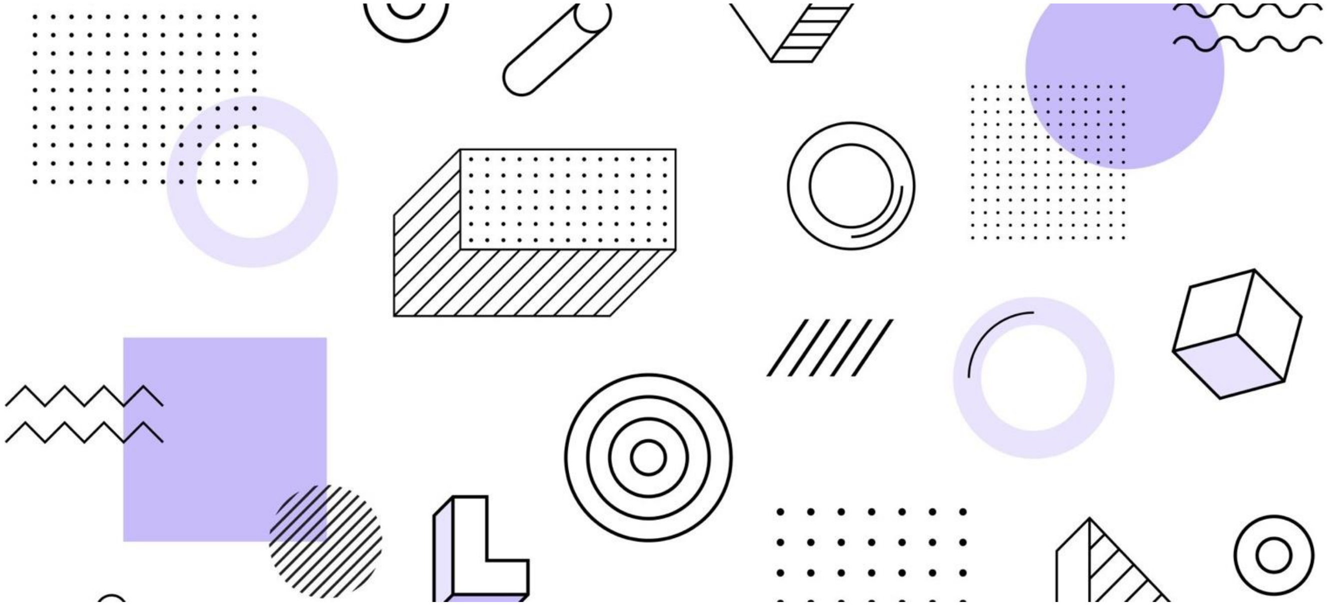
Cross-Sell Prediction Hackathon Classification Problem



Prepared by Mohamed Suhaib

Problem Statement

- Your client is an Insurance company that has provided Health Insurance to its customers now they need your help in building a model to predict whether the policyholders (customers) from past year will also be interested in Vehicle Insurance provided by the company.
- An insurance policy is an arrangement by which a company undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for the payment of a specified premium. A premium is a sum of money that the customer needs to pay regularly to an insurance company for this guarantee.
- Building a model to predict whether a customer would be interested in Vehicle Insurance is extremely helpful for the company because it can then accordingly plan its communication strategy to reach out to those customers and optimise its business model and revenue.
- Now, in order to predict, whether the customer would be interested in Vehicle insurance, you have information about demographics (gender, age, region code type), Vehicles (Vehicle Age, Damage), Policy (Premium, sourcing channel) etc.

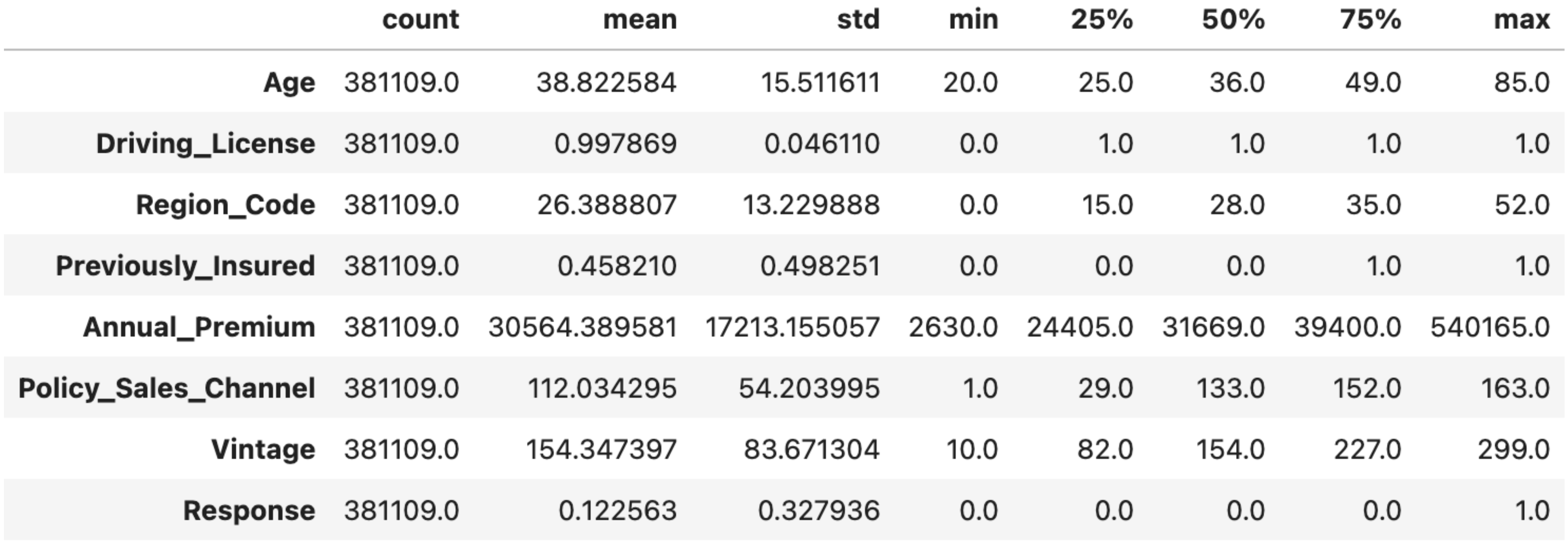
Scope of this project

- Using this model, will try to predict whether a Probability of Customer being interested in Vehicle Loan in the test set will be **1 : Customer is interested, 0 : Customer is not interested** after the evaluation process.

Data set provided

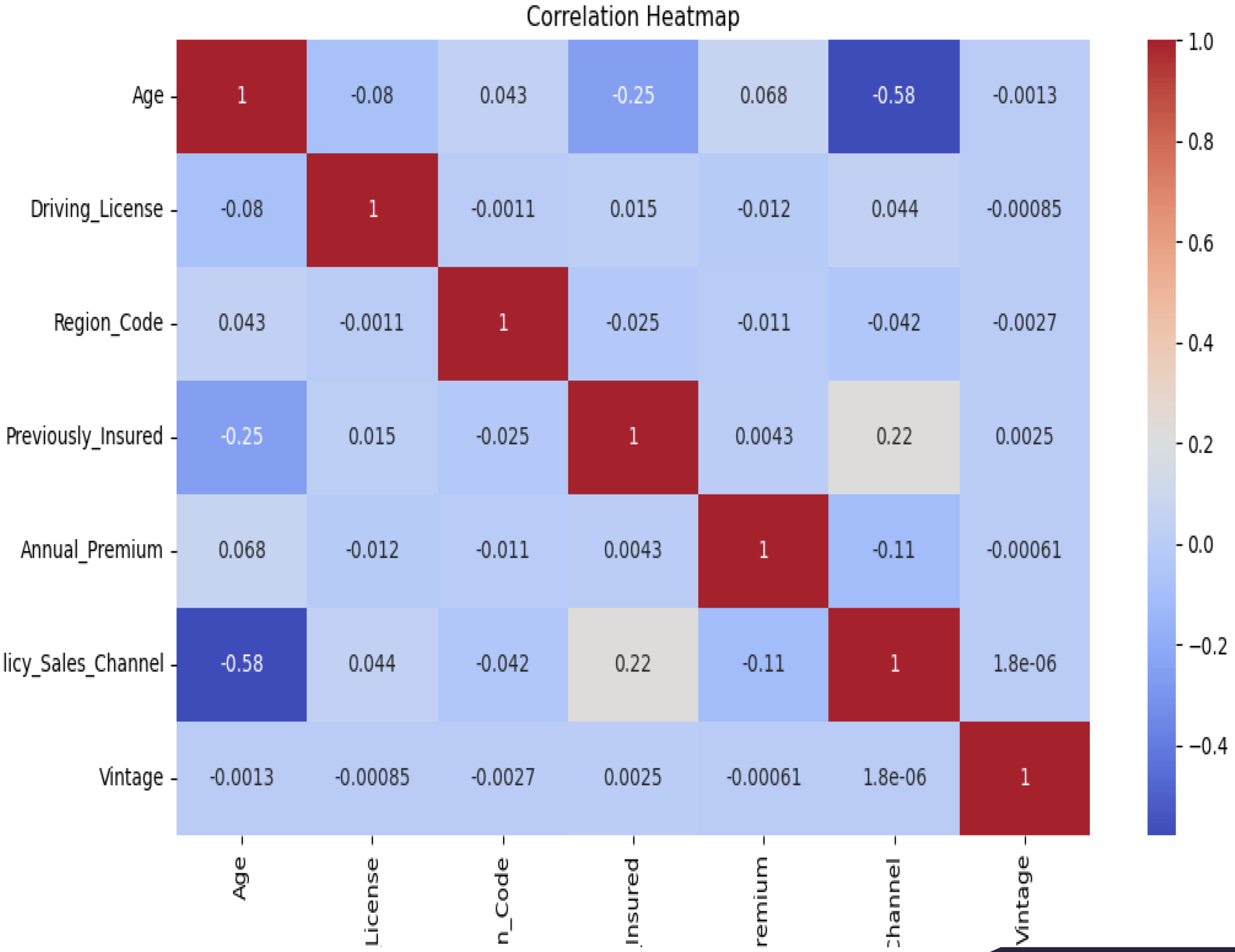
- train.csv - 381109 records with 12 columns
- test.csv - 127037 records with 11 columns

Observation:



	count	mean	std	min	25%	50%	75%	max
Age	381109.0	38.822584	15.511611	20.0	25.0	36.0	49.0	85.0
Driving_License	381109.0	0.997869	0.046110	0.0	1.0	1.0	1.0	1.0
Region_Code	381109.0	26.388807	13.229888	0.0	15.0	28.0	35.0	52.0
Previously_Insured	381109.0	0.458210	0.498251	0.0	0.0	0.0	1.0	1.0
Annual_Premium	381109.0	30564.389581	17213.155057	2630.0	24405.0	31669.0	39400.0	540165.0
Policy_Sales_Channel	381109.0	112.034295	54.203995	1.0	29.0	133.0	152.0	163.0
Vintage	381109.0	154.347397	83.671304	10.0	82.0	154.0	227.0	299.0
Response	381109.0	0.122563	0.327936	0.0	0.0	0.0	0.0	1.0

Correlation Heatmap



Correlation Heatmap

	Age	License	n_Code	Insured	remium	hannel	Vintage
Age	1	-0.08	0.043	-0.25	0.068	-0.58	-0.0013
Driving_License	-0.08	1	-0.0011	0.015	-0.012	0.044	-0.00085
Region_Code	0.043	-0.0011	1	-0.025	-0.011	-0.042	-0.0027
Previously_Insured	-0.25	0.015	-0.025	1	0.0043	0.22	0.0025
Annual_Premium	0.068	-0.012	-0.011	0.0043	1	-0.11	-0.00061
licy_Sales_Channel	-0.58	0.044	-0.042	0.22	-0.11	1	1.8e-06
Vintage	-0.0013	-0.00085	-0.0027	0.0025	-0.00061	1.8e-06	1

category and numeric features

Categorical Columns:

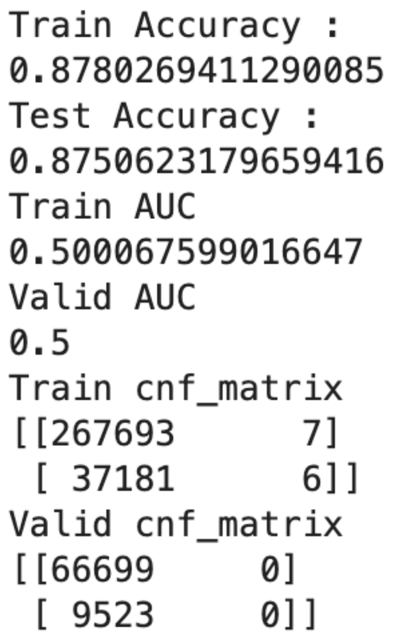
- ['Gender', 'Vehicle_Age', 'Vehicle_Damage']

Numerical Columns:

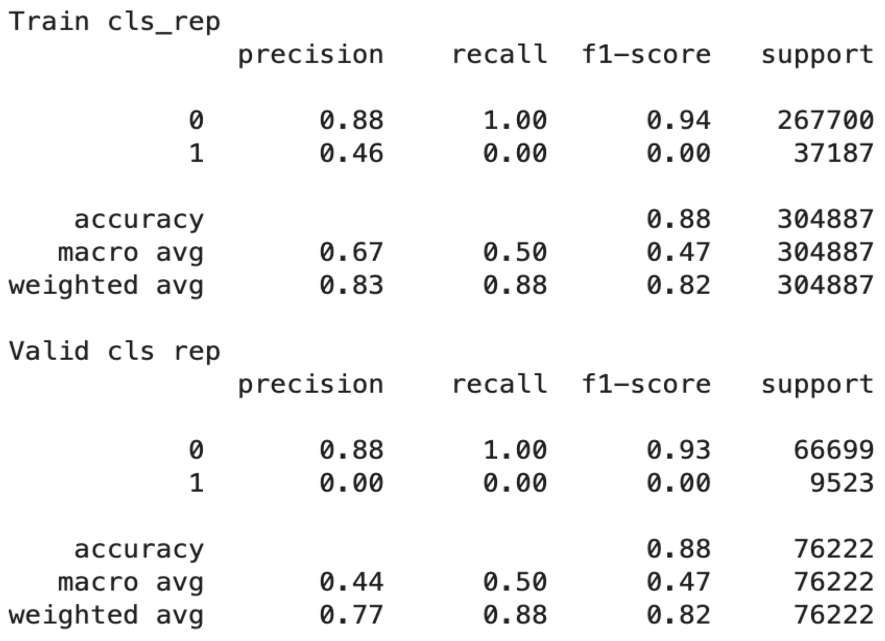
- ['Age', 'Driving_License', 'Region_Code', 'Previously_Insured', 'Annual_Premium', 'Policy_Sales_Channel', 'Vintage']

Model Prediction Using Base Model:

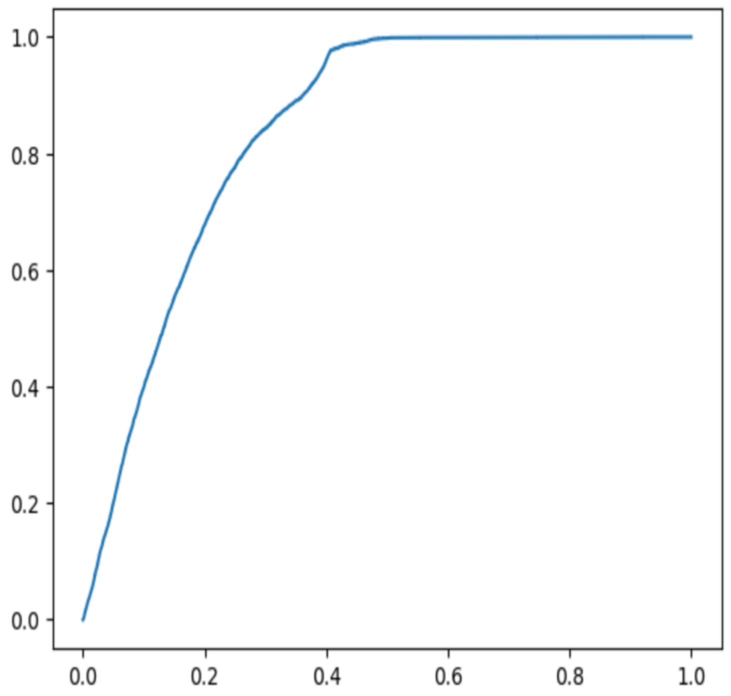
Using Logistic Regression:



```
Train Accuracy :
0.8780269411290085
Test Accuracy :
0.8750623179659416
Train AUC
0.500067599016647
Valid AUC
0.5
Train cnf_matrix
[[267693      7]
 [ 37181      6]]
Valid cnf_matrix
[[66699     0]
 [ 9523     0]]
```



```
Train cls_rep
              precision    recall  f1-score   support

           0       0.88      1.00      0.94    267700
           1       0.46      0.00      0.00     37187

    accuracy                           0.88    304887
   macro avg       0.67      0.50      0.47    304887
weighted avg       0.83      0.88      0.82    304887

Valid cls rep
              precision    recall  f1-score   support

           0       0.88      1.00      0.93     66699
           1       0.00      0.00      0.00      9523

    accuracy                           0.88     76222
   macro avg       0.44      0.50      0.47     76222
weighted avg       0.77      0.88      0.82     76222
```

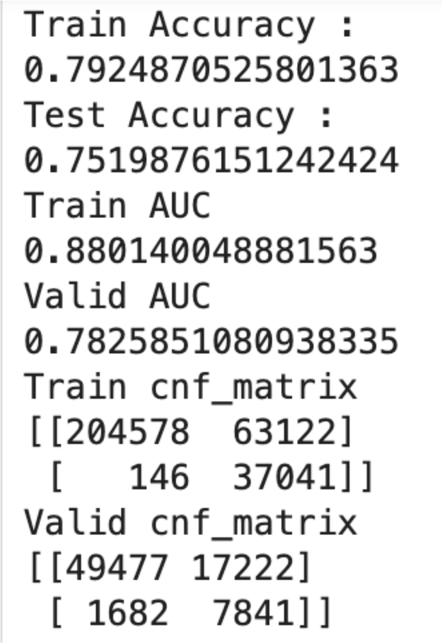


Train ROC_AUC Score 0.500067599016647

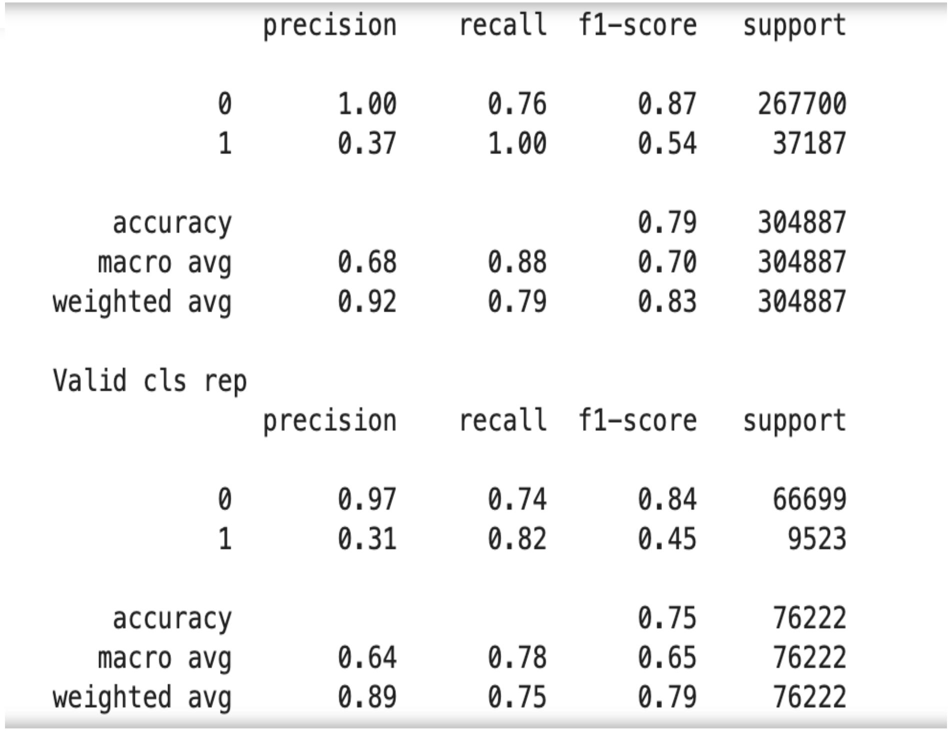
Test ROC_AUC Score 0.5

Model Comparison:

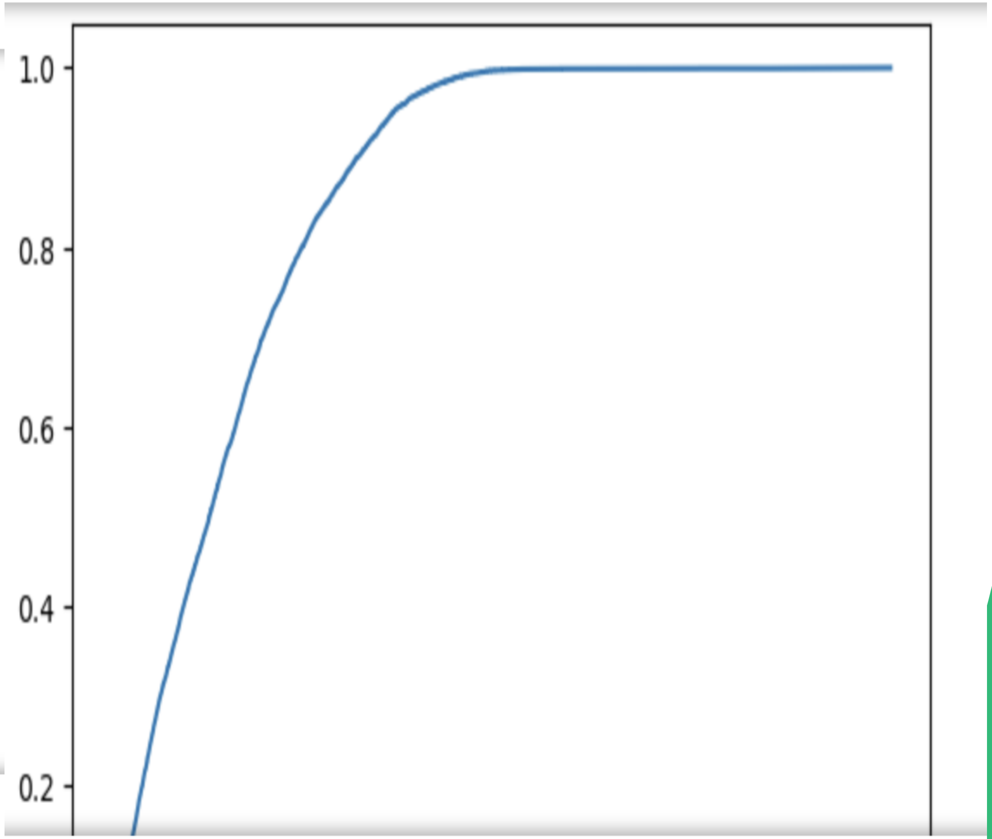
Using RandomForestClassifier:



```
Train Accuracy :
0.7924870525801363
Test Accuracy :
0.7519876151242424
Train AUC
0.880140048881563
Valid AUC
0.7825851080938335
Train cnf_matrix
[[204578  63122]
 [   146  37041]]
Valid cnf_matrix
[[49477 17222]
 [ 1682  7841]]
```

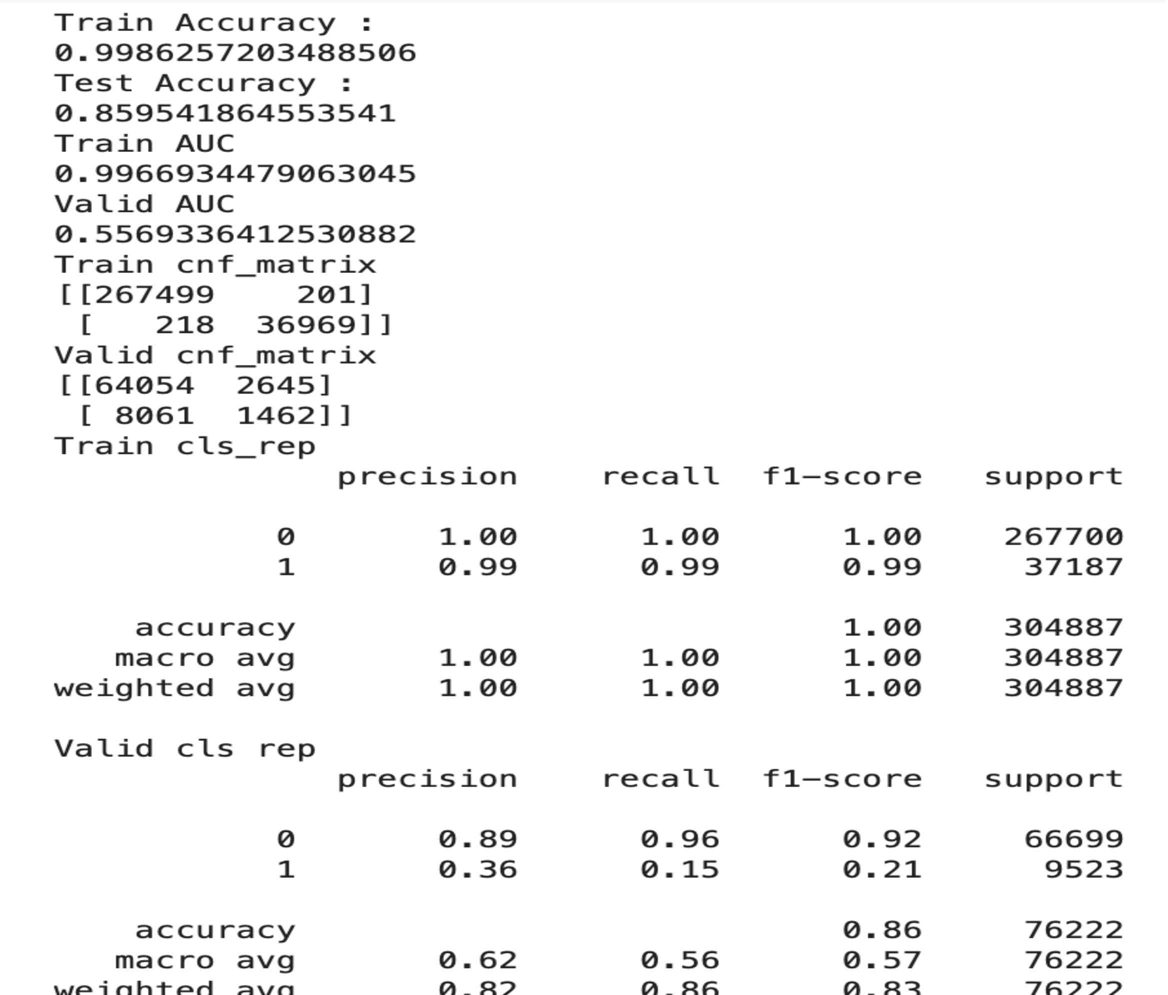


```
              precision    recall  f1-score   support

           0       1.00      0.76      0.87    267700
           1       0.37      1.00      0.54     37187

    accuracy                           0.79    304887
   macro avg       0.68      0.88      0.70    304887
weighted avg       0.92      0.79      0.83    304887

Valid cls rep
              precision    recall  f1-score   support

           0       0.97      0.74      0.84     66699
           1       0.31      0.82      0.45      9523

    accuracy                           0.75     76222
   macro avg       0.64      0.78      0.65     76222
weighted avg       0.89      0.75      0.79     76222
```

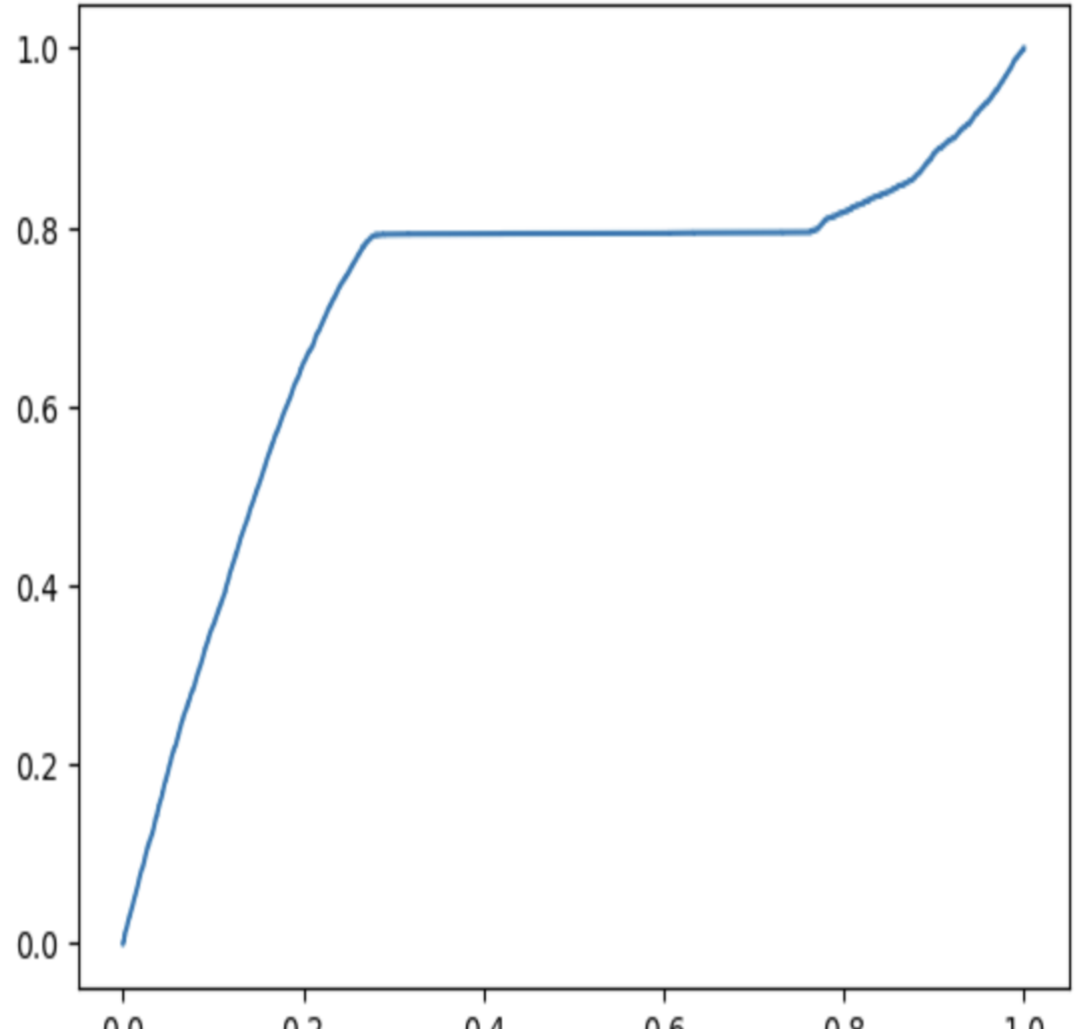


Hyperparameters Tuning for each model

Using StackingClassifier & GradientBoostingClassifier & XGBClassifier & AdaboostClassifier:

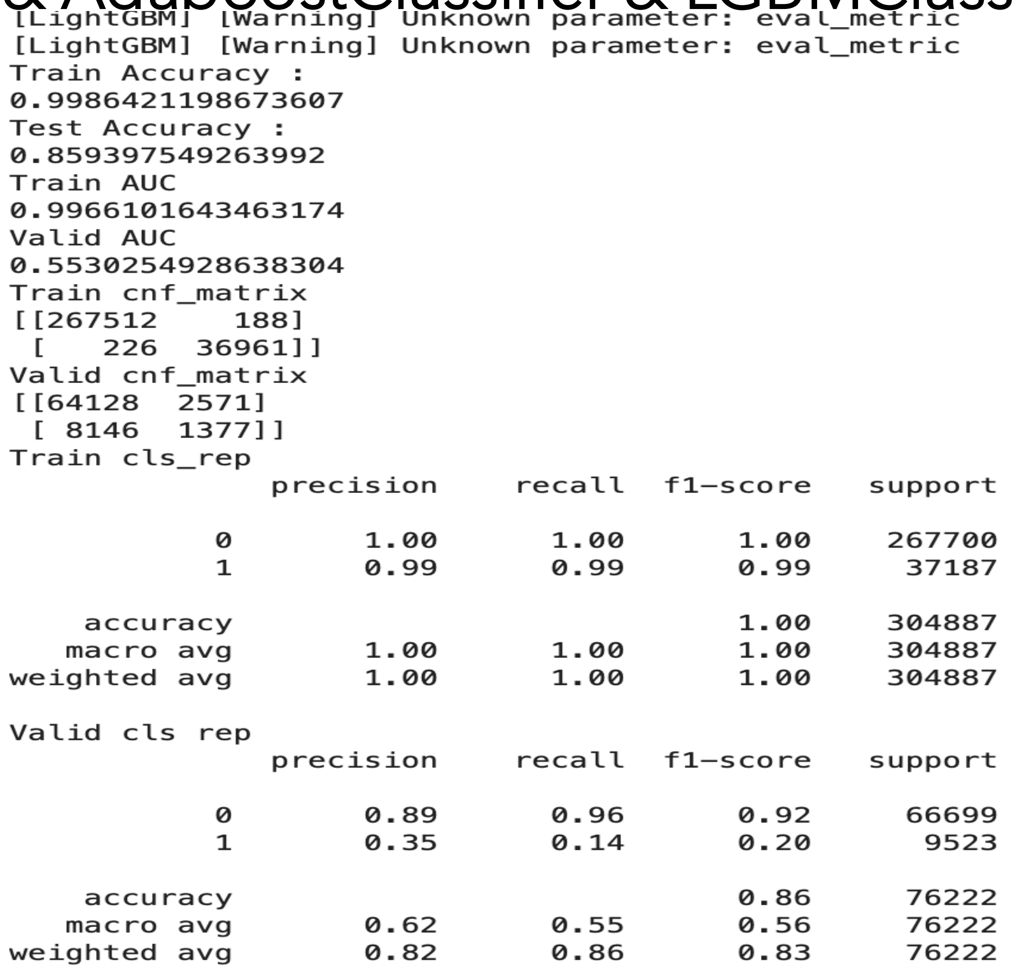


```
Train Accuracy :
0.9986257203488506
Test Accuracy :
0.859541864553541
Train AUC
0.9966934479063045
Valid AUC
0.5569336412530882
Train cnf_matrix
[[267499    201]
 [   218  36969]]
Valid cnf_matrix
[[64054  2645]
 [ 8061  1462]]
Train cls_rep
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    267700
           1       0.99      0.99      0.99     37187

    accuracy                           1.00    304887
   macro avg       1.00      1.00      1.00    304887
weighted avg       1.00      1.00      1.00    304887

Valid cls rep
              precision    recall  f1-score   support

           0       0.89      0.96      0.92     66699
           1       0.36      0.15      0.21      9523

    accuracy                           0.86     76222
   macro avg       0.62      0.56      0.57     76222
weighted avg       0.82      0.86      0.83     76222
```

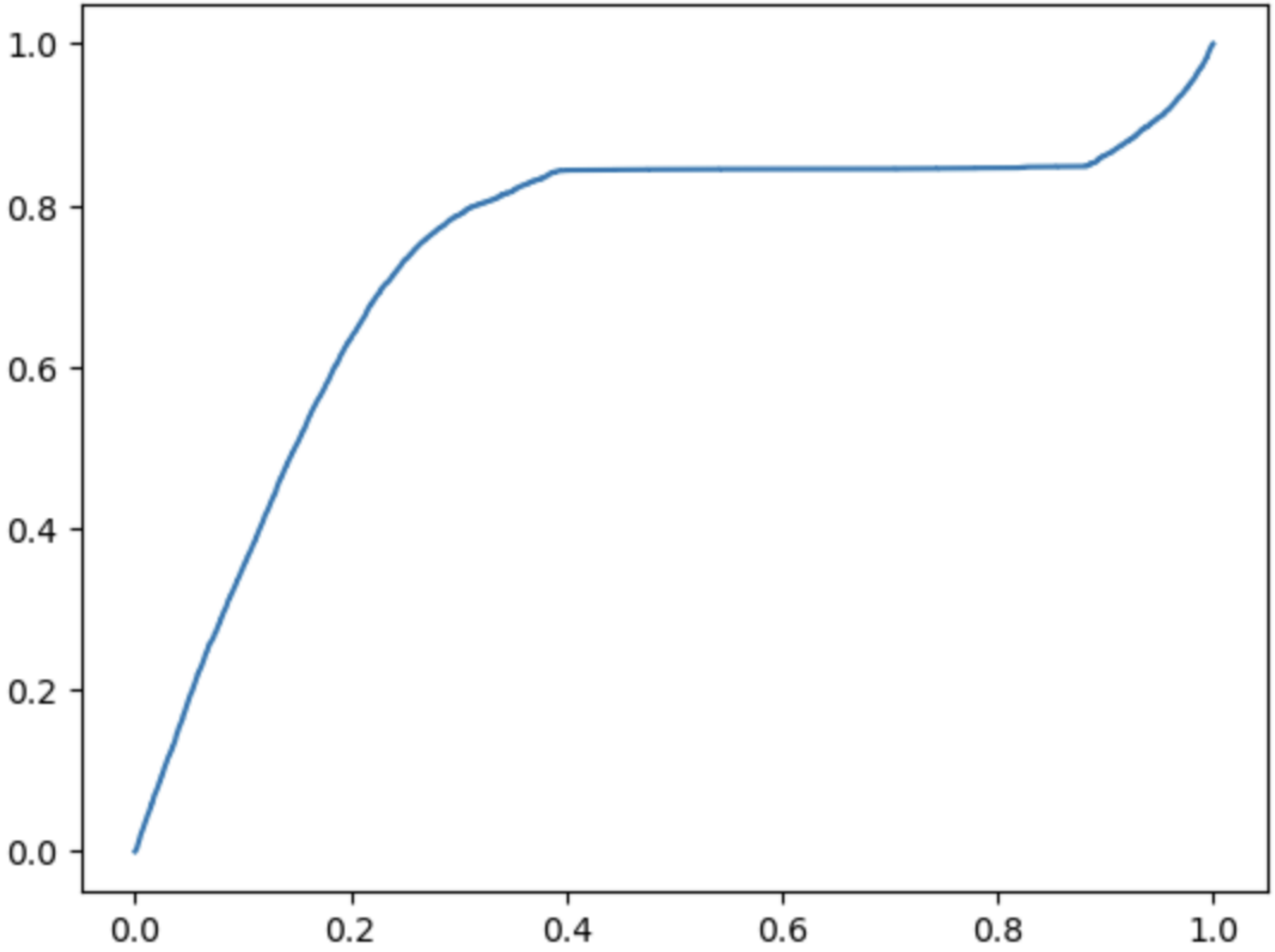


Hyperparameters Tuning for each model

Using StackingClassifier & GradientBoostingClassifier & XGBClassifier & AdaboostClassifier & LGBMClassifier:

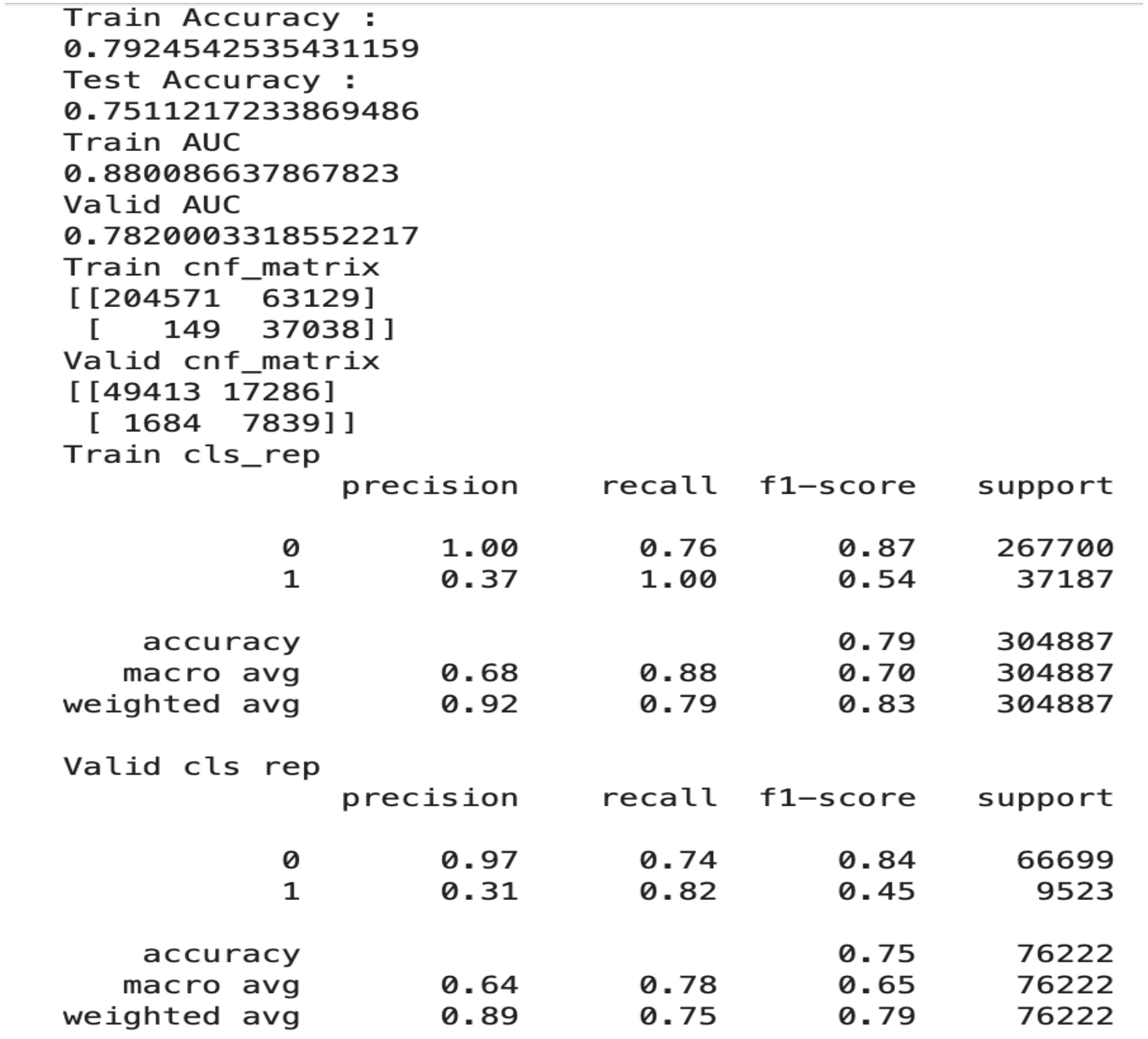


```
[LightGBM] [Warning] Unknown parameter: eval_metric
[LightGBM] [Warning] Unknown parameter: eval_metric
Train Accuracy :
0.9986421198673607
Test Accuracy :
0.859397549263992
Train AUC
0.9966101643463174
Valid AUC
0.5530254928638304
Train cnf_matrix
[[267512    188]
 [   226  36961]]
Valid cnf_matrix
[[64128  2571]
 [ 8146  1377]]
Train cls_rep
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    267700
           1       0.99      0.99      0.99     37187

    accuracy                           1.00    304887
   macro avg       1.00      1.00      1.00    304887
weighted avg       1.00      1.00      1.00    304887

Valid cls rep
              precision    recall  f1-score   support

           0       0.89      0.96      0.92     66699
           1       0.35      0.14      0.20      9523

    accuracy                           0.86     76222
   macro avg       0.62      0.55      0.56     76222
weighted avg       0.82      0.86      0.83     76222
```

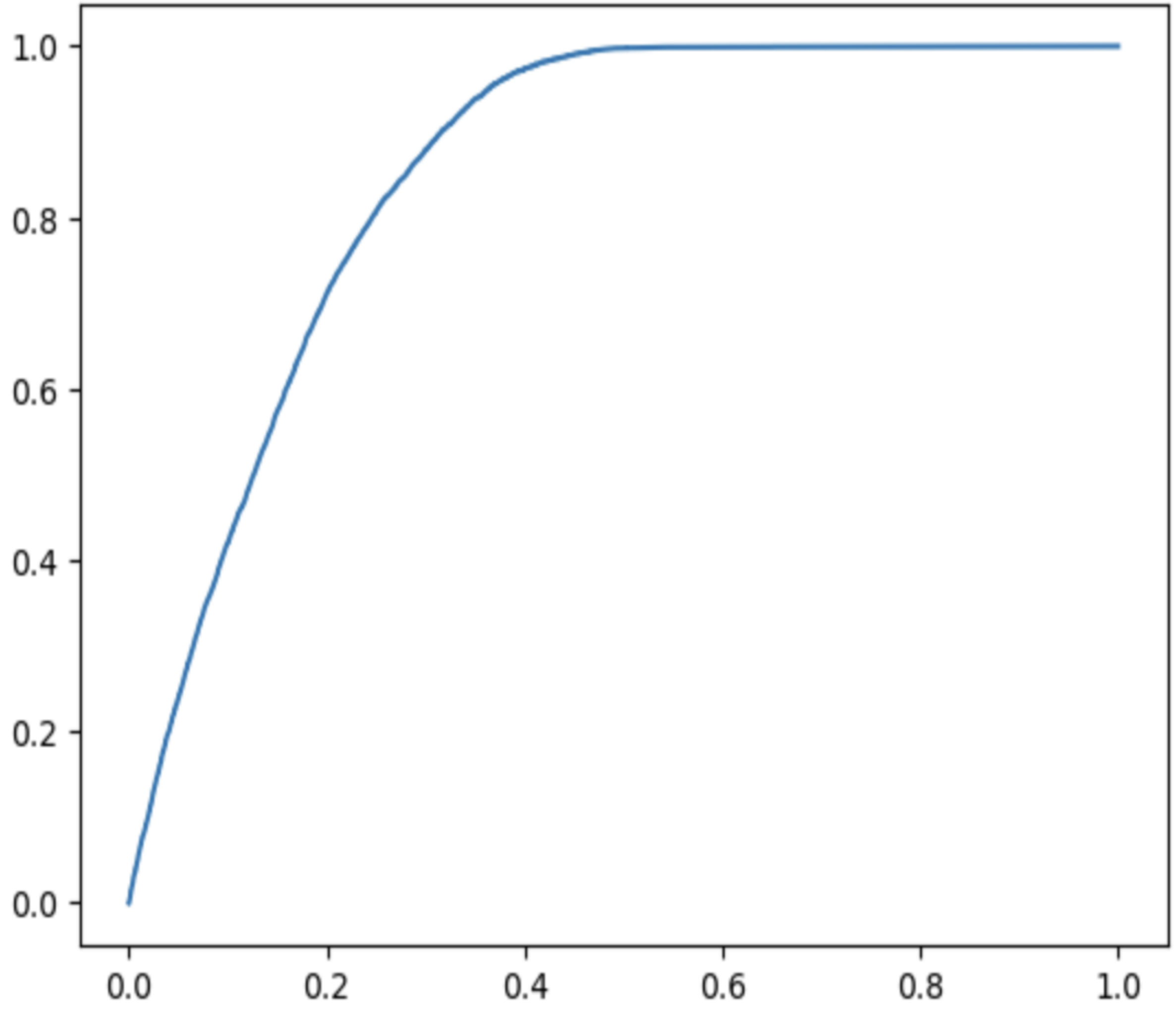


Hyperparameters Tuning for each model

Using StackingClassifier & XGBClassifier & CatBoostClassifier:

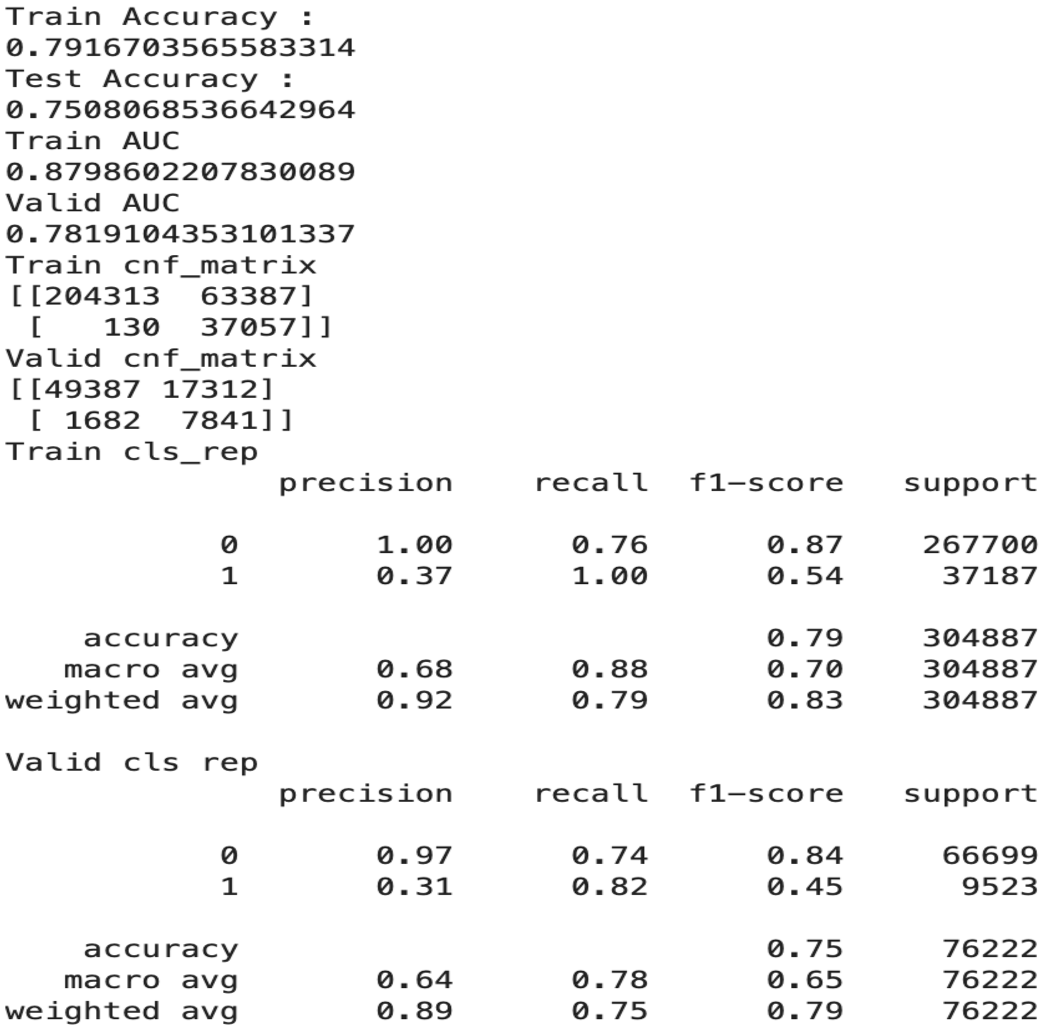


```
Train Accuracy :
0.7924542535431159
Test Accuracy :
0.7511217233869486
Train AUC
0.880086637867823
Valid AUC
0.7820003318552217
Train cnf_matrix
[[204571  63129]
 [   149  37038]]
Valid cnf_matrix
[[49413 17286]
 [ 1684  7839]]
Train cls_rep
              precision    recall  f1-score   support

           0       1.00      0.76      0.87    267700
           1       0.37      1.00      0.54     37187

    accuracy                           0.79    304887
   macro avg       0.68      0.88      0.70    304887
weighted avg       0.92      0.79      0.83    304887

Valid cls rep
              precision    recall  f1-score   support

           0       0.97      0.74      0.84     66699
           1       0.31      0.82      0.45      9523

    accuracy                           0.75     76222
   macro avg       0.64      0.78      0.65     76222
weighted avg       0.89      0.75      0.79     76222
```

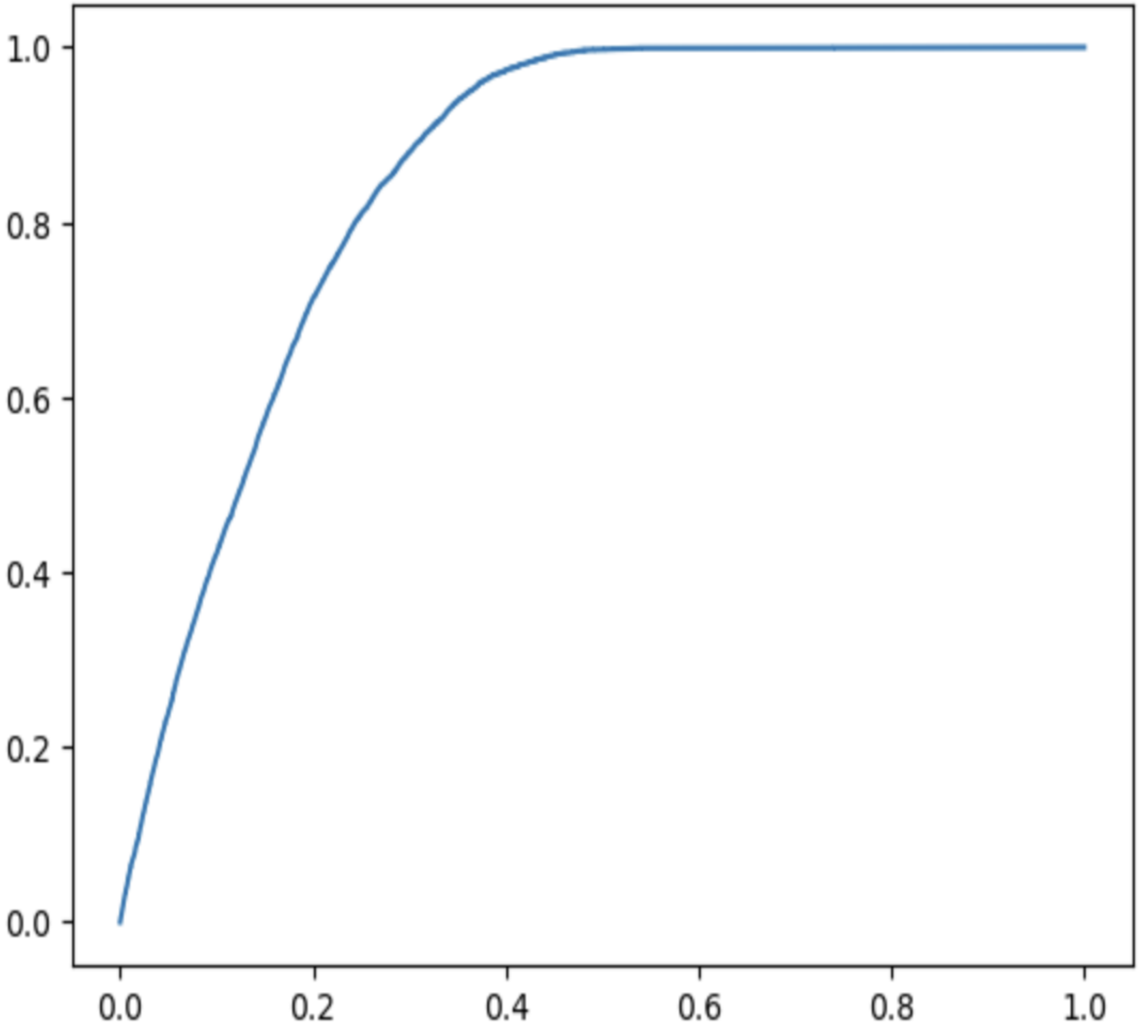


Hyperparameters Tuning for each model

Using StackingClassifier & XGBClassifier & RandomForestClassifier:

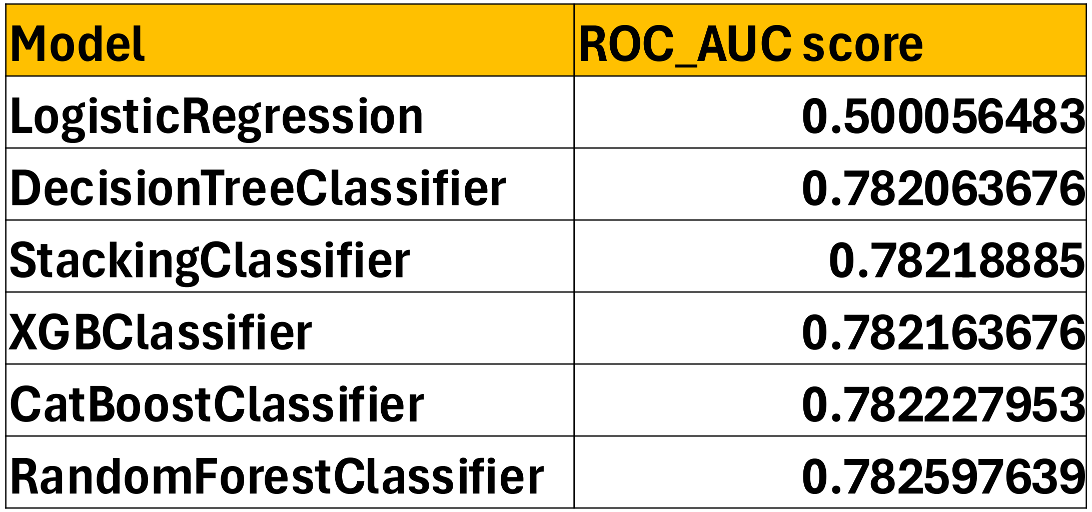


```
Train Accuracy :
0.7916703565583314
Test Accuracy :
0.7508068536642964
Train AUC
0.8798602207830089
Valid AUC
0.7819104353101337
Train cnf_matrix
[[204313  63387]
 [   130  37057]]
Valid cnf_matrix
[[49387 17312]
 [ 1682  7841]]
Train cls_rep
              precision    recall  f1-score   support

           0       1.00      0.76      0.87    267700
           1       0.37      1.00      0.54     37187

    accuracy                           0.79    304887
   macro avg       0.68      0.88      0.70    304887
weighted avg       0.92      0.79      0.83    304887

Valid cls rep
              precision    recall  f1-score   support

           0       0.97      0.74      0.84     66699
           1       0.31      0.82      0.45      9523

    accuracy                           0.75     76222
   macro avg       0.64      0.78      0.65     76222
weighted avg       0.89      0.75      0.79     76222
```



Classification Analysis:

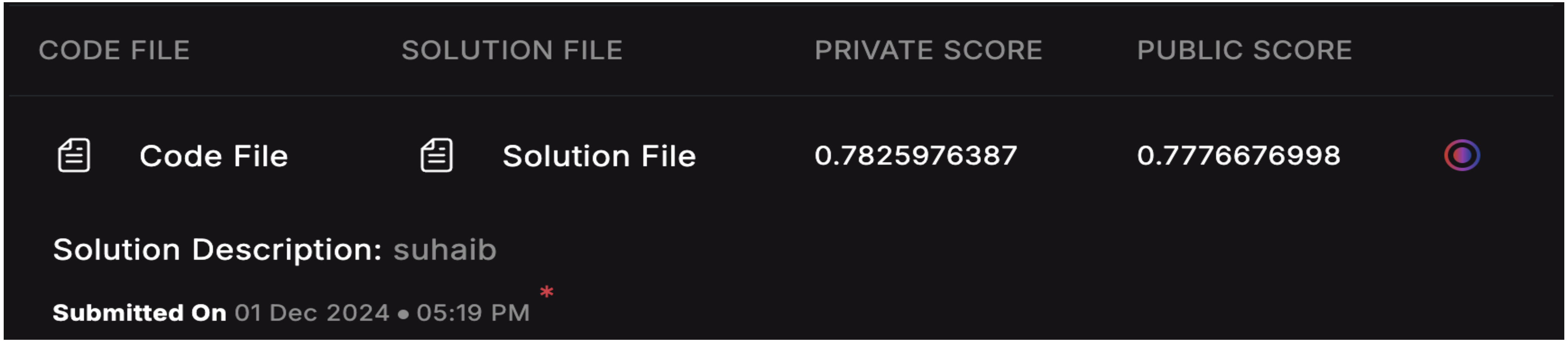
Applied the following models and scores obtained



Model	ROC_AUC score
LogisticRegression	0.500056483
DecisionTreeClassifier	0.782063676
StackingClassifier	0.78218885
XGBClassifier	0.782163676
CatBoostClassifier	0.782227953
RandomForestClassifier	0.782597639

Result uploaded in the analytics vidhya website and the recieved score

Final Evaluation Metric : ROC_AUC score

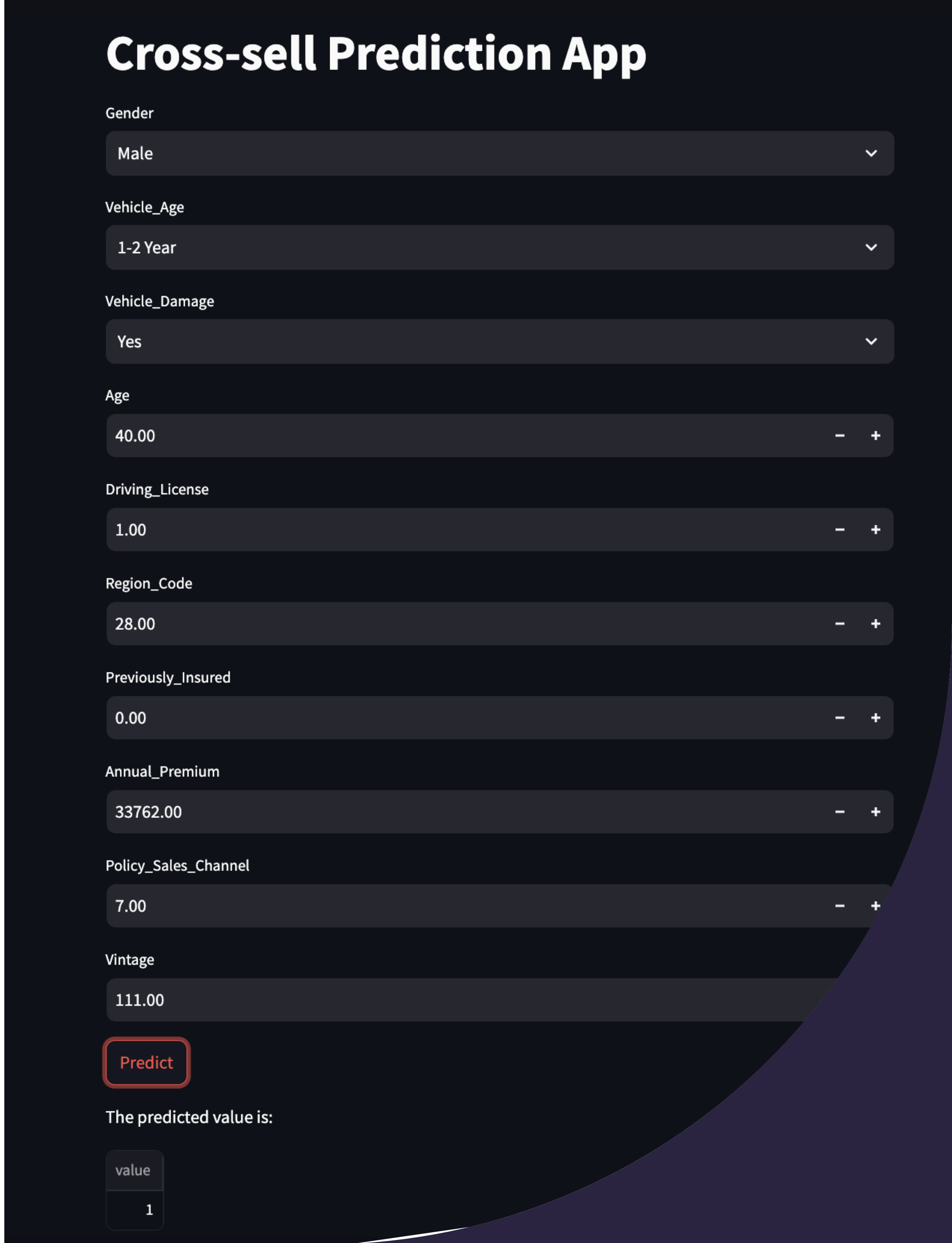


CODE FILE	SOLUTION FILE	PRIVATE SCORE	PUBLIC SCORE
Code File	Solution File	0.7825976387	0.7776676998

Solution Description: suhaib

Submitted On 01 Dec 2024 • 05:19 PM

Web API Creation End-Point



Cross-sell Prediction App

Gender: Male

Vehicle_Age: 1-2 Year

Vehicle_Damage: Yes

Age: 40.00

Driving_License: 1.00

Region_Code: 28.00

Previously_Insured: 0.00

Annual_Premium: 33762.00

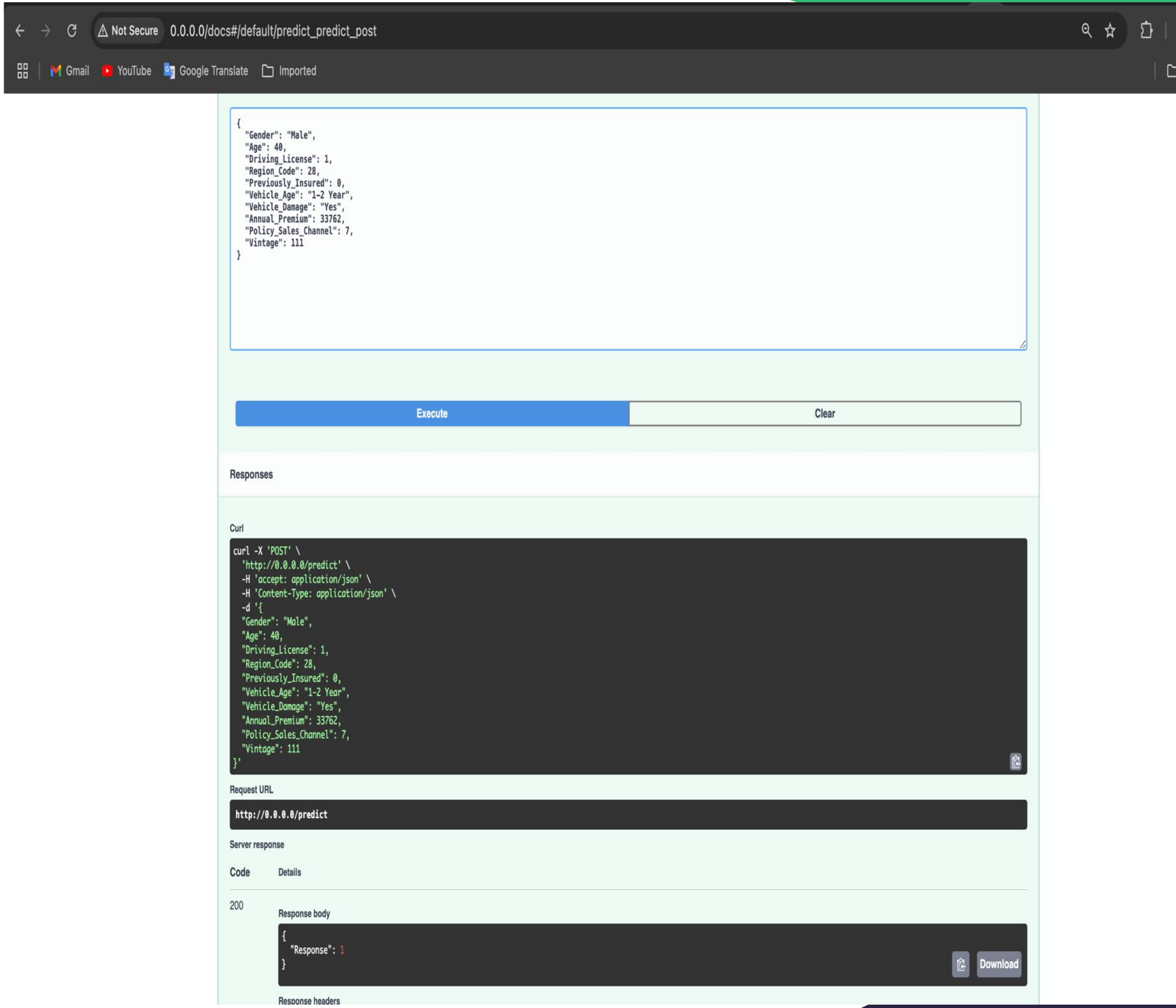
Policy_Sales_Channel: 7.00

Vintage: 111.00

The predicted value is:

value
1

FastAPI Creation End-Point



```
{
  "Gender": "Male",
  "Age": 40,
  "Driving_License": 1,
  "Region_Code": 28,
  "Previously_Insured": 0,
  "Vehicle_Age": "1-2 Year",
  "Vehicle_Damage": "Yes",
  "Annual_Premium": 33762,
  "Policy_Sales_Channel": 7,
  "Vintage": 111
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://0.0.0.0/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Gender": "Male",
  "Age": 40,
  "Driving_License": 1,
  "Region_Code": 28,
  "Previously_Insured": 0,
  "Vehicle_Age": "1-2 Year",
  "Vehicle_Damage": "Yes",
  "Annual_Premium": 33762,
  "Policy_Sales_Channel": 7,
  "Vintage": 111
}'
```

Request URL

```
http://0.0.0.0/predict
```

Server response

Code	Details
200	Response body `{ "Response": 1 }` Response headers

```
{
  "Response": 1
}
```

GCP CloudRun End-Points: FastApi and Streamlit

- https://crosssellfastapi-222441656201.us-central1.run.app

- https://crosssellstreamlit-222441656201.us-central1.run.app

Github Main URL: https://github.com/mohamedsuhaib88/MLHack2024/tree/main

Github URL for FastAPI CI/CD using GCP CloudRun:

- https://github.com/mohamedsuhaib88/crosssellstreamlit/tree/main

Github URL for Streamlit CI/CD using GCP CloudRun:

- https://github.com/mohamedsuhaib88/crosssellfastapi/tree/main

DockerHub Image:

msuhaiba/fastapi-crosssell:1.1

msuhaiba/fastapi-crosssellweb:1.2

DockerHub URL:

https://hub.docker.com/layers/msuhaiba/fastapi-crosssell/1.1/images/sha256-482dc6887664a9b2799d87cd34a20db28cd3a1c0c7d7feb8757e4cda2f2bd144?context=repo