Capstone Project
Fraud Detection in Electricity and Gas Consumption

Course Instructor: Dr. Suman Saha

Collaborators
Zakria Saad
Tinashe Hafe
Trymore Ncube

The team

“We are the ones to solve the problem we identified”

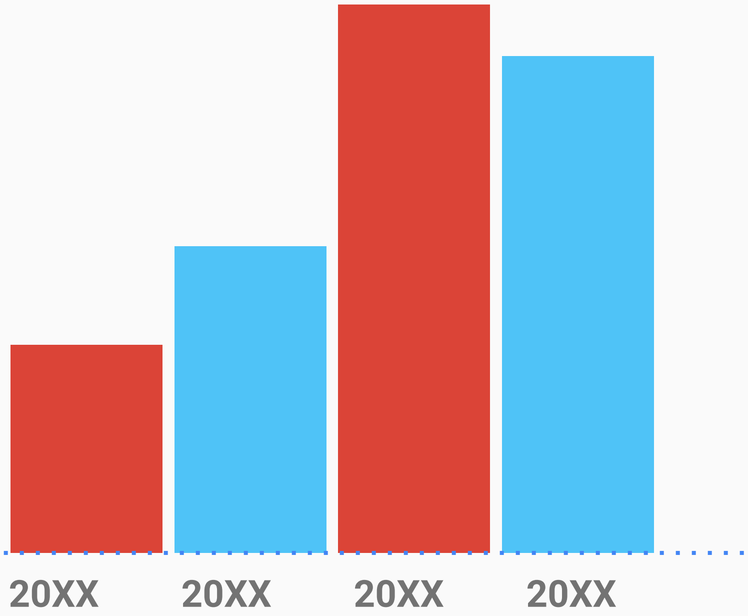
TINASHE HAFE

ZAKRIA SAAD

TRYMORE NCUBE

AGENDA

- Introduction
- Data Description
- Data Preprocessing
- EDA
- Feature Engineering
- Building Machine Learning Models
- Data Balancing Techniques
- Hyper Parameter Tuning
- Conclusion



INTRODUCTION

The Tunisian Company of Electricity and Gas (STEG) is a public and a 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. Our target is to identify the fraudulent customers with the help of historical data of consumers

Project Timeline

1st Week

JAN 24-29th

Data Preprocessing,
EDA
Report writing

2nd Week

JAN 30- 6 FEB

Building ML Models
Selecting Best ML models

3rd Week

FEB 7-13th

Feature Engineering
Data Balancing
Report Writing

4th Week

FEB 14-20th

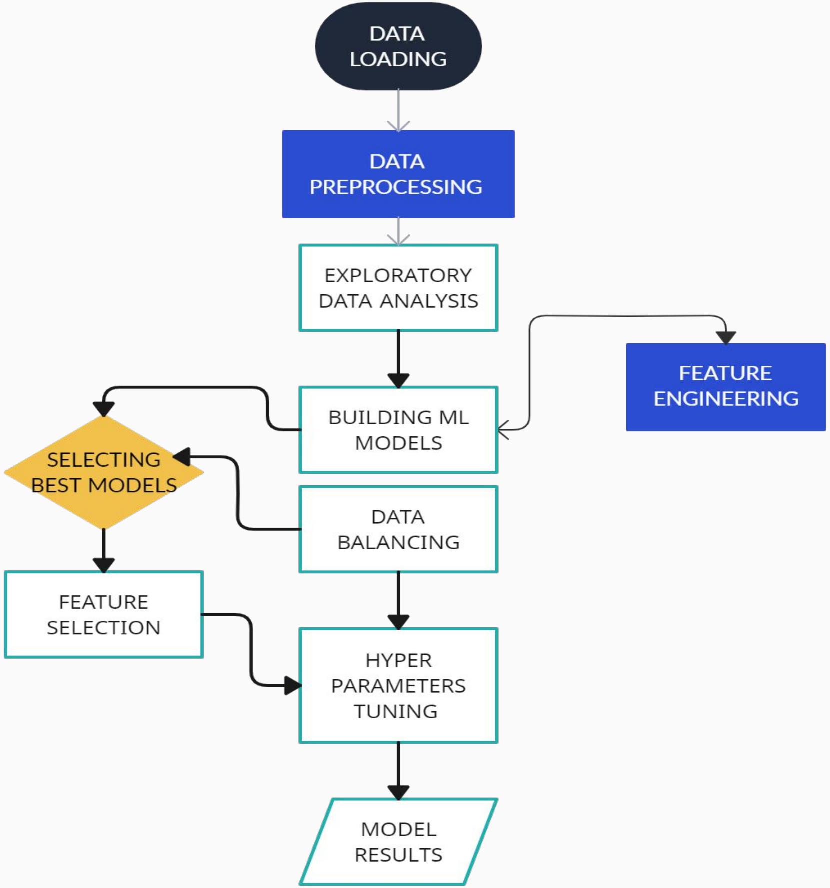
Training ML Models
Hyper Parameter Tuning

5th Week

FEB 21-25th

Feature Selection
Tuning Best ML Models
Presentatiom & Report Writing

PROJECT WORKFLOW



DATA DESCRIPTION

Client Dataset (135k,5)
Invoice Dataset(4.4M,16)

Dataset statistics

Number of variables	6
Number of observations	135493
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	6.2 MiB
Average record size in memory	48.0 B

```
invoice.shape

(4476749, 16)

 #   Column                 Dtype
---  ------                 -----
 0   client_id              object
 1   invoice_date           object
 2   tarif_type             int64
 3   counter_number         int64
 4   counter_statue         object
 5   counter_code           int64
 6   reading_remarque       int64
 7   counter_coefficient    int64
 8   consommation_level_1   int64
 9   consommation_level_2   int64
 10  consommation_level_3   int64
 11  consommation_level_4   int64
 12  old_index              int64
 13  new_index              int64
 14  months_number          int64
 15  counter_type           object
```

DATA PREPROCESSING AND EDA

Understanding the Data

Formatting data types

Working with date feature

Data Visualization

INITIAL PREPROCESSING

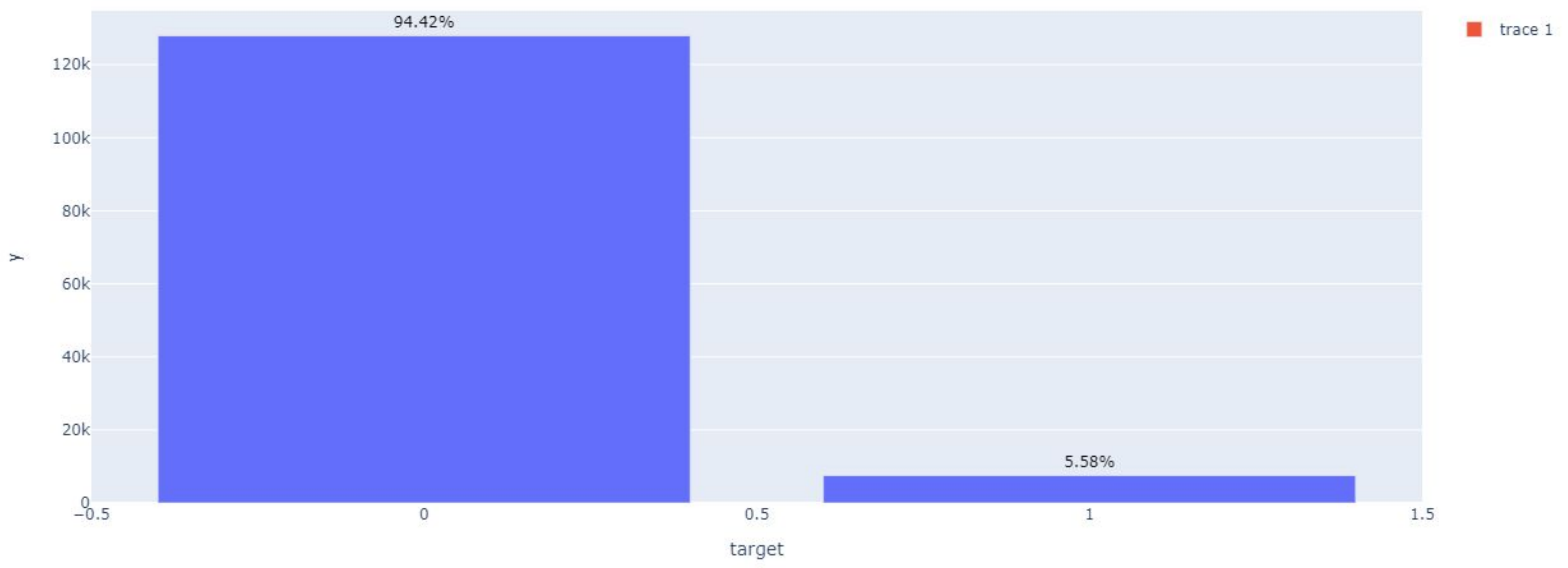
```
Number of missing rows in invoice_train: 0
Number of missing rows in invoice_test: 0

Number of missing rows in client_train: 0
Number of missing rows in client_test: 0
```

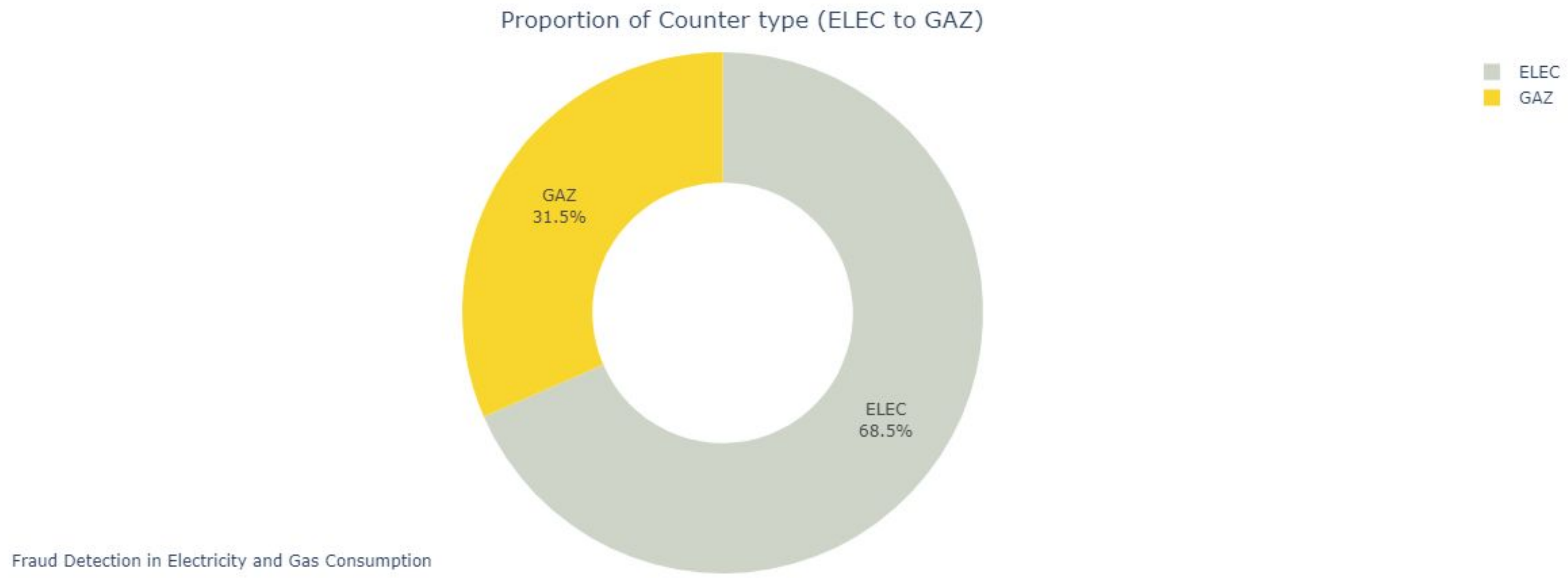
- extracted month, year from 'invoice_date', also added binary feature - 'is_weekday'
- client_catg', 'district' and 'region' were assigned as categories to use them as categorical features in modelling

```
invoice_train['invoice_date']=pd.to_datetime(invoice_train['invoice_date'])
invoice_train["year"] = invoice_train["invoice_date"].dt.year
invoice_train["month"] = invoice_train["invoice_date"].dt.month
```

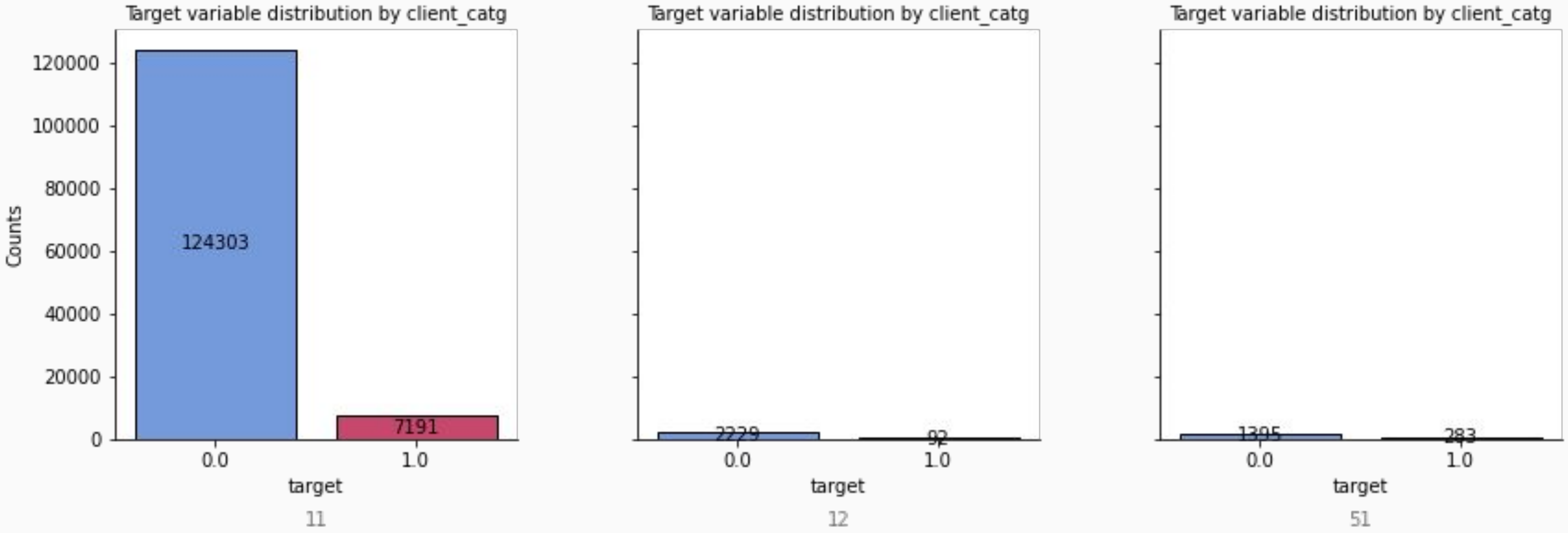
TARGET FEATURE DISTRIBUTION



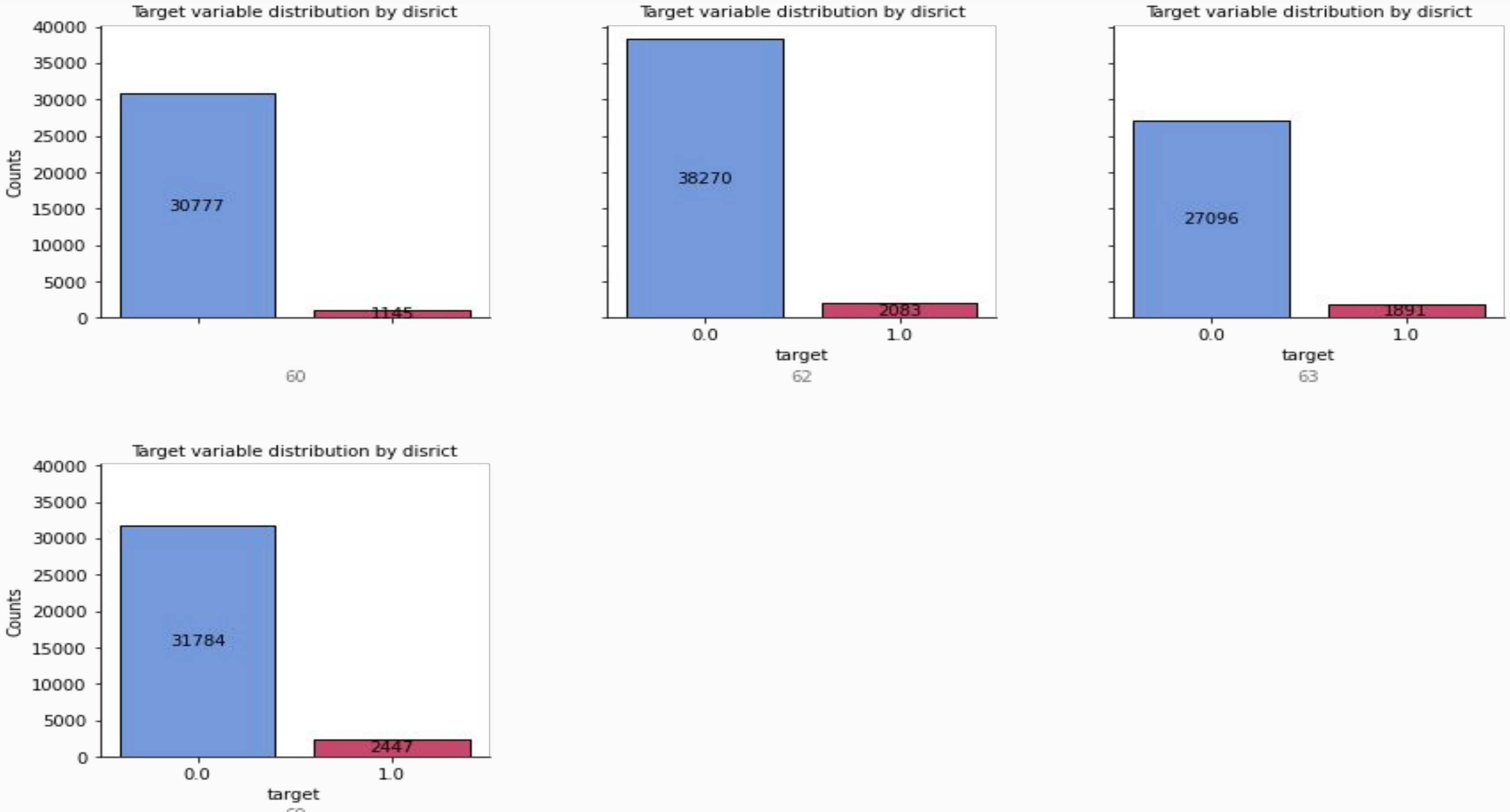
DISTRIBUTION OF ELECTRICITY AND GAS CONSUMERS



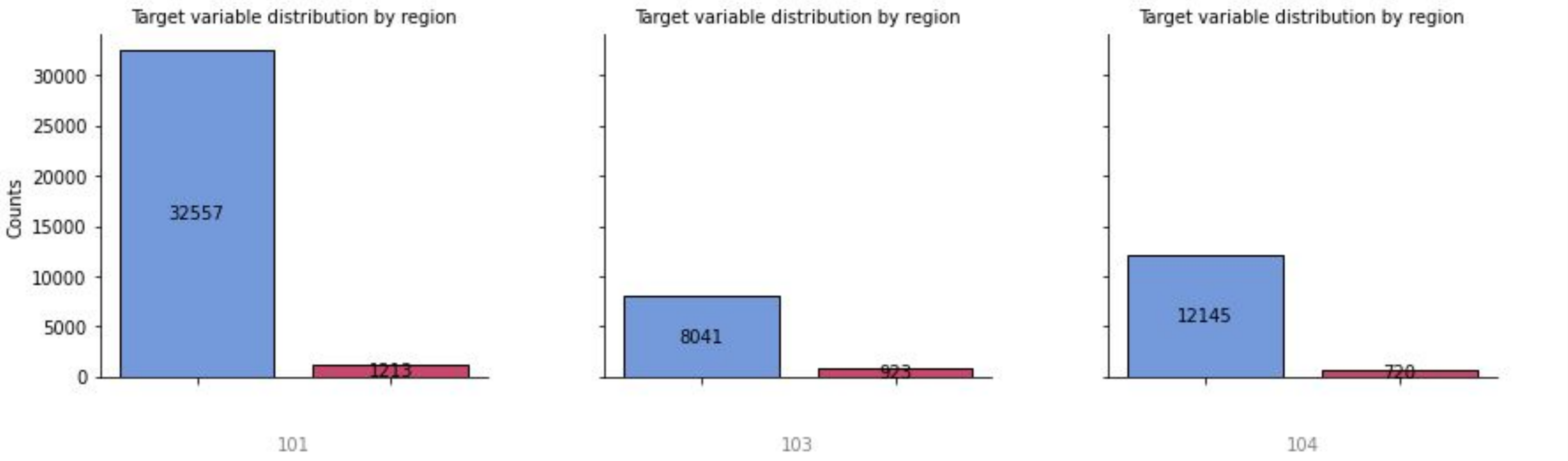
CONSUMER CATEGORIES

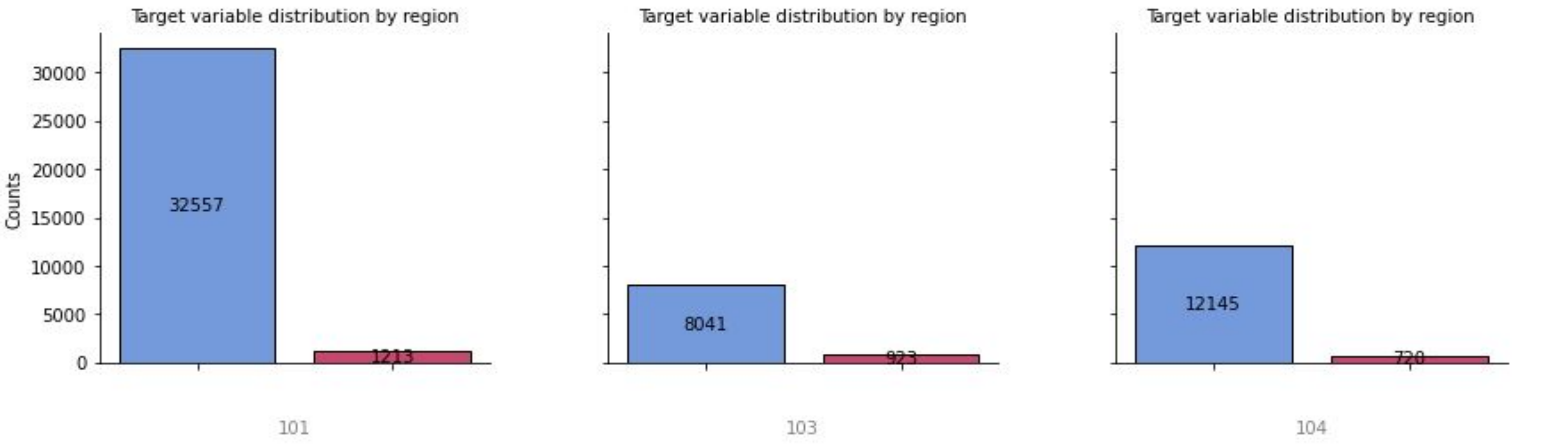


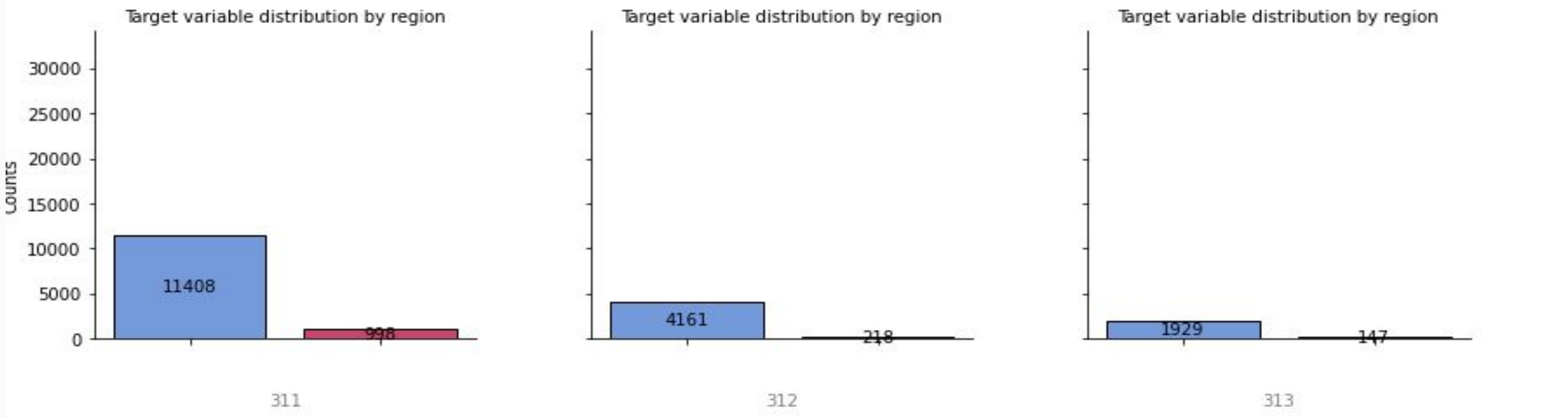
DISTRIBUTION OF CONSUMERS W.R.T DISTRICTS



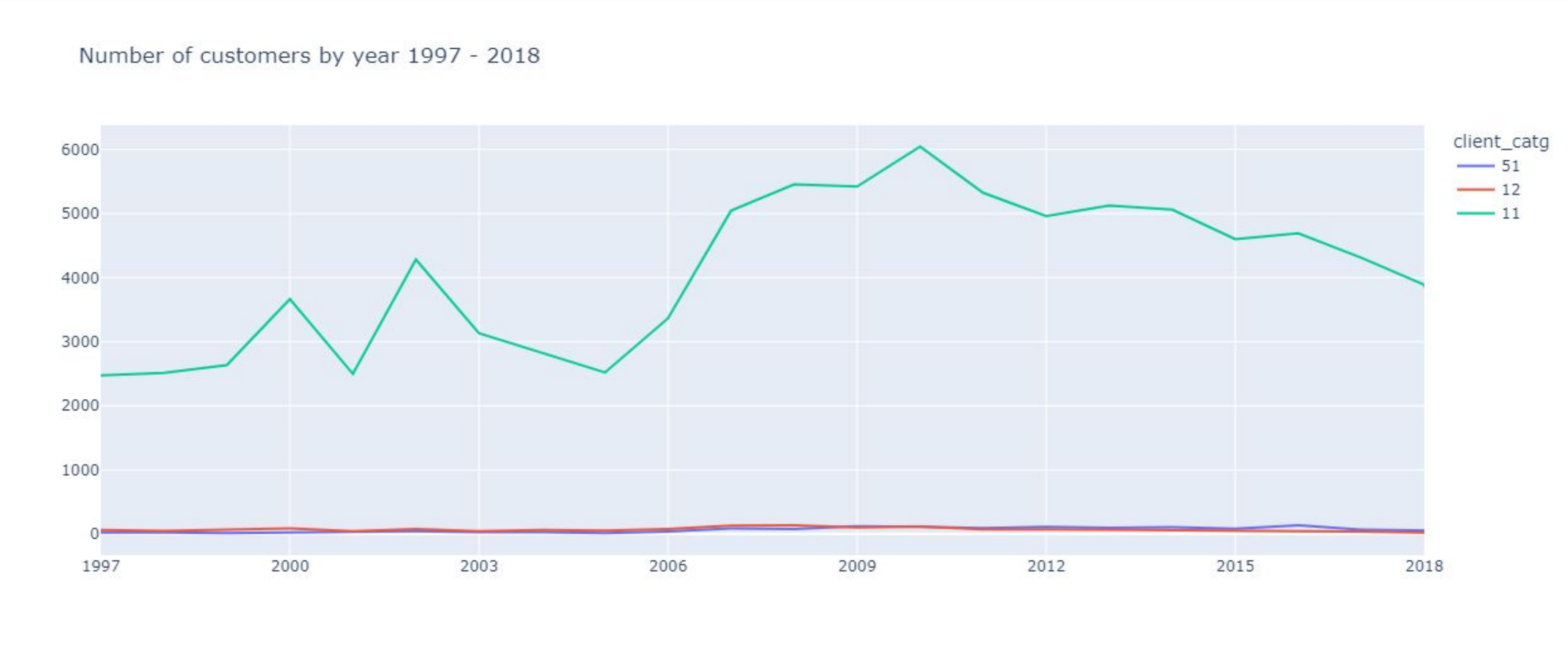
DISTRIBUTION OF CONSUMERS BY REGION



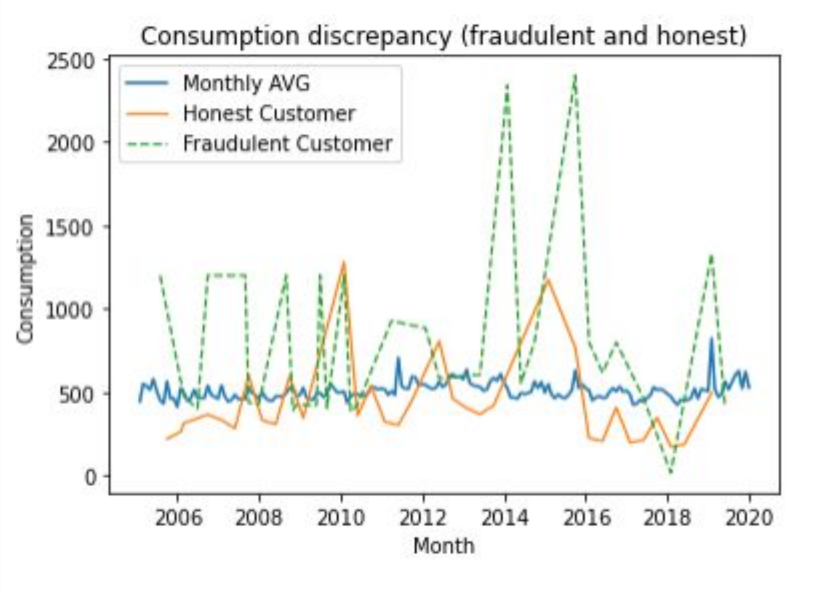

Target variable distribution by region
Counts
30000
25000
20000
15000
10000
5000
0
32557
1213
101
Target variable distribution by region
8041
923
103
Target variable distribution by region
12145
720
104

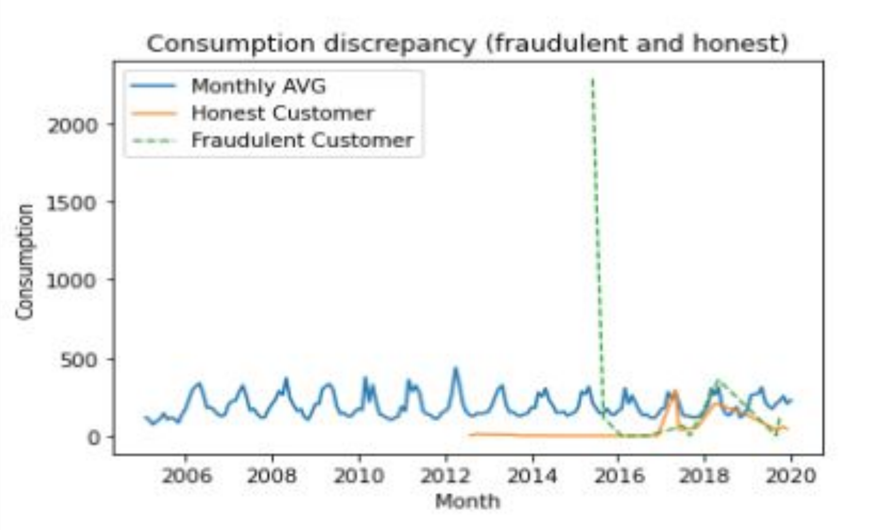

Target variable distribution by region
Counts
30000
25000
20000
15000
10000
5000
0
11408
998
311
Target variable distribution by region
4161
218
312
Target variable distribution by region
1929
147
313

CUSTOMERS FROM 1997 - 2018

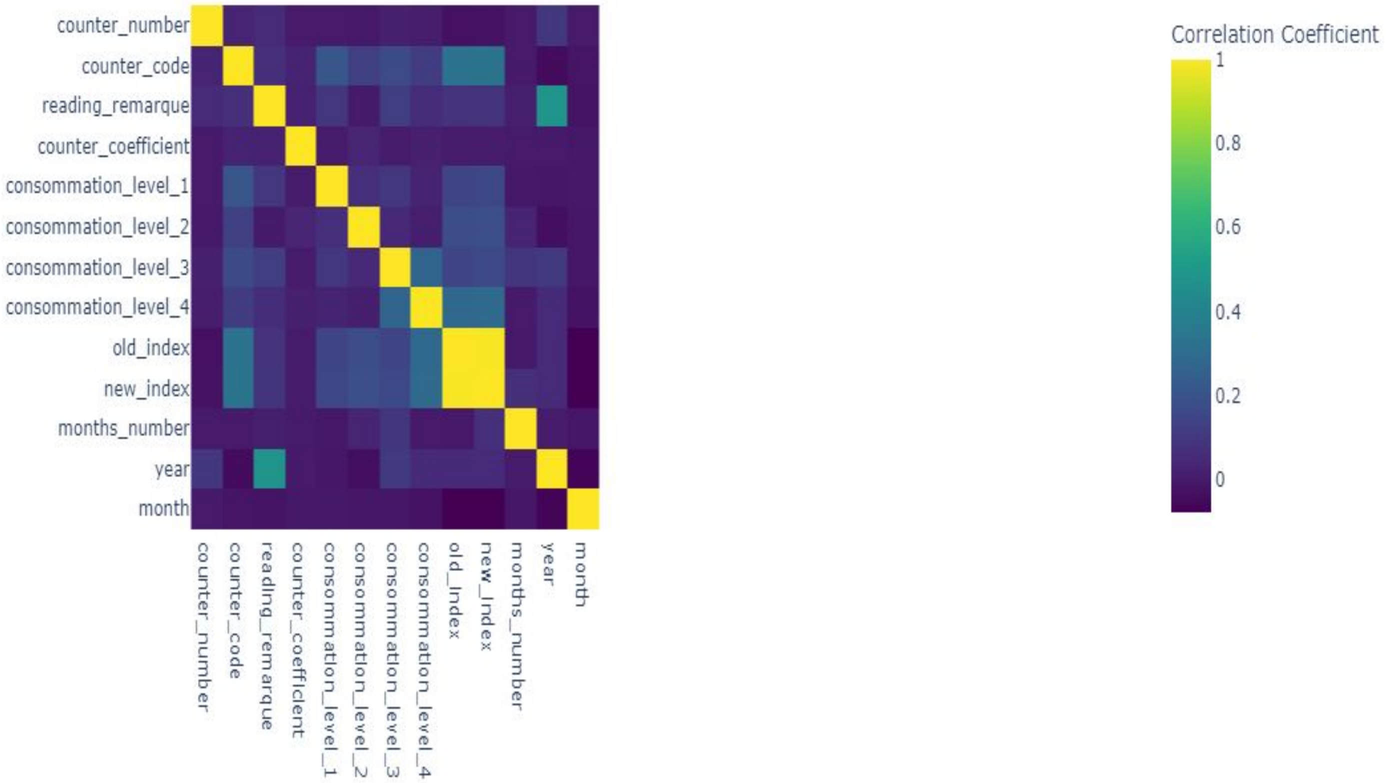


DISCREPANCY BETWEEN A FRAUDULENT AND AN HONEST CONSUMER





CORRELATION BETWEEN FEATURES



Feature Engineering

- Feature Transformation
- Methods and techniques

FEATURE TRANSFORMATION

Features that were used to generate new features

- Electric and Gas Consommation levels (1-4)
- Old Index
- New Index
- Tariff Type
- Counter Status

FEATURE TRANSFORMATION METHODS

- Cumulative Sum
- Measures of Central Tendencies (Mean, mode, median)
- Measures of Spread (range, standard deviation, variance)
- Group By methods in Pandas

```
summary_invoice_train = (
    invoice_train_cumsum.loc[:, ~invoice_train_cumsum.columns.isin(["counter_code", "counter_number"])].groupby(["client_id", "counter_type"]).agg(avg_conson_1_1=("consommation_level_1", "mean"),
        var_conson_1_1=("consommation_level_1", "var"),
        sd_conson_1_1=("consommation_level_1", "std"),
        median_conson_1_1=("consommation_level_1", "median"),
        mode_conson_1_1=("consommation_level_1", find_mode),
        avg_diff_conson_1_1=("consommation_level_1", lambda x: np.mean(np.diff(x))),
        range_conson_1_1=("consommation_level_1", lambda x: np.max(x) - np.min(x)),
        sd_cumsum_consommation_level_1=("cumsum_consommation_level_1", "std"),
        avg_cumsum_consommation_level_1=("cumsum_consommation_level_1", "mean"),
```

	client_id	Unnamed:_0	disrict	client_catg	region	avg_consom_l_1_ELEC	avg_consom_l_1_GAZ	var_consom_l_1_ELEC	var_consom_l_1_GAZ	sd_consom_l_1_ELEC
0	train_Client_0	0	60	11	101	352.400000	0.000000	96313.070588	0.000000	310.343472
1	train_Client_1	1	69	11	107	557.540541	0.000000	39178.644144	0.000000	197.935960
2	train_Client_10	2	62	11	301	798.611111	0.000000	264032.957516	0.000000	513.841374
3	train_Client_100	3	69	11	105	1.200000	0.000000	13.010526	0.000000	3.607011
4	train_Client_1000	4	62	11	303	663.714286	0.000000	50549.142857	0.000000	224.831365
...	...	...	...	...	...	...	...	...	...	...
135488	train_Client_99995	135488	62	11	304	0.000000	4.088235	0.000000	568.264706	0.000000
135489	train_Client_99996	135489	63	11	311	309.700000	67.904762	49830.852632	3465.190476	223.228252
135490	train_Client_99997	135490	63	11	311	405.000000	65.785714	26984.857143	686.181319	164.270683
135491	train_Client_99998	135491	60	11	101	300.000000	0.000000	20000.000000	0.000000	141.421356
135492	train_Client_99999	135492	60	11	101	459.333333	0.000000	31992.333333	0.000000	178.864008

135493 rows × 137 columns

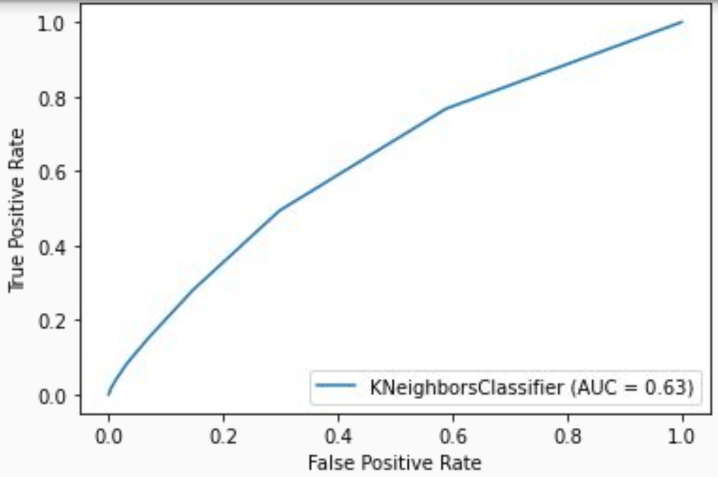
MACHINE LEARNING

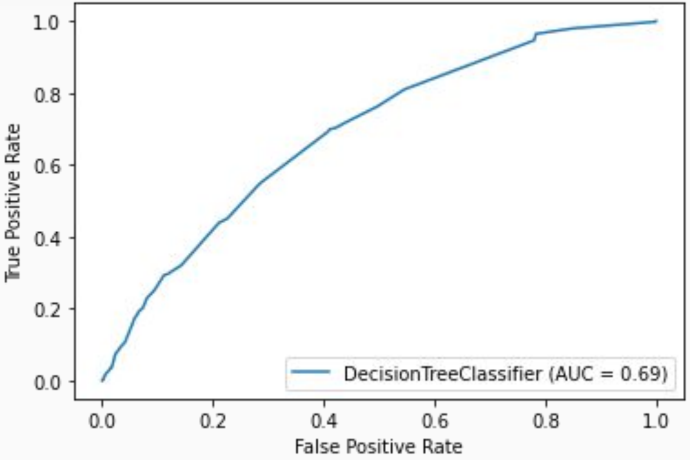
Building Machine Learning Models
Selecting Best Models
Hyper Parameter Tuning
Feature Selection

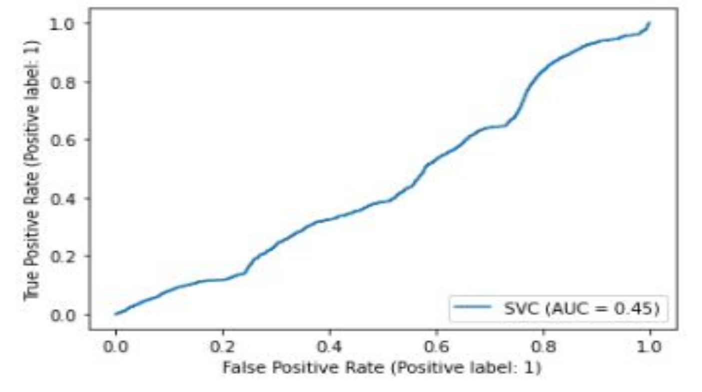
MACHINE LEARNING MODELS RESULTS BEFORE FEATURE ENGINEERING

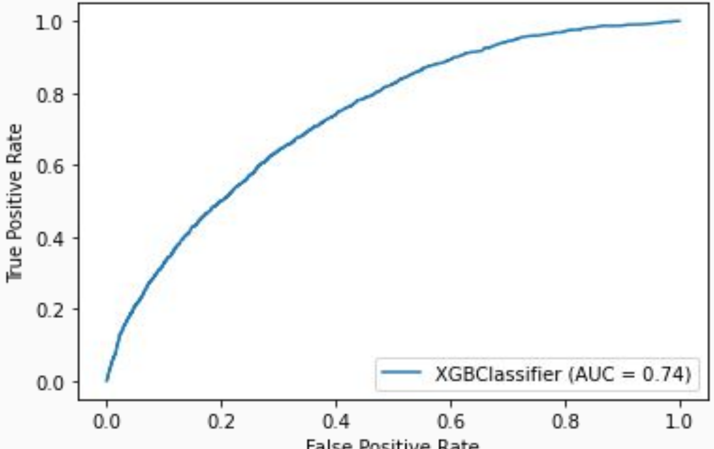
MODELS	ACCURACY	PRECISION	RECALL	AUC
Linear Reg	0.943	0.14	0.0568	067
Decision Tree	0.891	0.007	0.002	0.69
KNN	0.940	0.000	0.000	0.63
SVM	0.943	0.000	0.000	0.45
XG BOOST	0.944	0.22	0.002	0.74
ADA BOOST	0.943	0	0	0.69
CAT BOOST	0.944	0.29	0.017	0.72
LGBM BOOST	0.941	0.967	0.4	0.73

ROCs BEFORE FEATURE ENGINEERING



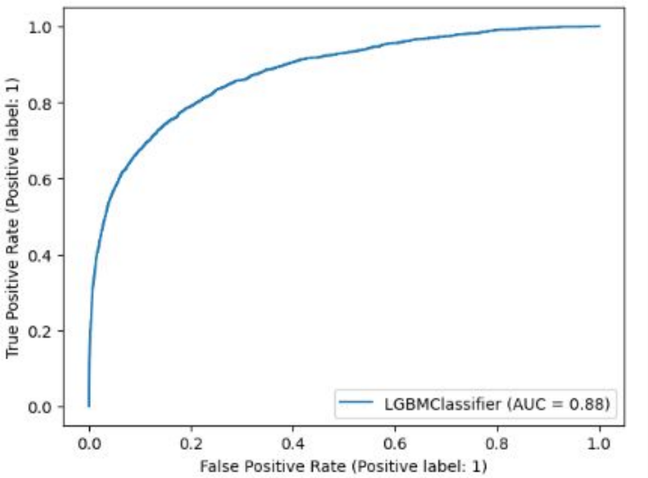


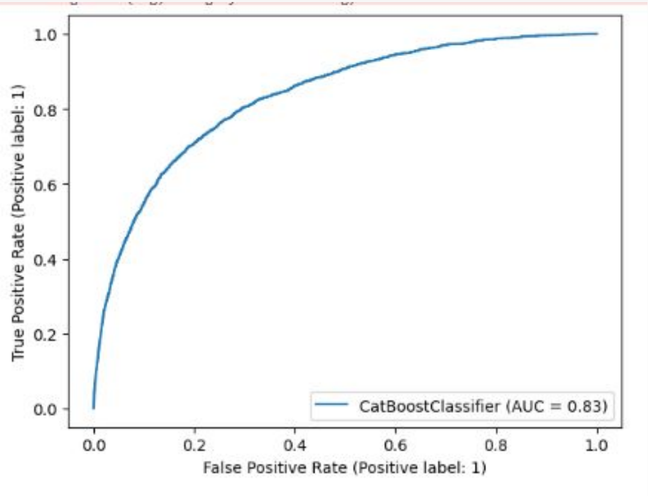


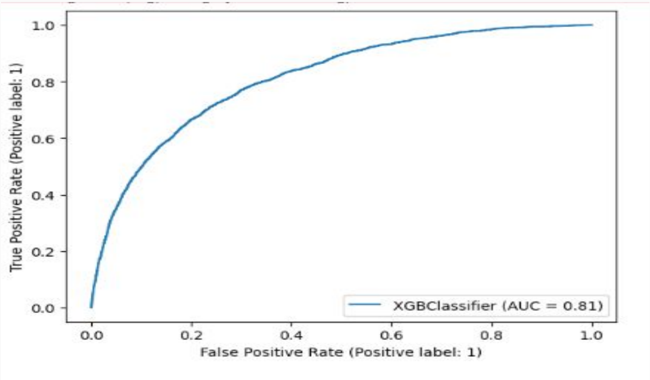


MODEL RESULTS AFTER FEATURE ENGINEERING

MODELS	ACCURACY	PRECISION	RECALL	AUC
XG BOOST	0.936	0.48	0.07	0.81
ADA BOOST	0.934	0	0	0.76
CAT BOOST	0.938	0.67	0.09	0.83
LGBM BOOST	0.961	0.95	0.4	0.88


1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1)
LGBMClassifier (AUC = 0.88)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)


1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1)
CatBoostClassifier (AUC = 0.83)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)


1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1)
XGBClassifier (AUC = 0.81)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)

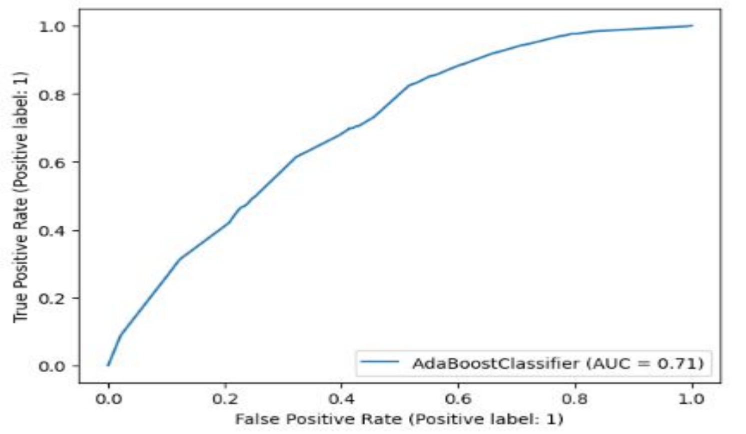
DATA BALANCING

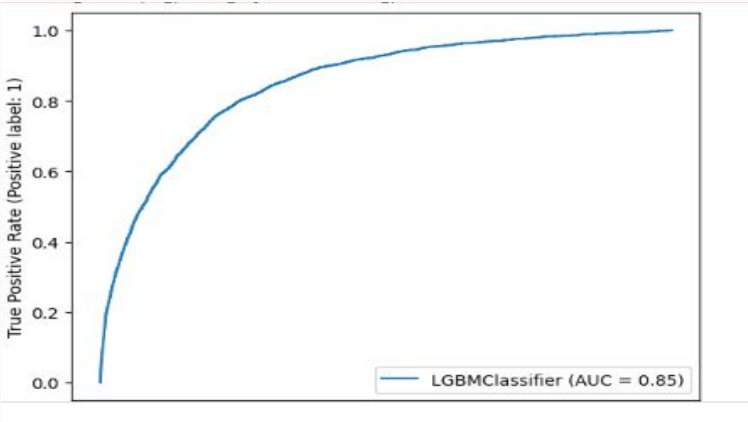
- Over Sampling
- Under Sampling

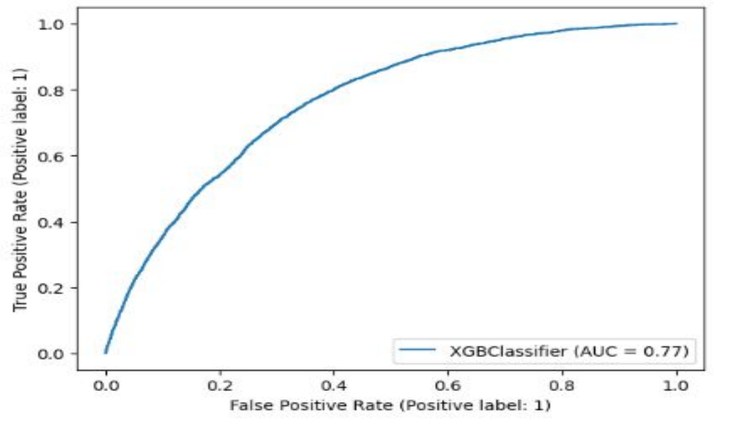
RESULTS AFTER OVER SAMPLING

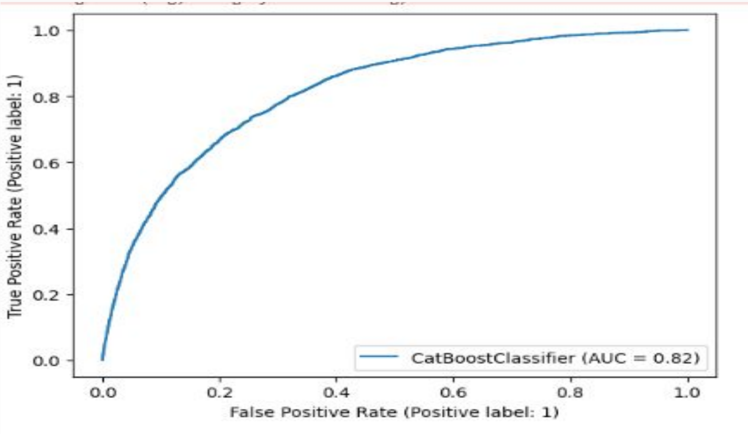
MODELS	ACCURACY	PRECISION	RECALL	AUC
XG BOOST	0.844	0.17	0.42	0.74
ADA BOOST	0.92	0.21	0.08	0.71
CAT BOOST	0.92	0.37	0.17	0.82
LGBM BOOST	0.87	0.27	0.58	0.85

ROC CURVES AFTER OVER SAMPLING





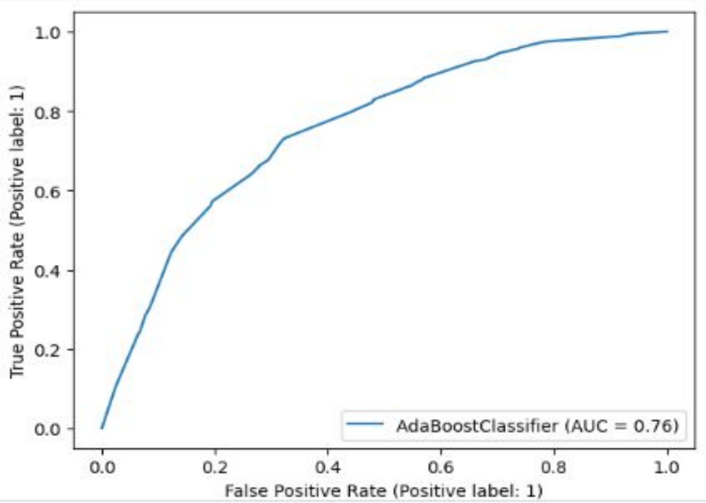


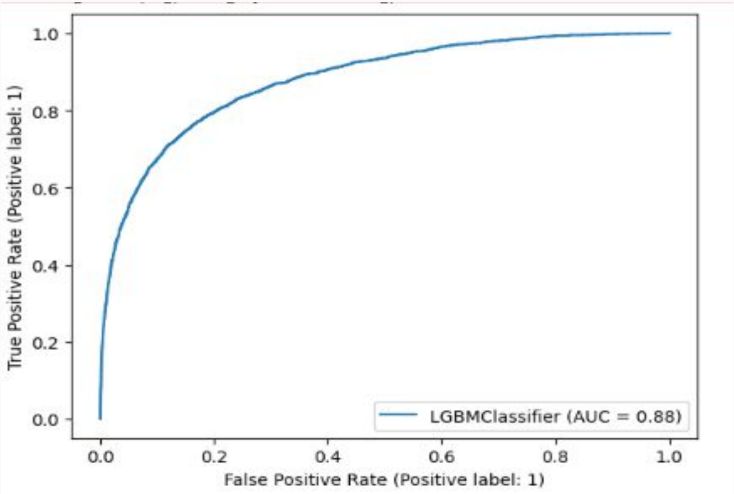


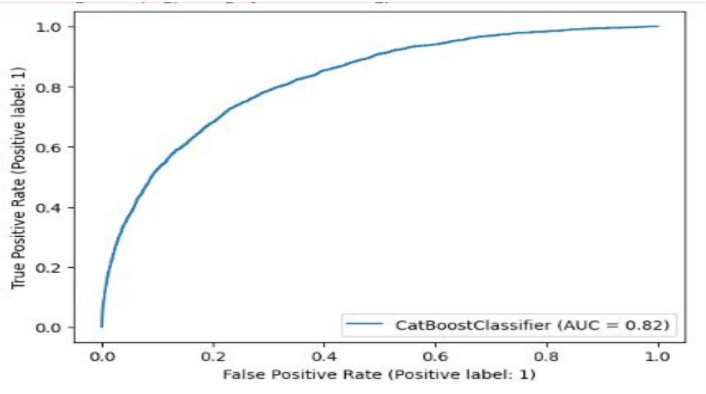
RESULTS AFTER UNDER SAMPLING

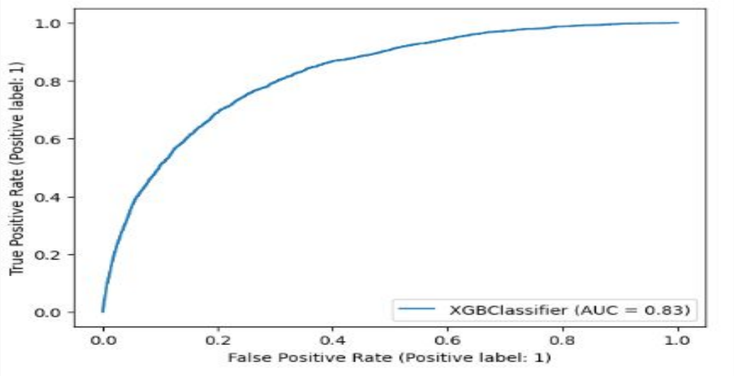
MODELS	ACCURACY	PRECISION	RECALL	AUC
XG BOOST	0.93	0.42	0.12	0.83
ADA BOOST	0.93	0.00	0.00	0.76
CAT BOOST	0.938	0.64	0.09	0.82
LGBM BOOST	0.94	0.88	0.4	0.88

ROC CURVES AFTER UNDER SAMPLING







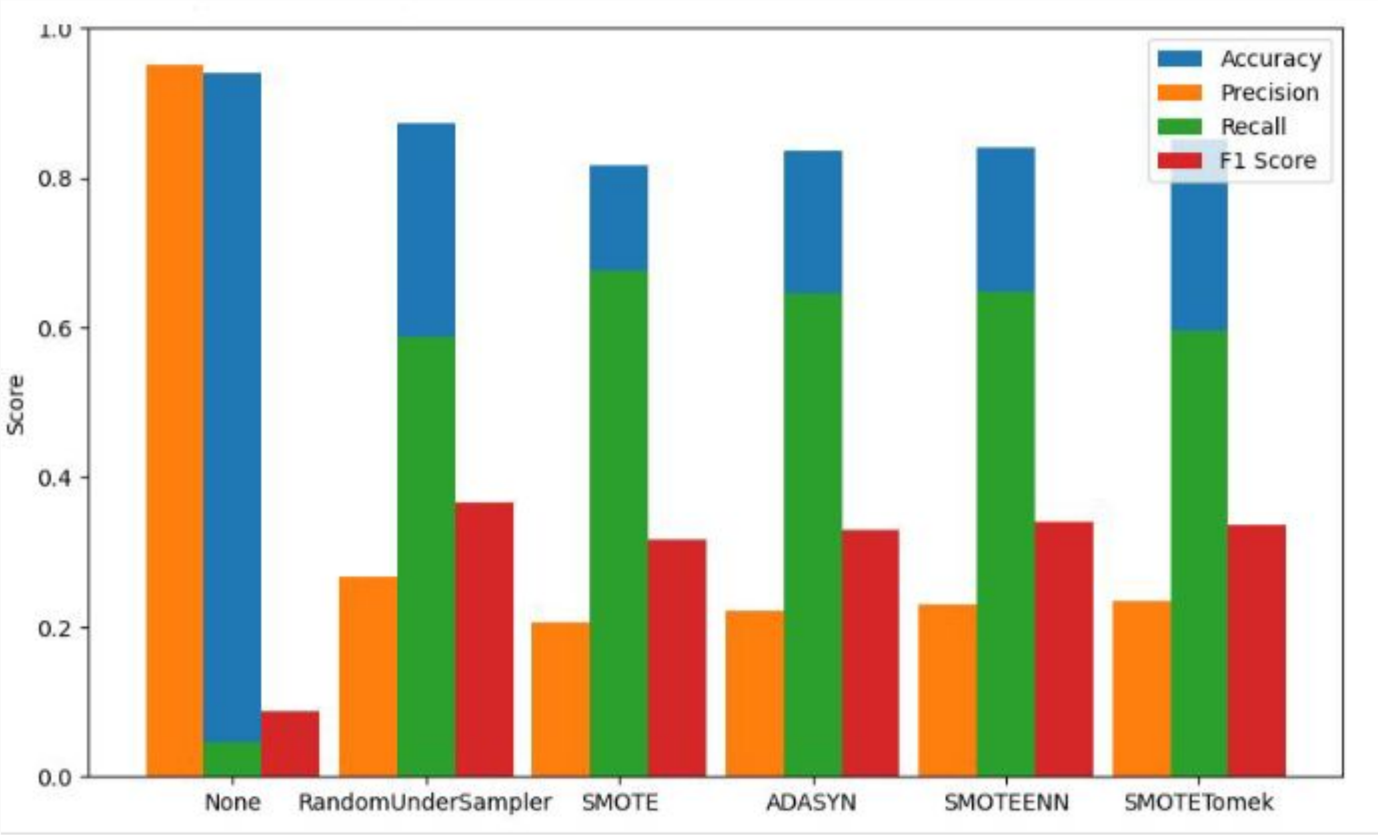


FEATURE SELECTION

- Selection K-Best

K-BEST FOR FEATURE SELECTION

SelectKBest is a feature selection method in machine learning that selects the K most significant features from a dataset based on a statistical test. It is a supervised learning technique that can be used for classification and regression problems.


1.0
0.8
0.6
0.4
0.2
0.0
Score
Accuracy
Precision
Recall
F1 Score
None
RandomUnderSampler
SMOTE
ADASYN
SMOTEENN
SMOTETomek

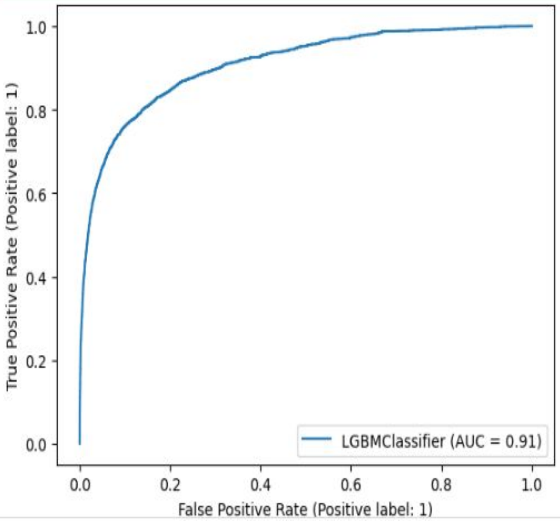
```
/opt/conda/lib/python3.7/site-packages/lightgbm/sklearn.py:598: UserWarning: 'silent' argument is deprecated and will be removed in a future release of LightGBM. Pass 'verbose' parameter via keyword arguments instead.
  _log_warning("'silent' argument is deprecated and will be removed in a future release of LightGBM. "
LGBMClassifier(learning_rate=0.0695305887282317, max_depth=14,
               min_child_samples=11, n_estimators=1954, num_leaves=16383,
               objective='binary', silent=True)
Results on test data
Test accuracy =  0.9412
Test precision =  0.9674
Test recall =  0.0689
Classification report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97     38399
           1       0.97      0.07      0.13      2582

    accuracy                           0.94     40981
   macro avg       0.95      0.53      0.55     40981
weighted avg       0.94      0.94      0.92     40981

Confusion matrix (Rows actual, Columns predicted):
       0    1
0  38393    6
1   2404  178

ROC curve
/opt/conda/lib/python3.7/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot_roc_curve is deprecated; Function :func:`plot_roc_curve` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: :meth:`sklearn.metric.RocCurve
play.from_predictions` or :meth:`sklearn.metric.RocCurveDisplay.from_estimator`.
  warnings.warn(msg, category=FutureWarning)
```



CONCLUSION

Our analysis has shown that the LGBM model is the most effective model in detecting fraud, with an accuracy of 94 percent, precision of 96 percent, and an AUC score of 91. Although the recall rate of 8 percent is relatively low, it is still considered satisfactory as the cost of false negatives is much higher than false positives.