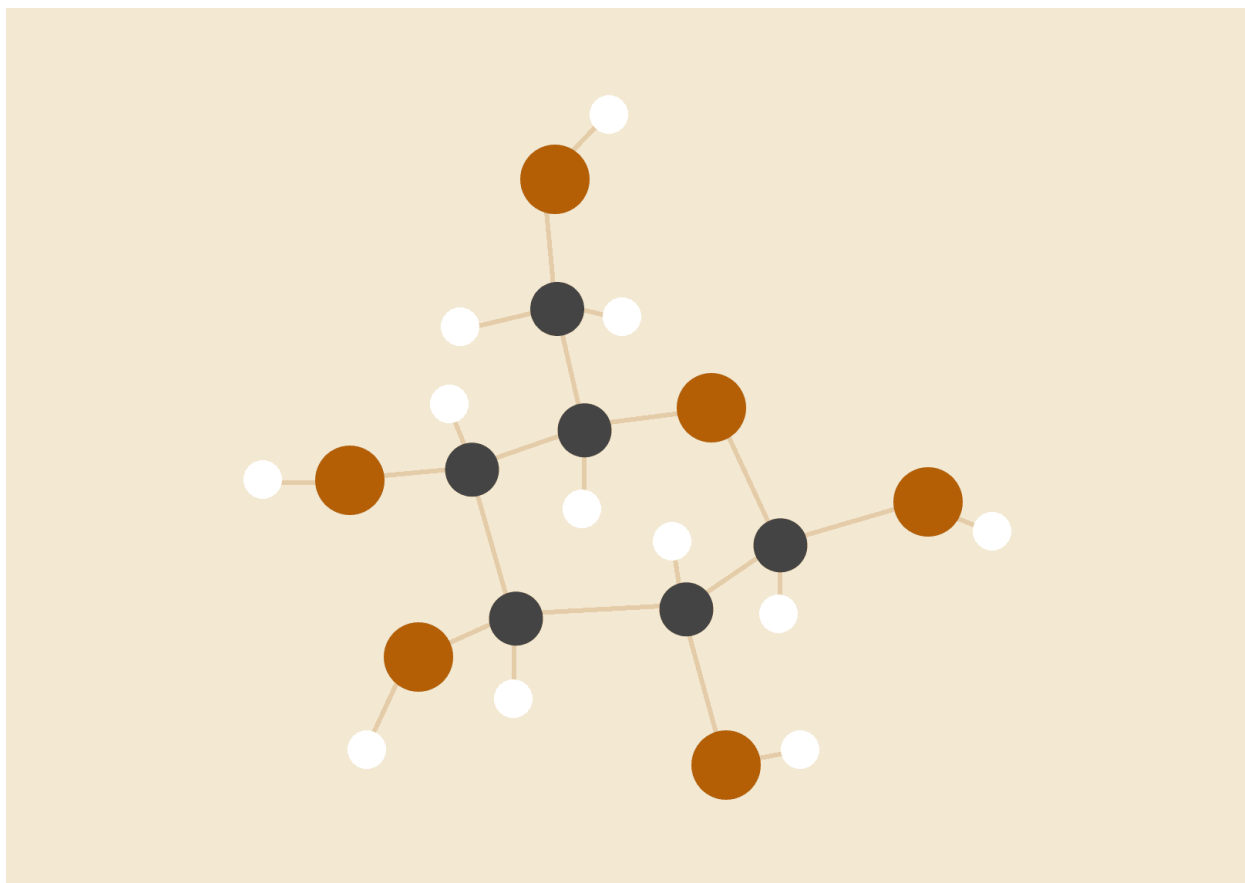


# Fraud Detection in Electricity and Gas Consumption in Tunisia

*[CIE 417] Machine Learning - Fall 2021 - Project 1*



## Team Members:

Ahmed Saad 201-800-251

Mazen Hassan 201-801-897

Yossef Eldesoky 201-800-504

## Problem Definition

The Tunisian Company of Electricity and Gas (STEG) is a public and non-administrative company, it is responsible for delivering electricity and gas across Tunisia. The company suffered tremendous losses in the order of 200 million Tunisian Dinars due to fraudulent manipulations of meters by consumers. Using the client's billing history, the aim of the challenge is to detect and recognize clients involved in fraudulent activities. The solution that we propose will enhance the company's revenues and reduce the losses caused by such fraudulent activities.

## About the Dataset

The [data](#) provided by STEG is composed of two files. The first one is comprised of client data and the second one contains billing history since 2005.

### Files:

There are 2 .zip files for download, train.zip, and test.zip, and SampleSubmission.csv.

- train.zip
  - Client\_train.csv - Client information in the train population
  - Invoice\_train.csv - Clients invoice in the train set
- test.zip
  - Client\_test.csv - Client information for the test population
  - Invoice\_test.csv - Clients invoice in the test set
  - SampleSubmission.csv - is an example of what your submission file should look like. The order of the rows does not matter, but the names of the IDs must be correct. The column "target" is your prediction. Measured by ROC AUC.

## Variable definitions

- **Client data:**
  - Client\_id: Unique id for client
  - District: District where the client is
  - Client\_catg: Category client belongs to

- Region: Area where the client is
- Creation\_date: Date client joined
- Target: fraud:1 , not fraud: 0
- **Invoice data:**
  - Client\_id: Unique id for the client
  - Invoice\_date: Date of the invoice
  - Tarif\_type: Type of tax [more info](#)
  - Counter\_number: counter number
  - Counter\_statue: takes up to 5 values such as working fine, not working, on hold statue, etc.
  - Counter\_code: counter code
  - Reading\_remarque: notes that the STEG agent takes during his visit to the client (e.g: If the counter shows something wrong, the agent gives a bad score)
  - Counter\_coefficient: An additional coefficient to be added when standard consumption is exceeded
  - Consommation\_level\_1: Consumption\_level\_1
  - Consommation\_level\_2: Consumption\_level\_2
  - Consommation\_level\_3: Consumption\_level\_3
  - Consommation\_level\_4: Consumption\_level\_4
  - Old\_index: Old index
  - New\_index: New index
  - Months\_number: Month number
  - Counter\_type: Type of counter

## Approach and Methodology

In this part, we will describe the approach and steps we've taken through our analysis and training.

## A- Exploratory Data Analysis:

### 1) Importing Dataset and libraries:

We've imported the datasets and the needed libraries are:

- Sklearn
- NumPy
- Pandas
- Classifiers: XGBoost, CatBoost, LGBM, KNN, AdaBoost, SVC .. etc

### 2) Distinguish Attributes:

a) Train Client: about 135,000 instances that each describe a unique client with information about its region, account creation date, district, and target (1==fraud).

	disrict	client_id	client_catg	region	creation_date	target
0	60	train_Client_0	11	101	31/12/1994	0.0
1	69	train_Client_1	11	107	29/05/2002	0.0
2	62	train_Client_10	11	301	13/03/1986	0.0
3	69	train_Client_100	11	105	11/07/1996	0.0
4	62	train_Client_1000	11	303	14/10/2014	0.0

b) Train Invoice: about 4,500,000 instances, in which a specific client can have different invoices at different times, each with billing details like the consumption level, invoice date, tariff type, reading remark of STEG agent, etc.

	client_id	invoice_date	tarif_type	counter_number	counter_statue	counter_code	reading_remarque	counter_coefficient	consommation_level_1
0	train_Client_0	2014-03-24	11	1335667	0	203	8	1	82
1	train_Client_0	2013-03-29	11	1335667	0	203	6	1	1200
2	train_Client_0	2015-03-23	11	1335667	0	203	8	1	123
3	train_Client_0	2015-07-13	11	1335667	0	207	8	1	102
4	train_Client_0	2016-11-17	11	1335667	0	207	9	1	572

	consommation_level_2	consommation_level_3	consommation_level_4	old_index	new_index	months_number	counter_type
	0	0	0	14302	14384	4	ELEC
	184	0	0	12294	13678	4	ELEC
	0	0	0	14624	14747	4	ELEC
	0	0	0	14747	14849	4	ELEC
	0	0	0	15066	15638	12	ELEC

### 3) Missing data and Duplicates:

We've checked the missing and duplicated data of two datasets, and found duplicates in the "Train invoice" Dataset and we've successfully handled that by removing them.

## B- Feature Engineering:

### 1) Feature Creation and Transformation:

We tried to merge the datasets in one dataset, so it can be easier to apply univariate analysis and train/test models. We've aggregated the large dataset "Train Invoice", for each client we got the different aggregation functions of (minimum-maximum-mean-standard deviation). Now We're having for every client about 50:60 features instead of 15 features which can give our model more data to learn through.

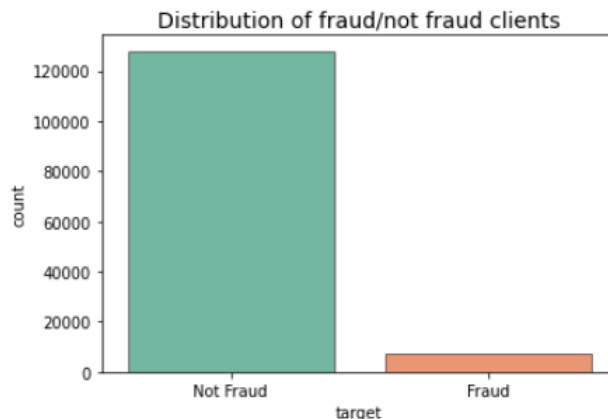
Eventually, the second dataset reached the same shape as the "Train Client" dataset. Then the datasets are merged together to X\_train, and we did the same to "Test Invoice".

Now, We're having **X\_train with shape (135493, 56)** and **X\_test with the shape (58069, 55)**.

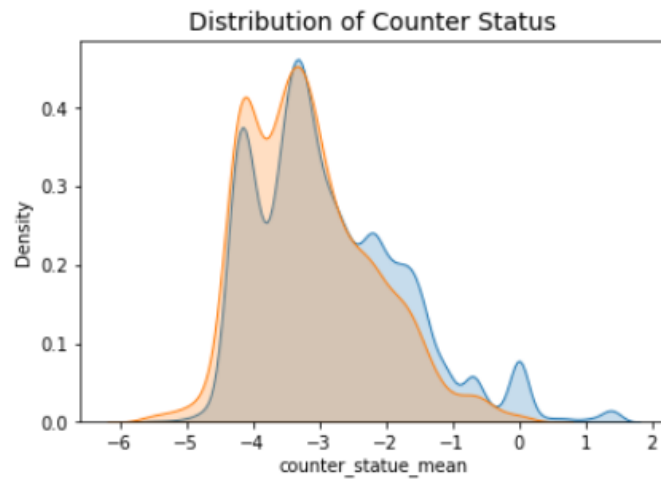
After Aggregating Dataset and merging them in one informative dataset, We'll go through Univariate Analysis to gain insights about our data.

### 2) Univariate Analysis:

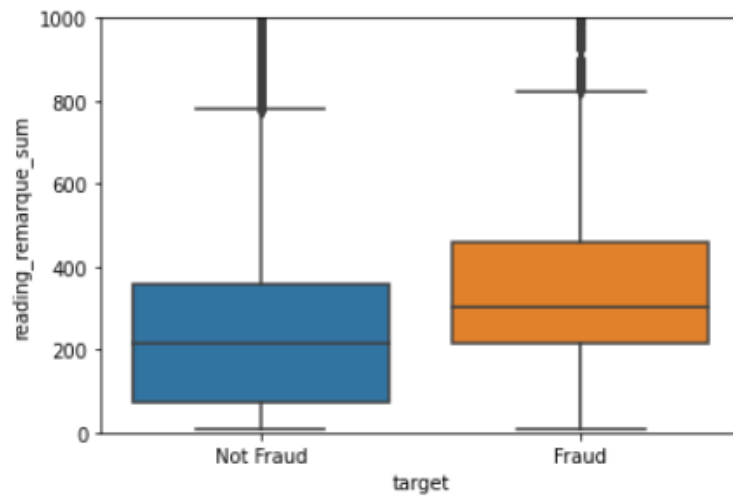
Q1- Is the data balanced?



According to the figure shown above, it's clear that our data is extremely imbalanced.



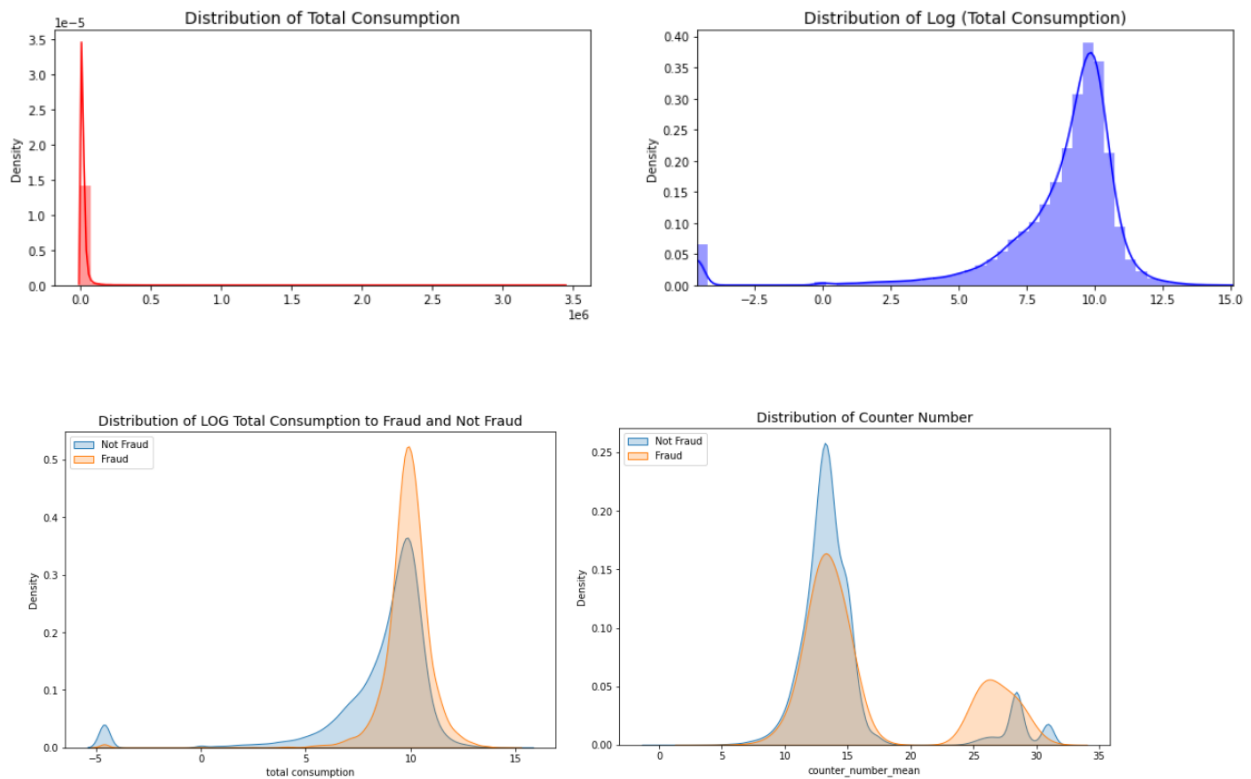
Q2- Are the STEG agent able to recognize the fraud case by its remark?



STEG Agent Readings boxplot

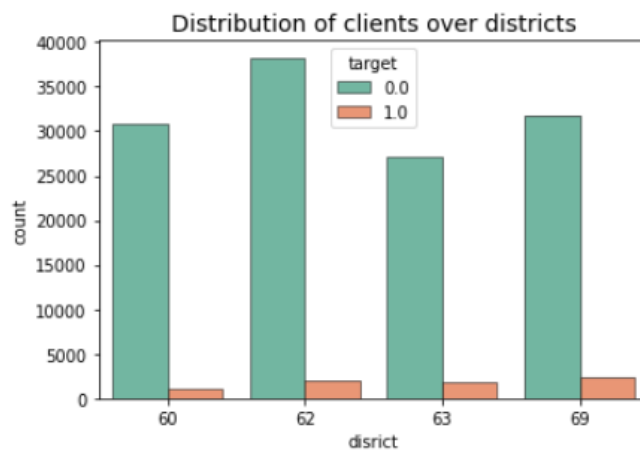
According to the above figure, the STEG agent will be almost able to give a good reading that differentiates between fraud and not a fraud.

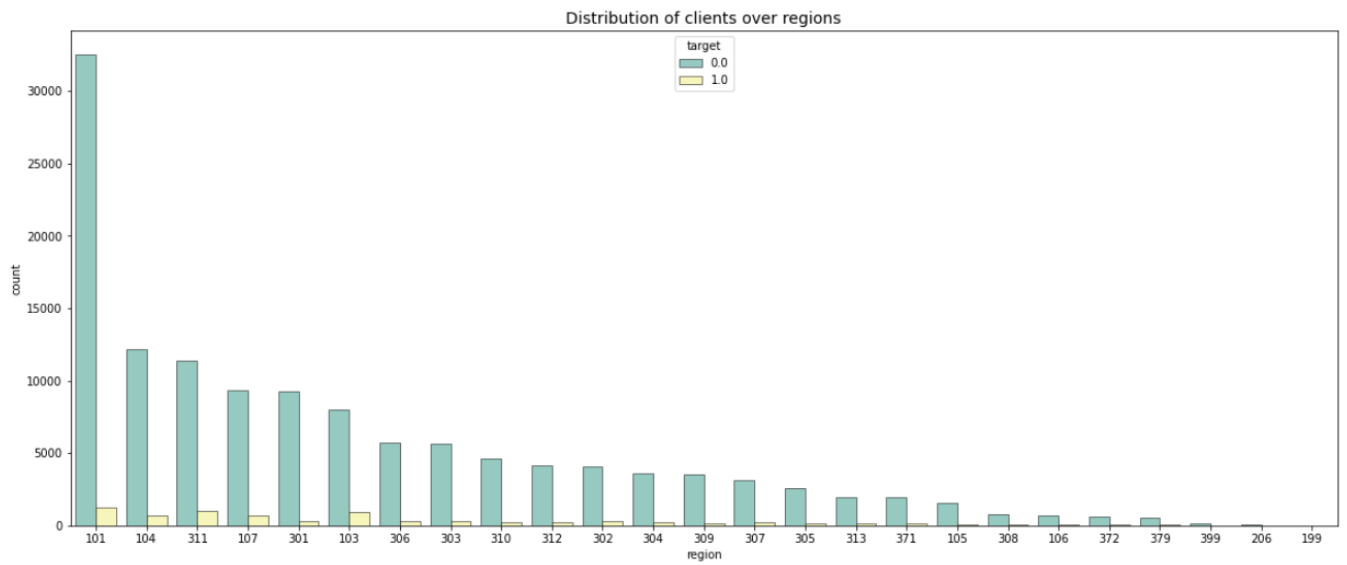
Q3- Does the ratio of fraud/not fraud increase with time?



According to the above figures, we notice that total consumption and counter number of fraud clients are larger than usual.

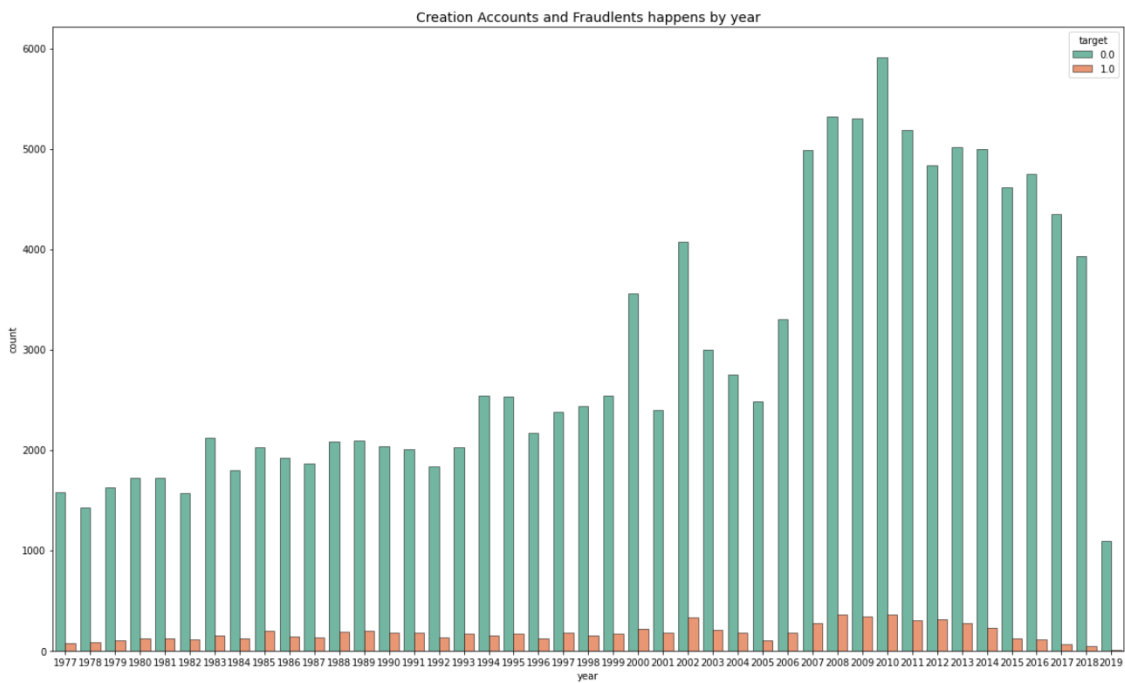
Q4- Is there a specific district that has highly fraudulent processes?





All districts/regions seem to have the same ratio of fraud/not fraud clients.

**Q5- Does the ratio of fraud/not fraud increase with time?**

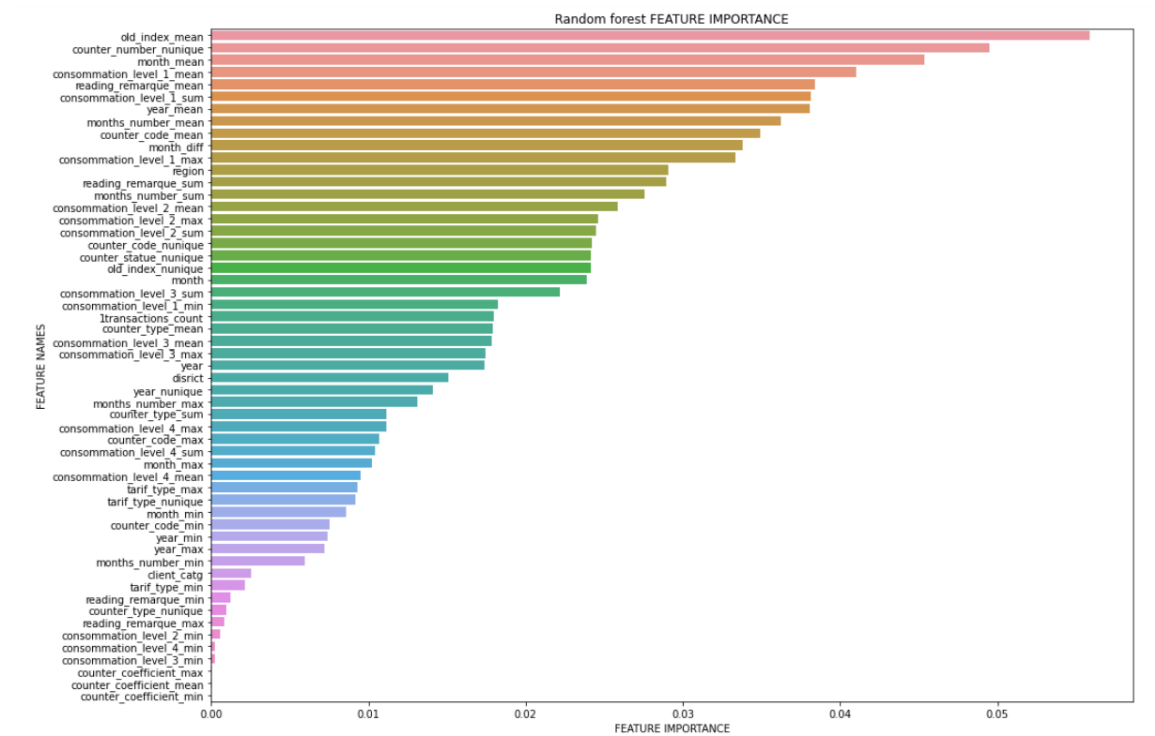


The ratio is almost the same, Although in the last 5 years it has been decreased not so much.



## 6) Feature Selection and Importance:

Using the Random Forest Classifier:



## C- Training and Testing Models:

We've tried many classifiers models and here's the list of them (Only XGBoost and LGBM are hypertuned):

Note: I've used SMOTE for balance data, but the score of hyper tuned balanced LGBM and hyper tuned imbalanced LGBM.

CatBoostClassifier.csv	<a href="#">↓</a>	<b>0.8787466116087976</b>
AdaBoostClassifier.csv	<a href="#">↓</a>	<b>0.8352588576680033</b>
XGBoost.csv	<a href="#">↓</a>	<b>0.8797492303629078</b>
RandomForest.csv	<a href="#">↓</a>	<b>0.8600924234647074</b>
LogisticRegression.csv	<a href="#">↓</a>	<b>0.7025783016287094</b>
LGBM.csv	<a href="#">↓</a>	<b>0.880139380303575</b>
ExtraTree.csv	<a href="#">↓</a>	<b>0.6060952138992476</b>

And the Stacking Model with LGBM as Meta: 0.8792

We've tried some of them gathered in the Ensemble Learning Model and Stacking Model, The two giant moves we took is through those two models:

- $0.35 \times \text{XGB} + 0.35 \times \text{LGBM} + 0.1 \times \text{RF} + 0.2 \times \text{AdaBoost}$
- $0.3 \times \text{XGB} + 0.4 \times \text{LGBM} + 0.3 \times \text{CatBoost}$

Those two ensemble learning models got the fourth rank with the hypertuning of the following:






- XGBoost
- LGBM: Meta Model

Notice that the Random Forest Benchmark is at 35th rank

We were able to achieve the 4th rank on the leaderboard between 900 competitors

## Competition Leaderboard

Unless stated otherwise in the Info Page, this leaderboard reflects scores based on only a portion of the total test set until the competition closes. See competition Info for more information.

RANK	USER	SCORE	LAST SUBM	# SUBM
1	 <b>Aziz_Belkhir</b> Expensya	0.891845228882063	over 1 year ago	51
2	 <b>Centre universitaire Ain temouchent</b> Team	0.889050849726237	over 1 year ago	60
3	 <b>ragnarok</b>	0.885872755606479	over 2 years ago	25
4	 <b>ahmedlila</b> University of science and technology at zewail city	0.885205561092212	4 days ago	68
5	 <b>GORNYAKI</b> Team	0.883030480986258	over 1 year ago	11

## C- Data Findings and Limitations

### Findings:

- SVM took much time, although we used scaling and gamma to auto
- Fraud clients consume more than normal people
- Fraud clients can be noticed by the STEG client
- The ratio of Fraud/Not Fraud is almost the same through all years.
- There is no specific district/region where exists a large number of fraudulent than others.

### Limitations:

- Can't detect the hour where the fraud occurs.
- Data is imbalanced
- The number of frauds might not represent the all frauds happened
- Lack of correlations between variables and target.

### **Improvements to get a high score:**

- Needs hyper tuning with Kfolds for the XGBoost and AdaBoost Models
- Need to remove useless outliers

## **CONCLUSION**

We've made about 70 submissions through 5 days, and got the 4th rank on the second day on [the leaderboard](#). And Wow, That was a Marvelous Journey We really loved :).