BANK
CUSTOMER CHURN

MindX TC_DA19
Nguyễn Minh Phát

AGENDA

Introduction

Exploratory Data Analysis (EDA)

Data Processing

Model Training and Evaluation

- Before Up-sampling Data
- After Up-sampling Data

Conclusion

References

INTRODUCTION

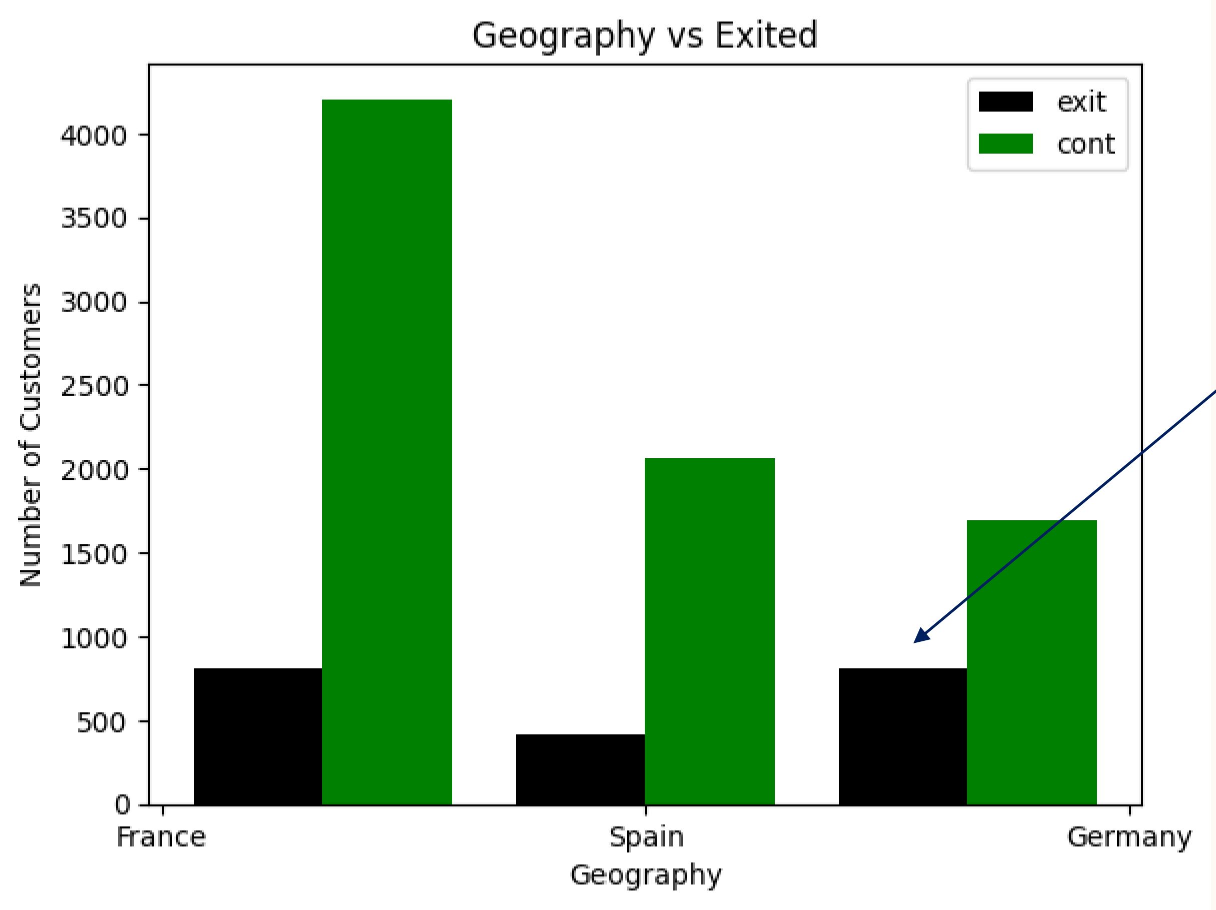
The objective of the bank customer churn dataset is to predict whether customers will continue to use the bank's services or not.

The dataset has 14 columns, 10000 observations.

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	Yes	Yes	101348.88	Yes
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	No	Yes	112542.58	No
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	Yes	No	113931.57	Yes

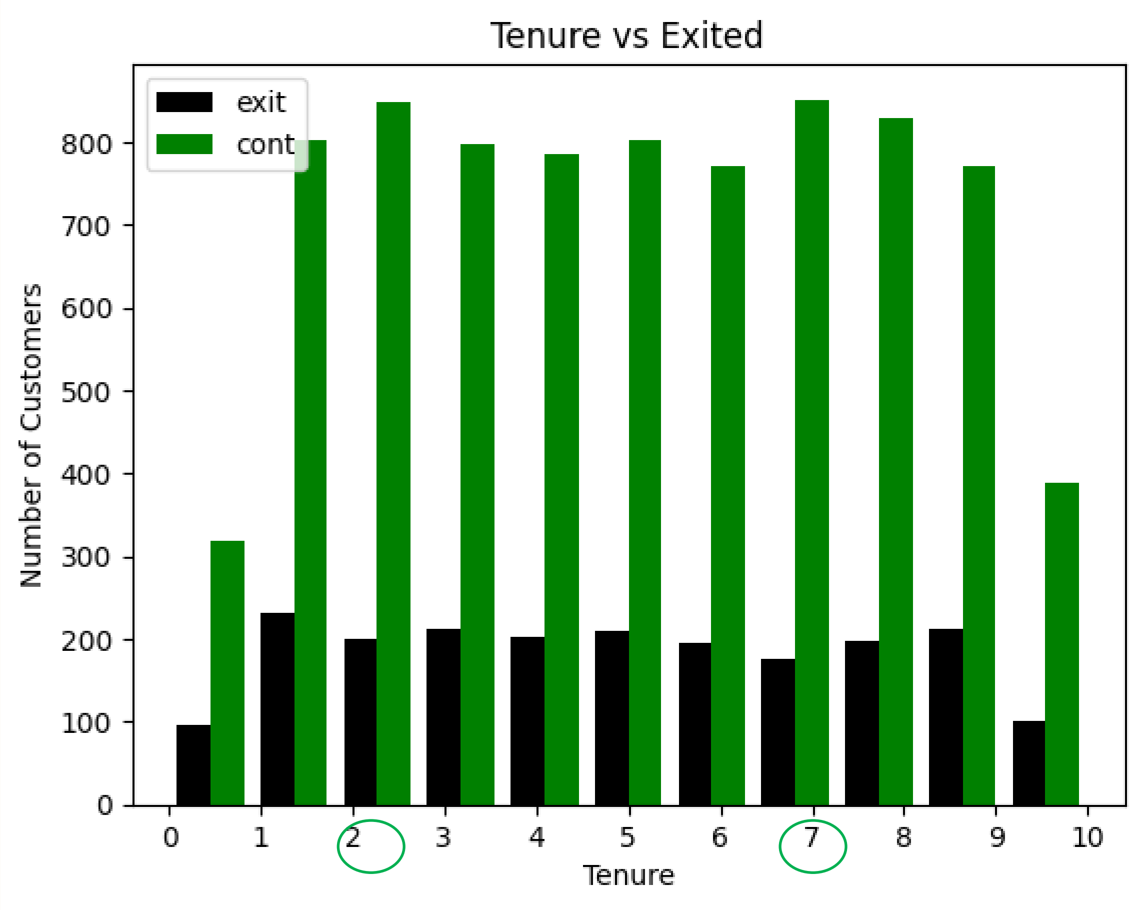
Predict

EXPLORATORY DATA ANALYSIS



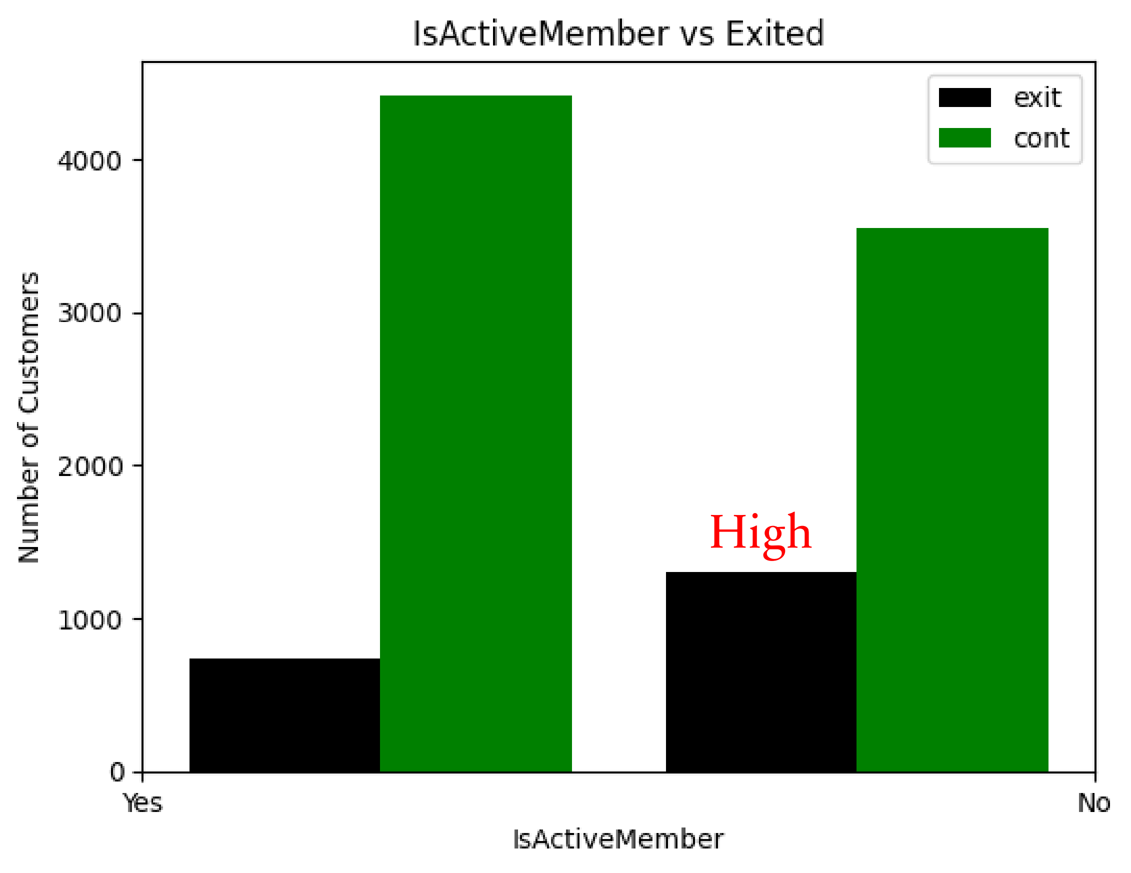
- Geography: France , Spain, Germany
- Most of the customers are from France
- In Germany, the number of customers leaving accounts for the highest percentage

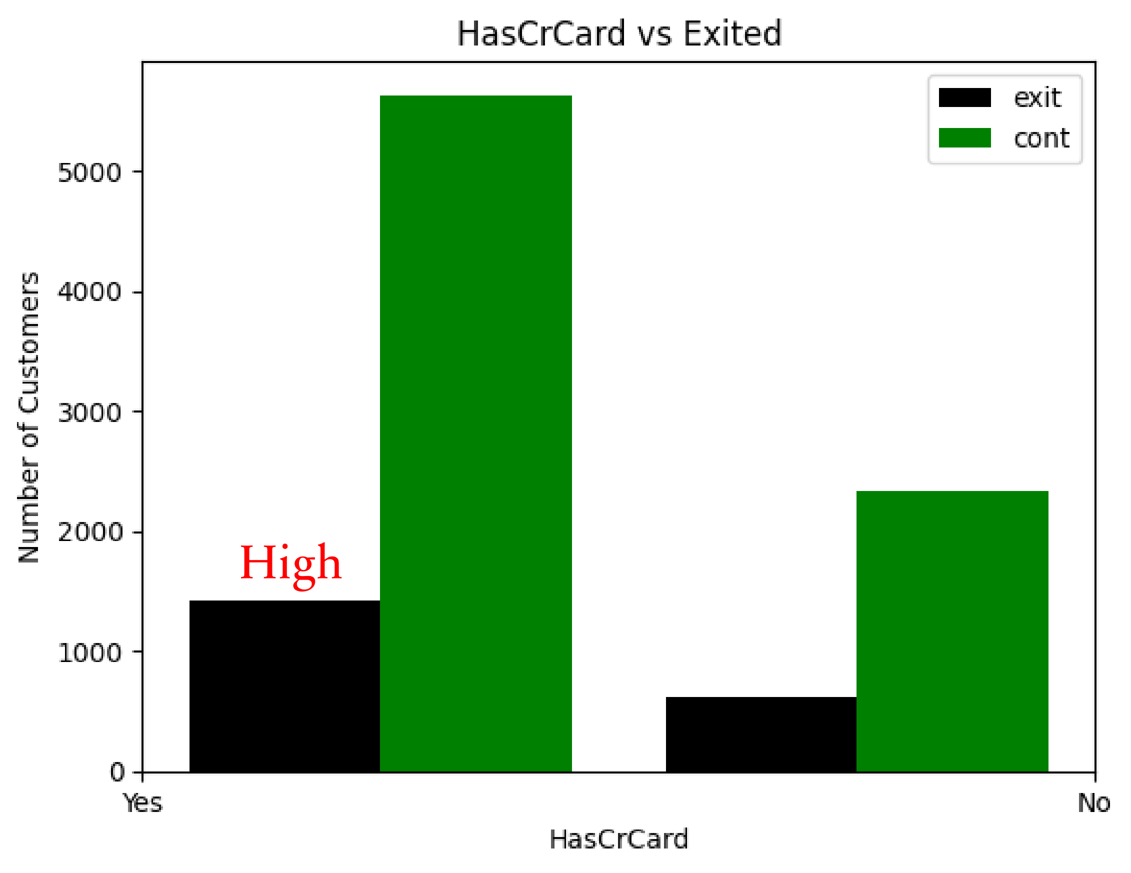
EXPLORATORY DATA ANALYSIS



- The majority of customer tenure in the data set is from 1 to 9 months.
- The number of customers who stay with the term of 2 months and 7 months account for the most.

EXPLORATORY DATA ANALYSIS





DATA PROCESSING

Drop columns

	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0.00	1	Yes	Yes	101348.88	Yes
1	2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	No	Yes	112542.58	No
2	3	15619304	Onio	502	France	Female	42	8	159660.80	3	Yes	No	113931.57	Yes

Result

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	Yes	Yes	101348.88	Yes
1	608	Spain	Female	41	1	83807.86	1	No	Yes	112542.58	No
2	502	France	Female	42	8	159660.80	3	Yes	No	113931.57	Yes

DATA PROCESSING

Replace Binary Values (Female = 1, Male = 0) (Yes = 1, No = 0)

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	Yes	Yes	101348.88	Yes
1	608	Spain	Female	41	1	83807.86	1	No	Yes	112542.58	No
2	502	France	Female	42	8	159660.80	3	Yes	No	113931.57	Yes

Result

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	1	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	1	41	1	83807.86	1	0	1	112542.58	0
2	502	France	1	42	8	159660.80	3	1	0	113931.57	1

DATA PROCESSING

One-hot Encoding (get dummies)

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	Yes	Yes	101348.88	Yes
1	608	Spain	Female	41	1	83807.86	1	No	Yes	112542.58	No
2	502	France	Female	42	8	159660.80	3	Yes	No	113931.57	Yes

Result

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
0	619	1	42	2	0.00	1	1	1	101348.88	1	1	0	0
1	608	1	41	1	83807.86	1	0	1	112542.58	0	0	0	1
2	502	1	42	8	159660.80	3	1	0	113931.57	1	1	0	0

DATA PROCESSING

Scale columns to range(0, 1) (Min Max Scaler)

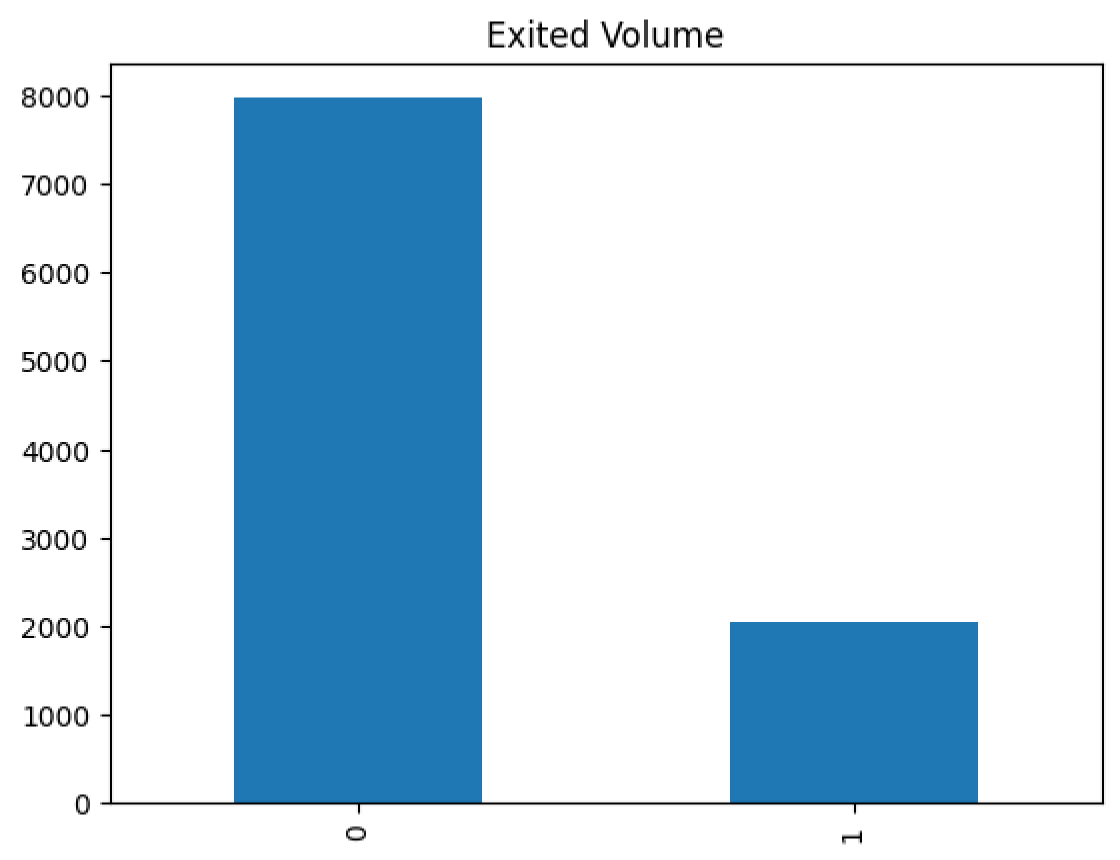
	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
0	619	1	42	2	0.00	1	1	1	101348.88	1	1	0	0
1	608	1	41	1	83807.86	1	0	1	112542.58	0	0	0	1
2	502	1	42	8	159660.80	3	1	0	113931.57	1	1	0	0

Result

	CreditScore	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited	Geography_France	Geography_Germany	Geography_Spain
0	0.538	1	0.324324	0.2	0.000000	0.000000	1	1	0.506735	1	1	0	0
1	0.516	1	0.310811	0.1	0.334031	0.000000	0	1	0.562709	0	0	0	1
2	0.304	1	0.324324	0.8	0.636357	0.666667	1	0	0.569654	1	1	0	0

MODEL TRAINING AND EVALUATION

y



Churn dataset got

Imbalanced data

➔ Up-sampling ???

MODEL TRAINING – BEFORE UP-SAMPLING

Input

```
X_train shape: (8000, 12)
X_test shape: (2000, 12)
y_train shape: (8000,)
y_test shape: (2000,)
```

Models Score and Time Executing

	Name	Score(%)	Time (s)
0	LogisticRegression	81.25	117.8
1	GaussianNB	82.8	17.9
2	SVC	84.45	974.0
3	RandomForestClassifier	86.55	134.5
4	DecisionTreeClassifier	78.3	47.8
5	BaggingClassifier	81.15	476.8
6	AdaBoostClassifier	81.4	797.5
7	GradientBoostingClassifier	86.5	366.2
8	XGBClassifier	86.1	458.2

- Run some models with default hyperparameters
- XGBoost and Random Forest model for following training

MODEL TRAINING – BEFORE UP-SAMPLING

Random Forest

	precision	recall	f1-score	support
0	0.88	0.96	0.92	1607
1	0.76	0.49	0.59	393
accuracy			0.87	2000
macro avg	0.82	0.72	0.76	2000
weighted avg	0.86	0.87	0.86	2000

XGBoost

	precision	recall	f1-score	support
0	0.89	0.95	0.92	1607
1	0.70	0.51	0.59	393
accuracy			0.86	2000
macro avg	0.80	0.73	0.75	2000
weighted avg	0.85	0.86	0.85	2000

Low Score in Precision, Recall, F1-Score of 1

➔ Try Up-sampling data

MODEL TRAINING – AFTER UP-SAMPLING

```
X_train shape: (8000, 12)
X_test shape: (2000, 12)
y_train shape: (8000,)
y_test shape: (2000,)
```

SMOTE

```
X_SM_train shape: (12740, 12)
X_SM_test shape: (3186, 12)
y_SM_train shape: (12740,)
y_SM_test shape: (3186,)
```

Input

Models Score and Time Executing

	Name	Score(%)	Time (s)
0	LogisticRegression	71.44	69.2
1	GaussianNB	71.34	8.4
2	SVC	78.28	172.8
3	RandomForestClassifier	90.08	548.2
4	DecisionTreeClassifier	82.74	68.0
5	BaggingClassifier	71.44	90.7
6	AdaBoostClassifier	66.67	574.5
7	GradientBoostingClassifier	87.1	3.9
8	XGBClassifier	90.96	256.9

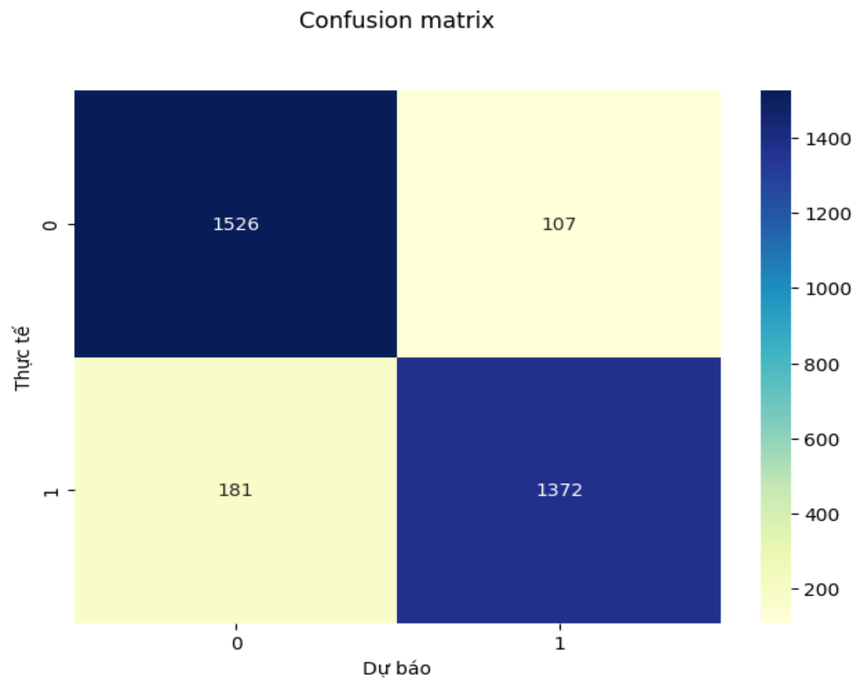
- Run some models with default hyperparameters
- Some models have increased score up to 90%
- XGBoost and Random Forest model for following training

MODEL EVALUATION – AFTER UP-SAMPLING

XGBoost - Default

	precision	recall	f1-score	support
0	0.89	0.93	0.91	1633
1	0.93	0.88	0.91	1553
accuracy			0.91	3186
macro avg	0.91	0.91	0.91	3186
weighted avg	0.91	0.91	0.91	3186

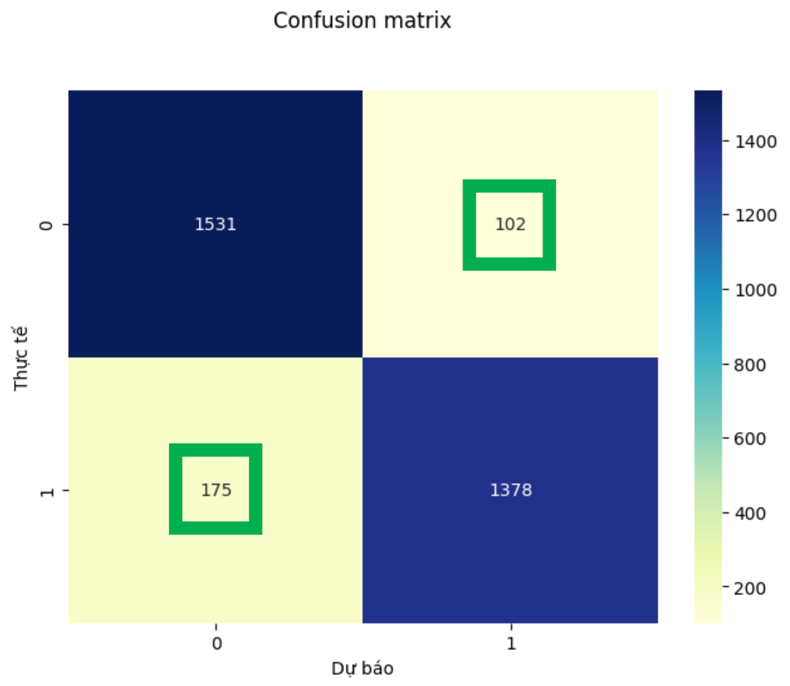
XGBoost score: 0.9096045197740112



XGBoost – Hyperparameter Tuning

Best parameters found: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 500}
Best accuracy score: 0.9040816326530612

	precision	recall	f1-score	support
0	0.90	0.94	0.92	1633
1	0.93	0.89	0.91	1553
accuracy			0.91	3186
macro avg	0.91	0.91	0.91	3186
weighted avg	0.91	0.91	0.91	3186



MODEL EVALUATION – AFTER UP-SAMPLING

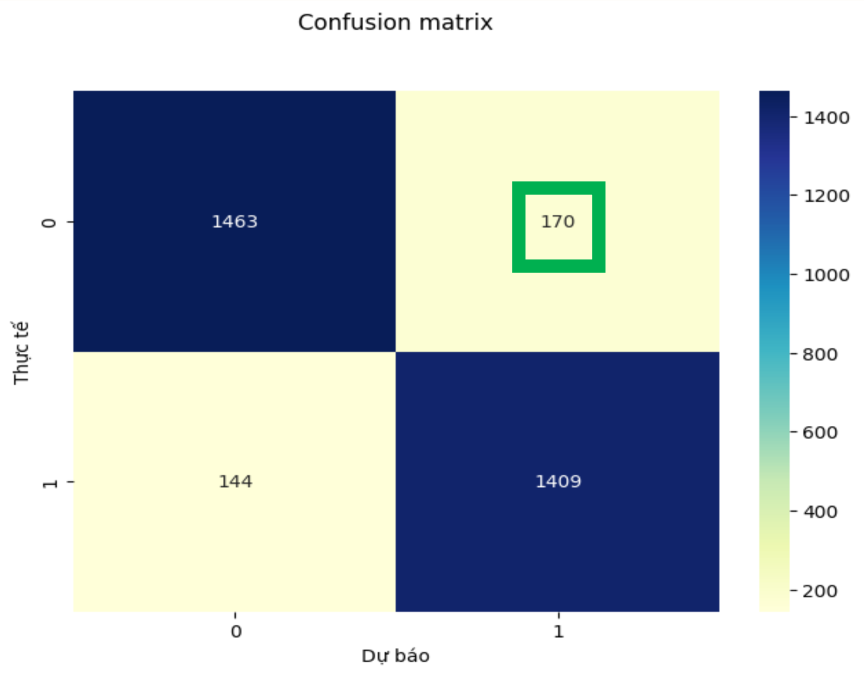
Random Forest - Default

```
              precision    recall  f1-score   support

           0       0.91      0.90      0.90      1633
           1       0.89      0.91      0.90      1553

    accuracy                           0.90      3186
   macro avg       0.90      0.90      0.90      3186
weighted avg       0.90      0.90      0.90      3186

0.901443816698054
```

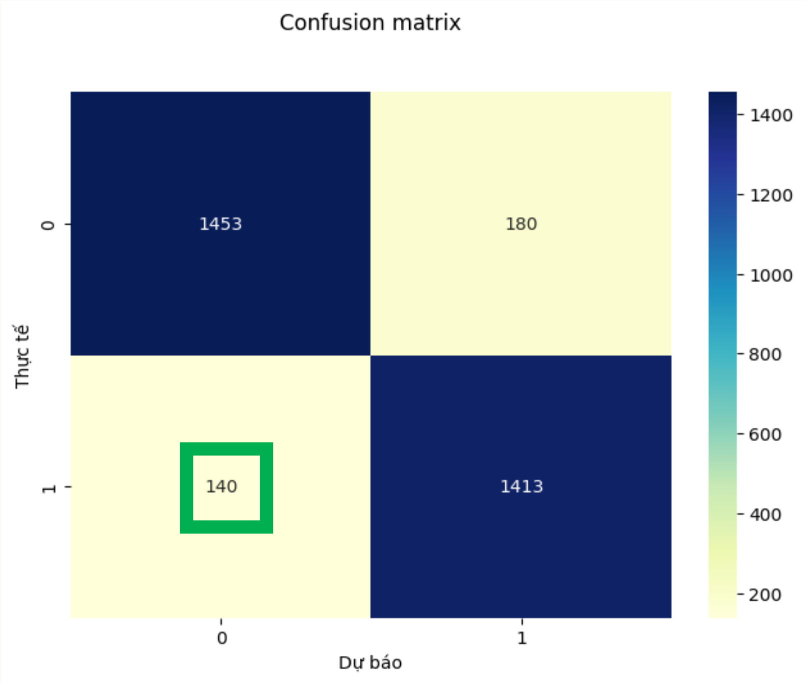


Random Forest – Hyperparameter Tuning

```
Best parameters found:  {'criterion': 'gini', 'min_samples_leaf': 1, 'min_samples_split': 2,
                                                                    'n_estimators': 200}
Best accuracy score:  0.8914442700156986
              precision    recall  f1-score   support

           0       0.91      0.89      0.90      1633
           1       0.89      0.91      0.90      1553

    accuracy                           0.90      3186
   macro avg       0.90      0.90      0.90      3186
weighted avg       0.90      0.90      0.90      3186
```



MODEL EVALUATION – FINAL JUDGING

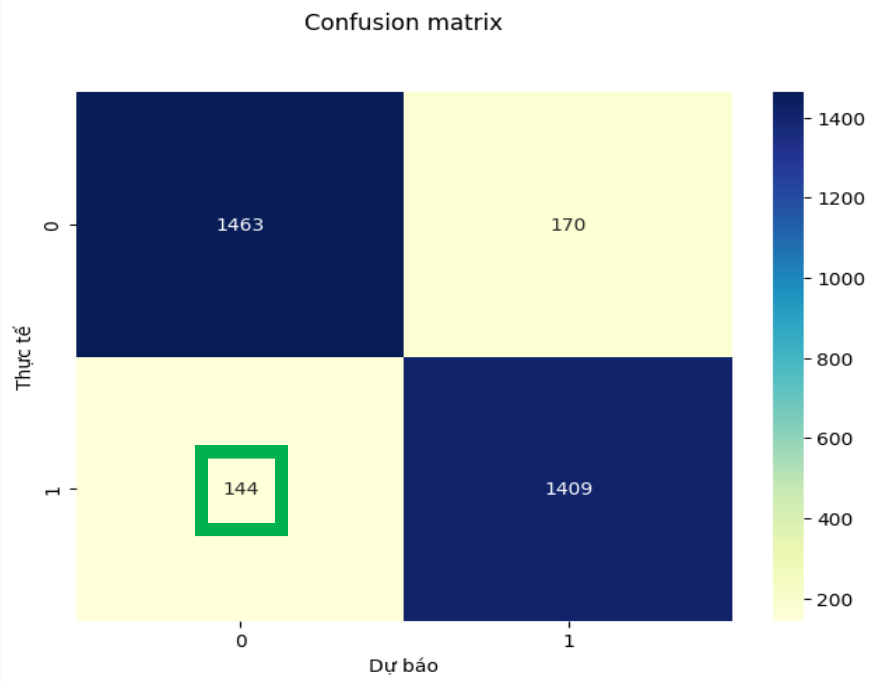
Random Forest - Default

```
              precision    recall  f1-score   support

           0       0.91      0.90      0.90      1633
           1       0.89      0.91      0.90      1553

    accuracy                           0.90      3186
   macro avg       0.90      0.90      0.90      3186
weighted avg       0.90      0.90      0.90      3186

0.901443816698054
```

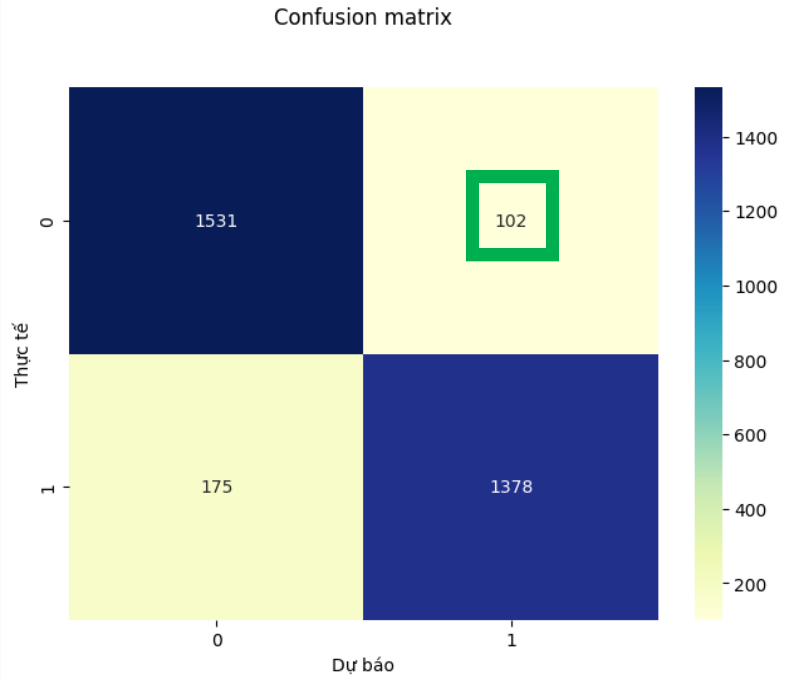


XGBoost – Hyperparameter Tuning

```
Best parameters found:  {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 500}
Best accuracy score:  0.9040816326530612
              precision    recall  f1-score   support

           0       0.90      0.94      0.92      1633
           1       0.93      0.89      0.91      1553

    accuracy                           0.91      3186
   macro avg       0.91      0.91      0.91      3186
weighted avg       0.91      0.91      0.91      3186
```



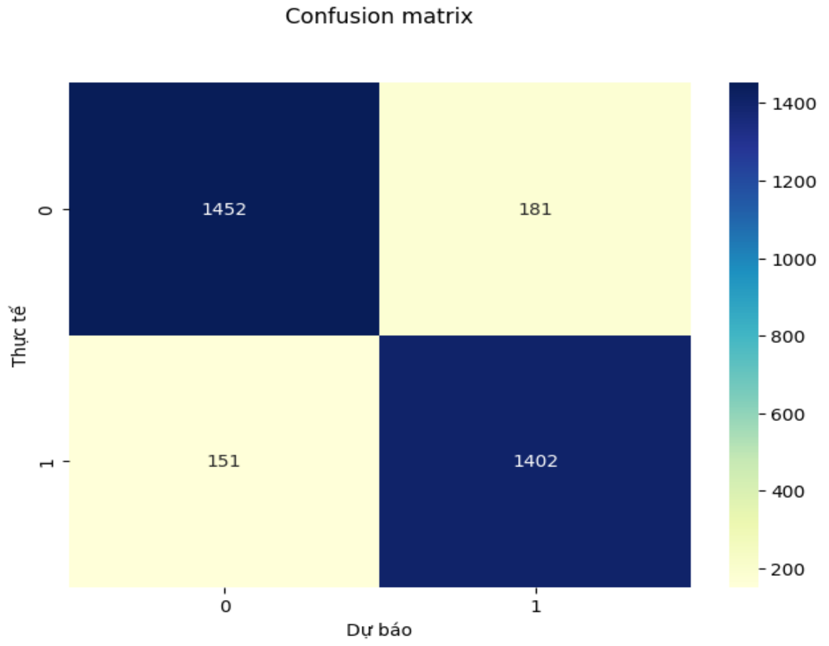
OPTION: MODEL TRAINING – DEEP LEARNING (NEURAL NETWORK)

Models Configuration

```
model = keras.Sequential([
    keras.layers.Dense(200, input_shape=(12,), activation='relu'),
    keras.layers.Dense(100, activation='relu'),
    keras.layers.Dense(50, activation='relu'),
    keras.layers.Dense(25, activation='relu'),
    keras.layers.Dense(1, activation='sigmoid')
])
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.fit(X_SM_train, y_SM_train, epochs=194, batch_size=32)
```

Result

	precision	recall	f1-score	support
0	0.91	0.89	0.90	1633
1	0.89	0.90	0.89	1553
accuracy			0.90	3186
macro avg	0.90	0.90	0.90	3186
weighted avg	0.90	0.90	0.90	3186



Score is lower than Tuned-XGBoost model ←

CONCLUSION

- Processing imbalanced data is important for classification tasks because it can improve the performance of the model by reducing the impact of the class imbalance on the training process.
- XGBoost is a highly effective algorithm for classification problem.
- Neural Network is a promising approach to work with complex dataset.

REFERENCES

DATASET

https://www.kaggle.com/datasets/barelydedicated/bank-customer-churn-modeling

YOUTUBE REFERENCE

https://www.youtube.com/watch?v=MSBY28IJ47U&ab_channel=codebasics

CHAT-GPT

THANK YOU