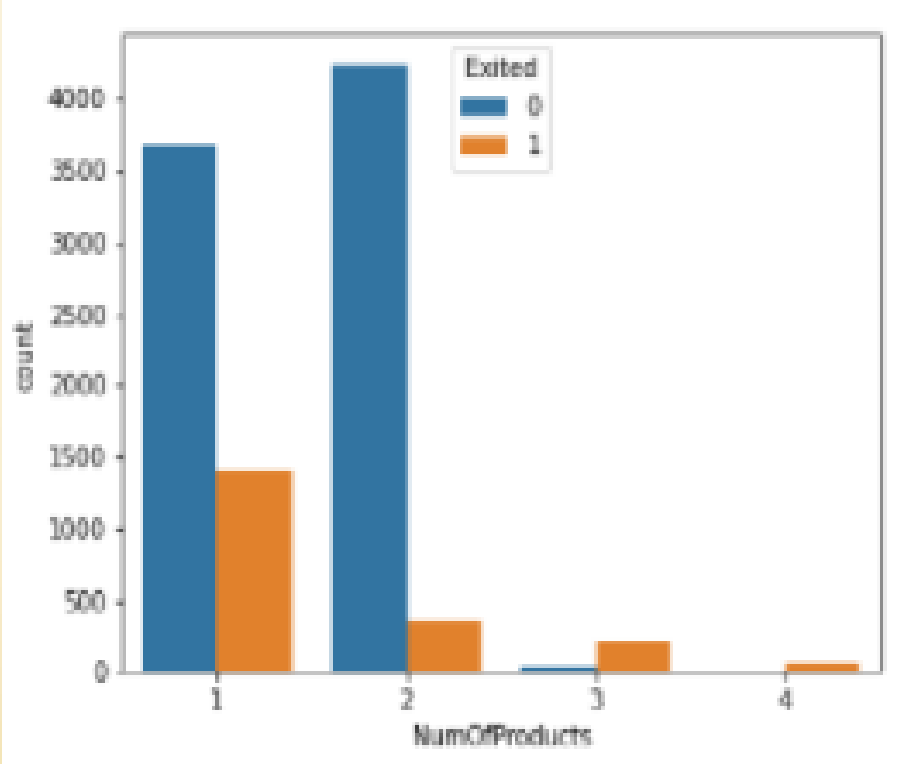


3. Proportion of female customers churning is greater than male customers. Although in our data sample there are more males than females.

4. In the chart, we can see that it is inversely that active tend to be exit from the bank and at least 50% of the customers are active.

Data Exploration/Insights

- ***Analysis – Categorical Attributes***



5. Customers who has 3 or 4 product are extremely likely to churn, whereby customers who has 2 products will most likely to stay with the bank.

Key Steps in Data Pre-processing

Missing Observation Analysis	There is no null values
Outlier Observation Analysis	No Outlier
Duplicate Values Checking	There is no duplicate values as customer ID is unique
Drop Unwanted Column	Drop features not needed for machine learning: "RowNumber","CustomerId","Surname"
Correlation between Variables Checking	Less correlation among all columns and no multicollinearity.
Train Test Split	Using Stratify to split the dataset for training.
Encoding Categorical Values	Encode “Gender” and “Geography” using OHE before passing it into model.
Feature Scaling	Using Standard Scaler to scale those attributes have bigger range, i.e. more than 10 (Balance, Credit Score and Estimated Salary)

Training Process of the machine learning model

1. Lazy Prediction

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
XGBClassifier	0.85	0.71	0.71	0.84	0.27
AdaBoostClassifier	0.85	0.71	0.71	0.84	0.30
LGBMClassifier	0.86	0.71	0.71	0.84	0.11
BaggingClassifier	0.85	0.71	0.71	0.84	0.23
RandomForestClassifier	0.86	0.70	0.70	0.84	0.78
ExtraTreesClassifier	0.85	0.69	0.69	0.84	0.64
DecisionTreeClassifier	0.79	0.69	0.69	0.79	0.05
SVC	0.86	0.69	0.69	0.84	1.12

- As we can see from above, the scores for all models looks identical.

How lazy predict's parameters work

> verbose: int data type, if non zero, progress messages are printed. Above 50, the output is sent to stdout. The frequency of the messages increases with the verbosity level. If it more than 10, all iterations are reported. I would suggest you try different values based on your depth of analysis.

>-predictions: Boolean data type if it is set to True then it will return all the predicted values from the models.

Classic will return two values:

> models: will have all the models and with some metrics

>-predictions: will have all the predicted values that is $\hat{y}$

Training Process of the machine learning model

2. Perform baseline model testing of the top 8 models chosen from lazy prediction.

- As mentioned, all the metric results are quite close, makes it hard to compare which is the better model hence we proceed to have log loss for comparison.

	model	roc	logloss	accuracy	precision	recall	f1_score
0	AdaBoost	0.71	5.01	0.85	0.72	0.48	0.57
1	RandomForest	0.71	4.84	0.86	0.76	0.46	0.57
2	XGBoost	0.71	5.18	0.85	0.69	0.49	0.57
3	SVC	0.50	7.03	0.80	0.00	0.00	0.00
4	lgbm	0.71	4.94	0.86	0.73	0.47	0.57
5	DecisionTrees	0.68	7.39	0.79	0.48	0.51	0.49
6	ExtraTrees	0.70	4.94	0.86	0.77	0.43	0.55
7	Bagging	0.69	5.32	0.85	0.69	0.44	0.54

How to consider which log loss is the best?
- The lower the loss, the better model is.

Hyperparameter Optimization – GridSearchCV

1. Random Forest Classifier

- Due to it is imbalanced data, we are using "class weight" to adjust the weight of target.

- Best parameters after 5 x attempts :
  - "n_estimators": [90,100,110,120,130] **– 110 the best**
  - "max_depth": [24,25,26,27,28] **– 26 the best**

- Overall results before and after tune does not have a huge discrepancy or improvement. Hence we proceed to check on confusion matrix.

Before Tune

	model	roc	logloss	accuracy	precision	recall	f1_score
1	RandomForest	0.71	4.84	0.86	0.76	0.46	0.57

After Tune

	precision	recall	f1-score	support
0	0.87	0.96	0.91	1593
1	0.75	0.44	0.56	407
accuracy			0.86	2000
macro avg	0.81	0.70	0.74	2000
weighted avg	0.85	0.86	0.84	2000

```
Log loss of tuned: 4.956338800391304
ROC of tuned: 0.7032124574497456
```

1. To optimize the parameters we will use GridSearch, an exhaustive search with a selected subset of relevant hyperparameters. Scikit-learn has a library specially dedicated to this task : GridSearchCV, we will use it with 5-fold cross-validation.

Hyperparameter Optimization

2. LGBM Classifier

- Due to it is imbalanced data, we are using the parameter "is_unbalance" to adjust the weight of target.

- Best parameters after 2 x attempts :
  - "n_estimators": [395, 400, 410, 415, 420] – **410 the best**
  - "max_depth": [10, 20, 30, 40, 50] – **20 the best**
  - "learning_rate":[0.2, 0.25, 0.3, 0.35, 0.4] – **0.30 the best**

- Comparison of before and after tune metric results, although accuracy have dropped, recall improved by 6%. Overall metric results does not show improvement and log loss have also increased.

Before Tune

	model	roc	logloss	accuracy	precision	recall	f1_score
4	lgbm	0.71	4.94	0.86	0.73	0.47	0.57

After Tune

	precision	recall	f1-score	support
0	0.88	0.92	0.90	1593
1	0.62	0.53	0.57	407
accuracy			0.84	2000
macro avg	0.75	0.72	0.74	2000
weighted avg	0.83	0.84	0.83	2000

```
Log loss of tuned: 5.543525585216128
ROC of tuned: 0.7245527499764788
```

Hyperparameter Optimization

3. AdaBoost Classifier

- Due to it is imbalanced data, we are using "sample_weight" to adjust the weight of target.

- Best parameters after 1 x attempt:
  - "n_estimators": [380,390,400,410,420] – **380 the best**
  - "learning_rate":[0.31,0.32,0.33,0.34,0.35] – **0.33 the best**
  - As the 2 parameters will keep changing due to another parameters change.

- Compare before and after tune results, accuracy have dropped but recall score increased by 0.32. However, log loss have been increased by 2.45 and precision have decreased 0.36.

Before Tune

	model	roc	logloss	accuracy	precision	recall	f1_score
0	AdaBoost	0.71	5.01	0.85	0.72	0.48	0.57

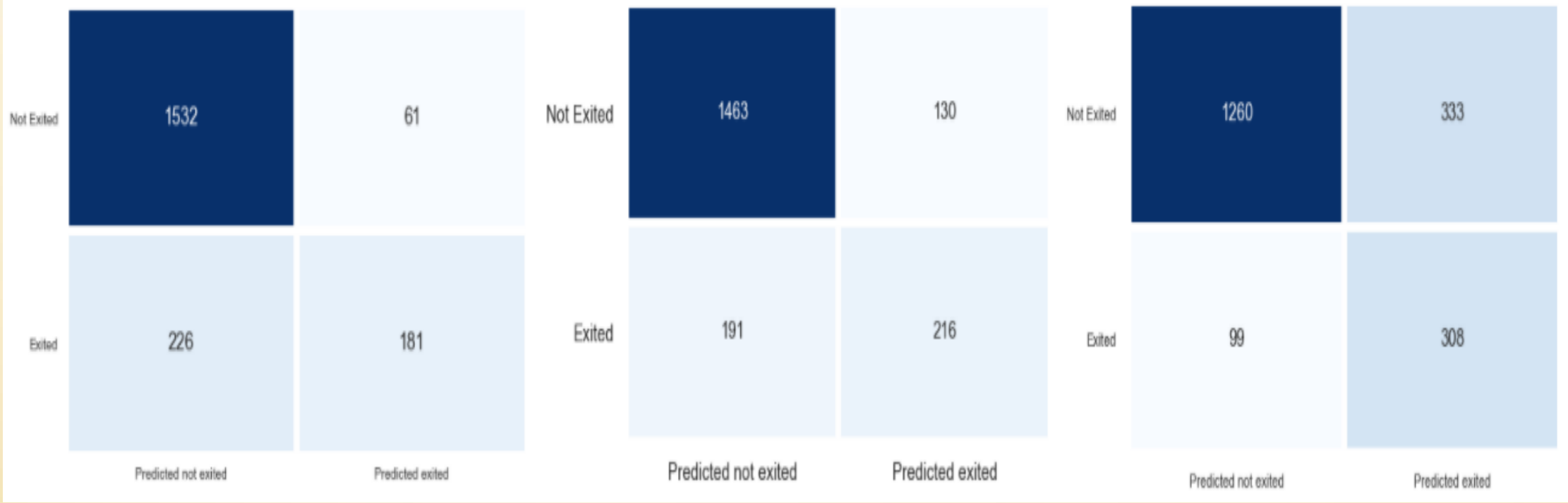
After Tune

	precision	recall	f1-score	support
0	0.93	0.79	0.85	1593
1	0.48	0.76	0.59	407
accuracy			0.78	2000
macro avg	0.70	0.77	0.72	2000
weighted avg	0.84	0.78	0.80	2000

```
Log loss of tuned: 7.460508834272835
ROC of tuned: 0.7738586043670791
```

Confusion Matrix for all 3 models after tuning

Random Forest	LGBM	AdaBoost



- Based on the confusion matrix above, Random Forest have the best true negative rate and lowest false positive rate.
- Although LGBM do not have the best true negative rate compare to RandomForest, overall LGBM have the best average results.
- Lastly, AdaBoost have the lowest false negative rate and highest true positive rate (75% of recall).

Conclusion

Best model performance: AdaBoost Classifier

i. Better recall & F1 score
ii. Highest ROC score
iii. It has the best true positive and false negative rate compare to others.

<u>Reasons for not choosing the other 2 models:</u>

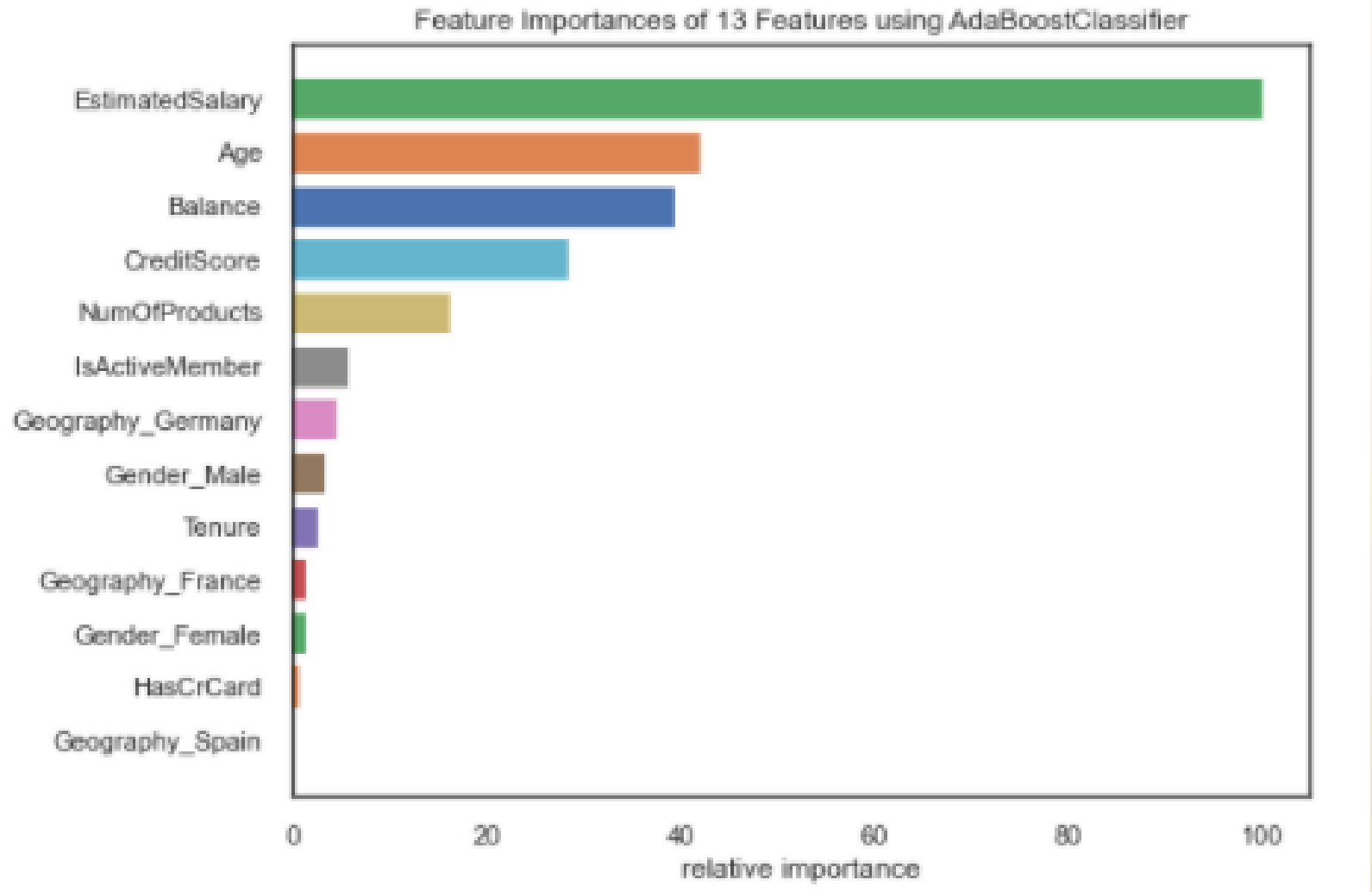
Random Forest Classifier

i. There is no improvement in metric results after tuning and all the results average have dropped 0.01.
ii. Poor recall metric results although it has the lowest log loss.

LGBM Classifier

i. In overall of LGMB metric results, it does not have much improvement as compared to other 2 models.
ii. Log loss have increased.

Feature Importance (AdaBoost Classifier)



Insights/Recommendation:

1. Based on the feature importance, we know that Age, Estimated Salary and the balance of the money customers deposited into the bank will affect the customer to exit the bank or not.
   - Target on the customers who have frequent withdrawal from the bank or monthly regular deposit not reflected for a period, i.e. Salary.
   - To have some client recovery services to keep the customers and get direct feedback from customers.

2. The bank may also focused on those customers that are in the age range to be churn that have shown in the chart and offering better incentives to them in order to minimize churn and keep more customers.

Post Review

- ***What would you have done differently if you were to do the project again?***
  - Understand the important of the feature
  - Train the feature separately
  - Improve the tuning part and have the best metric results

- ***What are some other things that you want to try in the next iteration?***
  - Perform other module or method that have for machine learning which is not included in our course, i.e. SHAP.
  - To summarize the steps and plotting chart for easier reference and efficiency, i.e. plot_count, using function or for loop to run most of the report.

Appendix

Dataset

- https://www.kaggle.com/santoshd3/bank-customers

GitHub

- https://github.com/aureliapy/bank_customer_churn_prediction

THANK YOU