Credit Card Fraud Detection

The problem that we want to solve is to detect when a client are going to commit a fraud

Research objectives

The following model analyze through different variable and methods the profile for Credit Card Fraud Detection
How are that features of the client that could commit fraud?
What are the types of application that apply more and have better range for be approved?

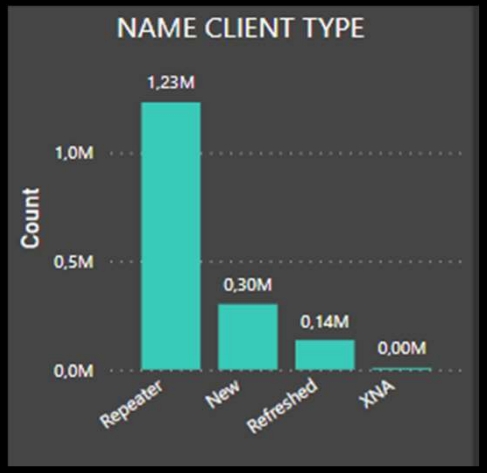
Data Acquisition

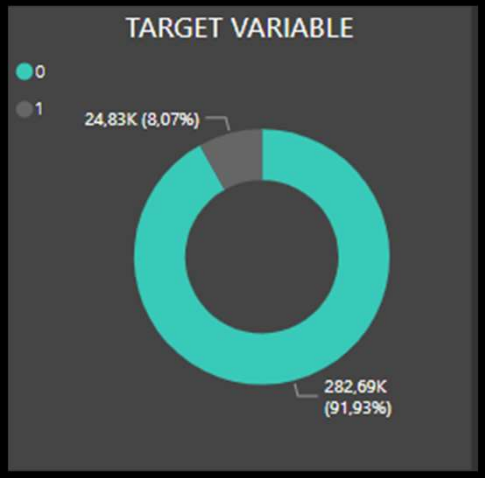
-previous_Application.csv: which contains all the variables that are the base of the model

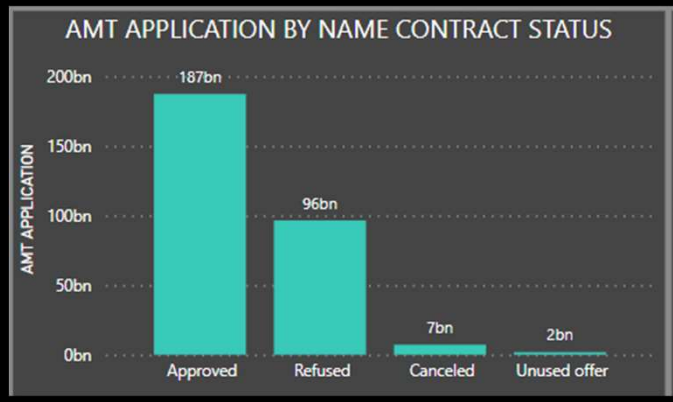
-application_data.csv: which contains other variables useful and some flags that help us for define client profile

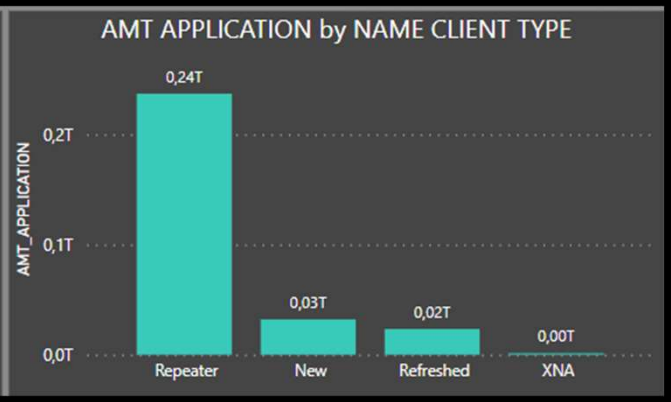
-columns_descrption.xlsx: which contains the description of all variables

UNIVARIATE





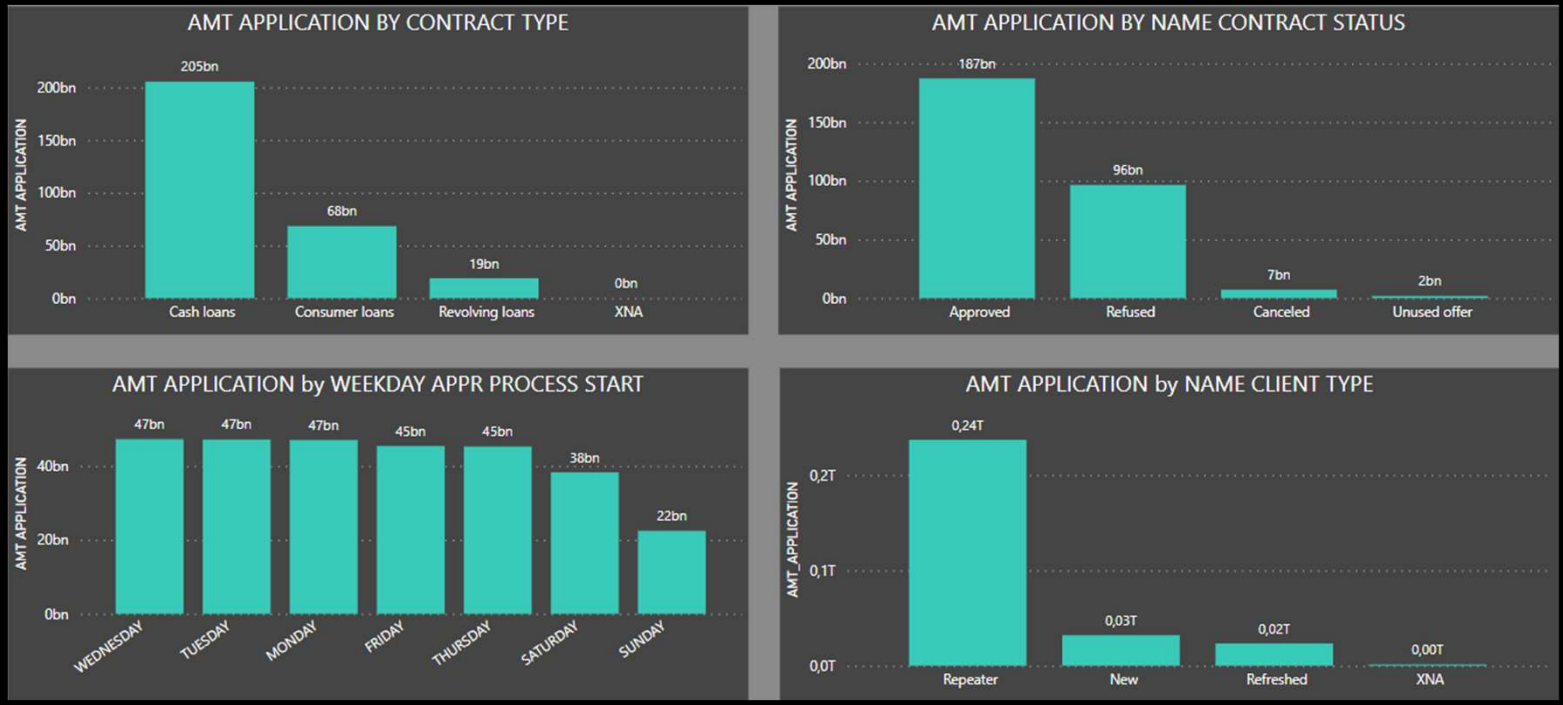




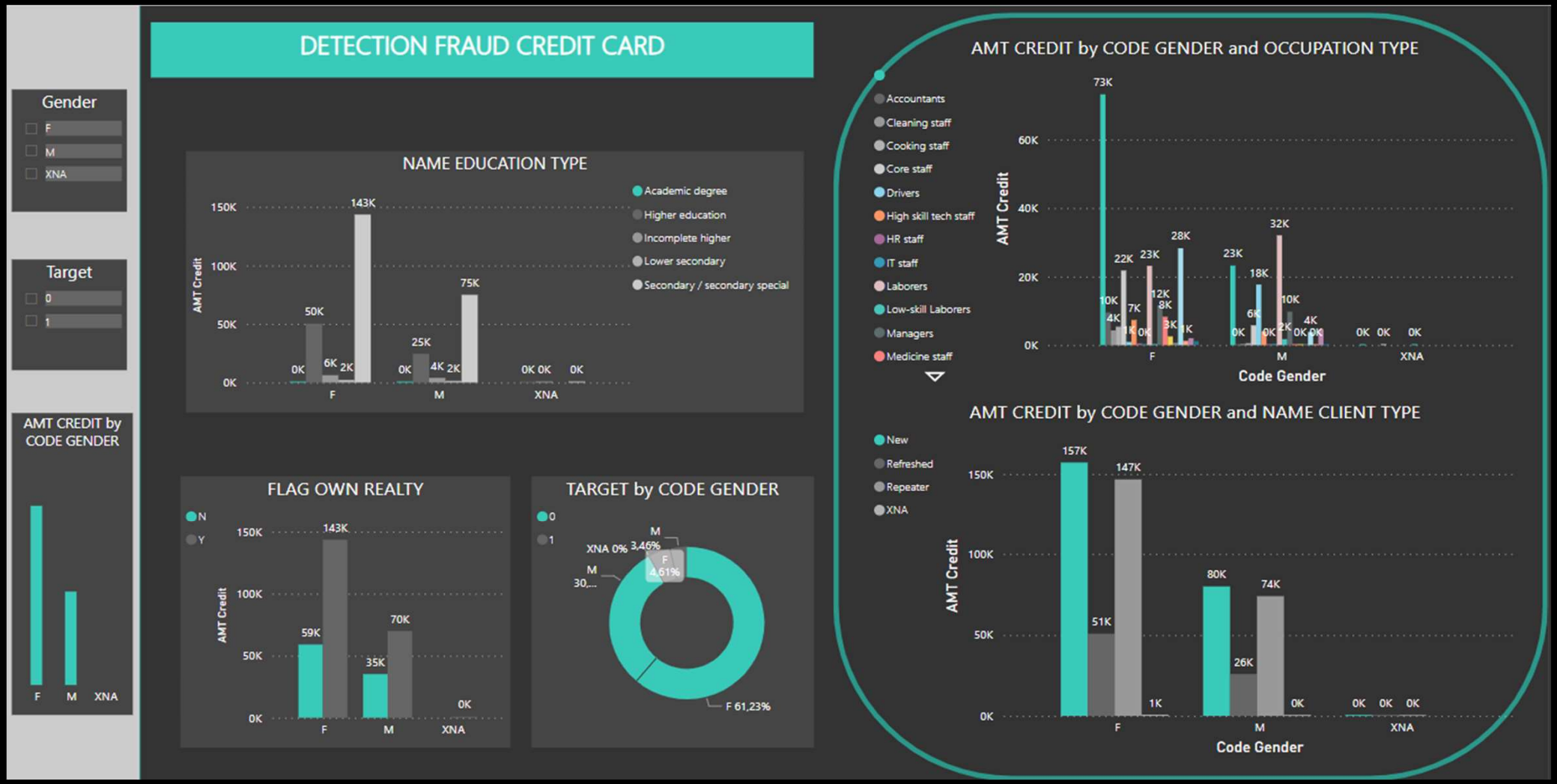
To understand the dataset here watch some variables with a bivariate analysis, plotting some variables against others

- Most of the credits with the highest amounts are cash loans
- The status approved of the credit have also the highest amounst and almost all of that application are client that already have previous application

Bivariate



Multivariate



Highlights

After analice different combination of the principal variables some conclusions:

- Most of loans are requested by the female gender
- Because the majority that request the loans are of the female gender, they have a higher percentage of the possibility of committing fraud when evaluating the target variable with respect to gender
- It can also be seen that most applicants have a maximum educational level of secondary school and less than half do not have a house or flat within their properties
- Finally, it can be seen that most of the clients are new and repeater and as already mentioned most og them are part of the female gender.

Algorithm and Model Training

Comparative analysis between algorithms

1) **KNN:** in this case the model present overfitting, while more neighbors were assigned it did not improve, the model was adapting to the dataset and not predicting

It can also be seen that on the surface it does not result in bad indicators but let see more algorithms for choose the best option

```
confusion_matrix(y_test,y_pred)
```

```
array([[27358,   214],
       [ 2395,    33]], dtype=int64)
```

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.92	0.99	0.95	27572
1	0.13	0.01	0.02	2428
accuracy			0.91	30000
macro avg	0.53	0.50	0.49	30000
weighted avg	0.86	0.91	0.88	30000

Algorithm and Model Training

2) Decision Tree: for this model we perform some optimizations and evaluate the performance before and after. We get better results if we compare it with the previous model and because of this it seems like a better model, more acceptable for the type of problem we are working on

Without stratify param

```
confusion_matrix(y_test,y_pred)
```

```
array([[27606,     1],
       [ 2393,     0]], dtype=int64)
```

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.92	1.00	0.96	27607
1	0.00	0.00	0.00	2393
accuracy			0.92	30000
macro avg	0.46	0.50	0.48	30000
weighted avg	0.85	0.92	0.88	30000

With stratify param

```
confusion_matrix(y_test,y_pred)
```

```
array([[27570,     2],
       [ 2428,     0]], dtype=int64)
```

```
print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.92	1.00	0.96	27572
1	0.00	0.00	0.00	2428
accuracy			0.92	30000
macro avg	0.46	0.50	0.48	30000
weighted avg	0.84	0.92	0.88	30000

Algorithm and Model Training

3) Lineal Regression: After evaluating this model we can see that it is not the type of algorithm that should be used for this case since our target variable is binary and it cannot be predicted with linear regression

We also calculate some metrics

```
Mean Absolute Error: 0.14849517150900227
Mean Squared Error: 0.07520691208552975
Root Mean Squared Error: 0.274238786617666
```

```
% de aciertos sobre el set de entrenamiento: 0.005149101716224291
% de aciertos sobre el set de evaluación: -0.011074866155084795
```

Algorithm and Model Training

After evaluating 3 models, we conclude that the one that best predict this type of dataset is the **Decision tree**, taking into account the metrics in addition to having an acceptable accuracy for the type of problema

As a last evaluation of performance, we created a comparative table between test and training of each model

Model	Train	Test
Decision Tree	0.918586	0.920200
Decision Tree Optimized	0.919114	0.919000
KNN	0.919071	0.919067
Linear Regression	0.005149	-0.011075