Data Analyst:: Cross selling recommendation

Team member's details

Group Name: Ctrl_C&Ctrl_V

Name:

1. Yihsuan Sun
2. Tiantian Xie

Email:

- Yihsuan S.: yihsuan.sun88@gmail.com
- Tiantian X.: tenxie0411@gmail.com

Country: United States

College/Company:

- Yihsuan S.: Washington State University
- Tiantian X.: Case Western Reserve University

Specialization: Data Analyst

Problem description & Business understanding

XYZ Credit Union, located in Latin America, does well in selling banking products such as: credit cards, deposit accounts, retirement accounts, safe deposit boxes, etc. However, after statistics, they found that their existing customers basically only buy one product, which means that the bank does not perform well in cross-selling. So XYZ Credit Union wants analysts to build models such as marketing models through machine learning to solve their problems**.**

Project lifecycle

Week 7	**Problem description & Business understanding**
Week 8	**Data Cleansing and Transformation**
Week 9	**Exploratory data analysis**

Week 10	**Exploratory data analysis**
Week 11	**EDA presentation**
Week 12	**Model Selection and Model Building**
Week 13	**Final Project Report, Code, and presentation**

Github Repo link: https://github.com/Yihsuansun/Cross_selling_recommendation.git

Data Intake Report

Name: Cross selling recommendation
Report Date: 06/17/2022
Internship Batch: LISUM09
Version: <1.0>
Data intake by: Data Glacier Virtual Internship 2022
Data intake reviewer:
Data storage location:
https://drive.google.com/file/d/16-nzZR91-ijrfjUcI2PniTpOgrvFAykA/view

Test.csv details:

Total number of observations	929,615
Total number of files	1
Total number of features	24
Base format of the file	csv
Size of the data	105 MB

Train.csv details:

Total number of observations	13,647,309
Total number of files	1
Total number of features	48
Base format of the file	csv

Size of the data	2.13 GB

EDA:

https://github.com/Yihsuansun/Cross_selling_recommendation/tree/main/Week%2011

Final result:

Random Forest Classifiers provide the best result.

In [35]:
```
#linear model
model_linear = LogisticRegression()
model_linear.fit(X_train,Y_train)
performance_met(model_linear,X_train,Y_train,X_test,Y_test)
```

```
train score: accuracy:0.7277077614023484 f1:0.0
test score: accuracy:0.727239253403882 f1:0.0
```

In [36]:
```
#ensemble model
model_ensemble= RandomForestClassifier(n_estimators = 20,max_depth=20,n_jobs=-1)
model_ensemble.fit(X_train,Y_train)
performance_met(model_ensemble,X_train,Y_train,X_test,Y_test)
```

```
train score: accuracy:0.8065376972789199 f1:0.6592528161897644
test score: accuracy:0.8054309267979541 f1:0.6577298991986077
```

In [37]:
```
#boosting model
model_boosting = AdaBoostClassifier()
model_boosting.fit(X_train,Y_train)
performance_met(model_boosting,X_train,Y_train,X_test,Y_test)
```

```
train score: accuracy:0.7807272617377347 f1:0.6282568367905806
test score: accuracy:0.7807176115371415 f1:0.6286158045841765
```