Credit Card Approval Prediction

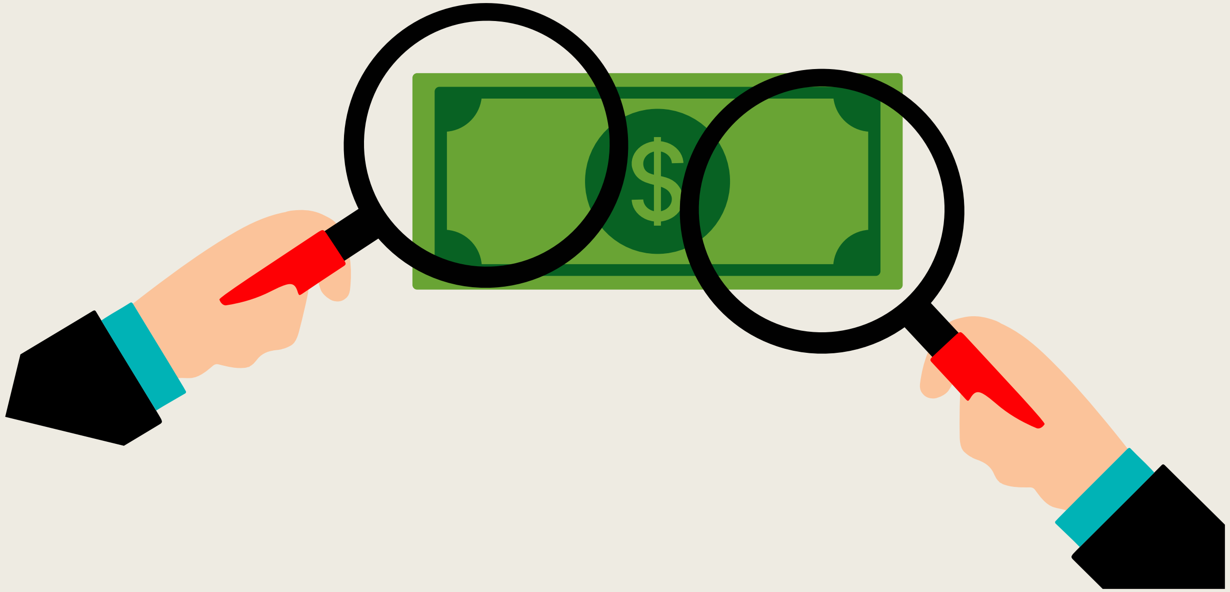
ANASTASIA DRAKOU 2022202304006

DESPOINA ANGELIKI MOYSIDOU 2022202304016

ANNA KOUTOUGERA 2022202304012

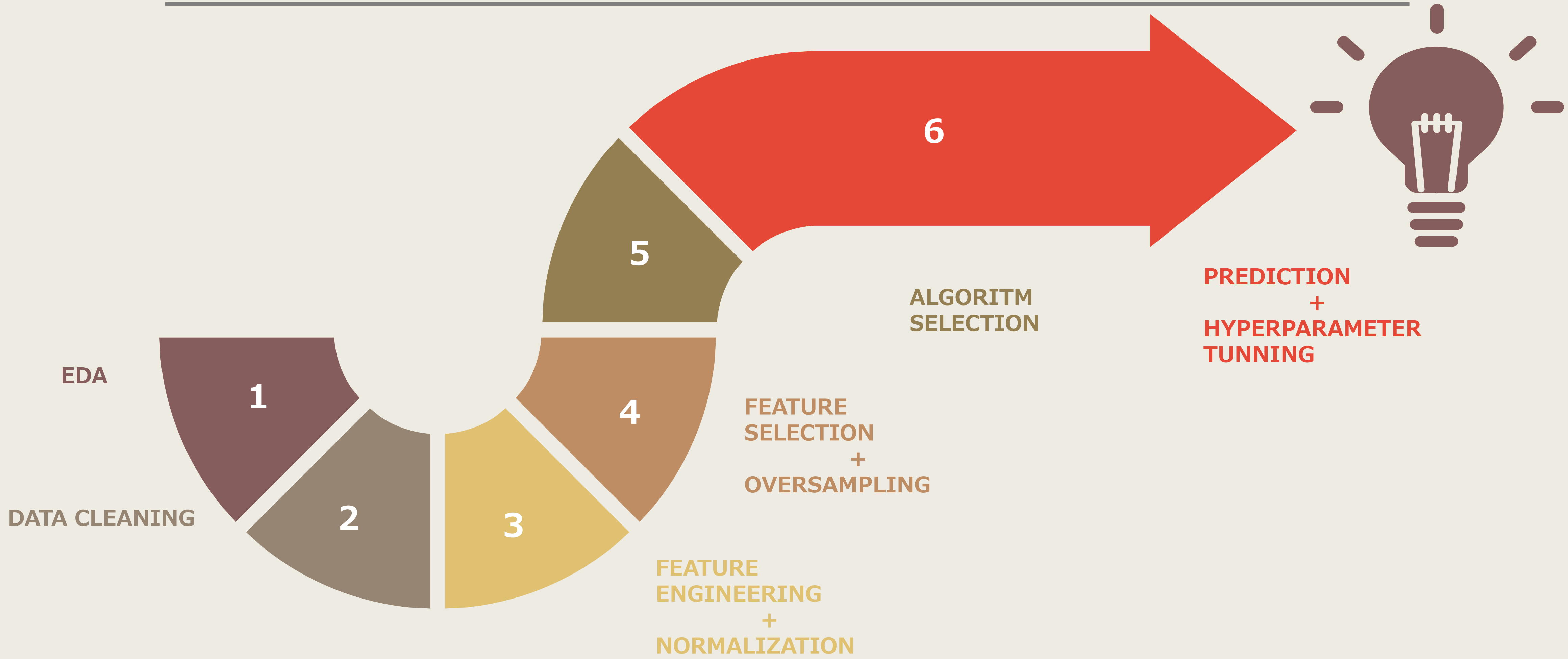
Our Goal

The Credit Card Approval Prediction dataset aims to predict whether a credit card application will be approved or denied based on various features



1. PEEK AT THE DATA
2. CREATION OF TARGET FEATURE
3. DATA CLEANING / FEATURE SELECTION/OVERSAMPLING
4. BUILD A MACHINE LEARNING MODEL FOR PREDICTION

WORKFLOW


1
EDA
2
DATA CLEANING
3
FEATURE ENGINEERING
+
NORMALIZATION
4
FEATURE SELECTION
+
OVERSAMPLING
5
ALGORITM SELECTION
6
PREDICTION
+
HYPERPARAMETER TUNNING

PEEK AT THE DATA

1

APPLICATION RECORD

NUMERICAL DATA
CATEGORICAL DATA
ORDINAL DATA

Contains appliers personal information, which we could use as features for predicting.

	0	1	2	3	4
ID	5008804	5008805	5008806	5008808	5008809
CODE_GENDER	M	M	M	F	F
FLAG_OWN_CAR	Y	Y	Y	N	N
FLAG_OWN_REALTY	Y	Y	Y	Y	Y
CNT_CHILDREN	0	0	0	0	0
AMT_INCOME_TOTAL	427500.0	427500.0	112500.0	270000.0	270000.0
NAME_INCOME_TYPE	Working	Working	Working	Commercial associate	Commercial associate
NAME_EDUCATION_TYPE	Higher education	Higher education	Secondary / secondary special	Secondary / secondary special	Secondary / secondary special
NAME_FAMILY_STATUS	Civil marriage	Civil marriage	Married	Single / not married	Single / not married
NAME_HOUSING_TYPE	Rented apartment	Rented apartment	House / apartment	House / apartment	House / apartment
DAYS_BIRTH	-12005	-12005	-21474	-19110	-19110
DAYS_EMPLOYED	-4542	-4542	-1134	-3051	-3051
FLAG_MOBIL	1	1	1	1	1
FLAG_WORK_PHONE	1	1	0	0	0
FLAG_PHONE	0	0	0	1	1
FLAG_EMAIL	0	0	0	1	1
OCCUPATION_TYPE	NaN	NaN	Security staff	Sales staff	Sales staff
CNT_FAM_MEMBERS	2.0	2.0	2.0	1.0	1.0

PEEK AT THE DATA

1

APPLICATION RECORD

NUMERICAL DATA
CATEGORICAL DATA
ORDINAL DATA

2

CREDIT RECORD

NUMERICAL DATA
CATEGORICAL DATA

In this table, a person - month record identifies a row. Every row represents a client's condition in different months

	ID	MONTHS_BALANCE	STATUS
0	5001711	0	X
1	5001711	-1	0
2	5001711	-2	0
3	5001711	-3	0
4	5001712	0	C
5	5001712	-1	C
6	5001712	-2	C
7	5001712	-3	C
8	5001712	-4	C
9	5001712	-5	C

PEEK AT THE DATA

1

APPLICATION RECORD

NUMERICAL DATA
CATEGORICAL DATA
ORDINAL DATA

2

CREDIT RECORD

NUMERICAL DATA
CATEGORICAL DATA

3

NUMBERS INSTANCES

438.510 unique clients from APPLICATION RECORD

45.985 unique clients from CREDIT RECORD

36.457 common clients

PEEK AT THE DATA

1 **APPLICATION RECORD**

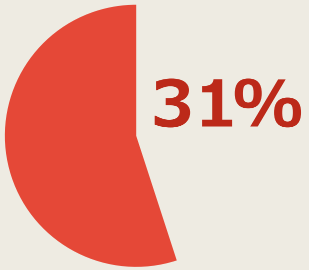
NUMERICAL DATA
CATEGORICAL DATA
ORDINAL DATA

2 **CREDIT RECORD**

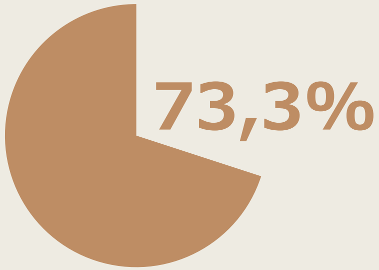
NUMERICAL DATA
CATEGORICAL DATA

3 **NUMBERS INSTANCES**

4 **MISSING VALUES AND DUPLICATES**

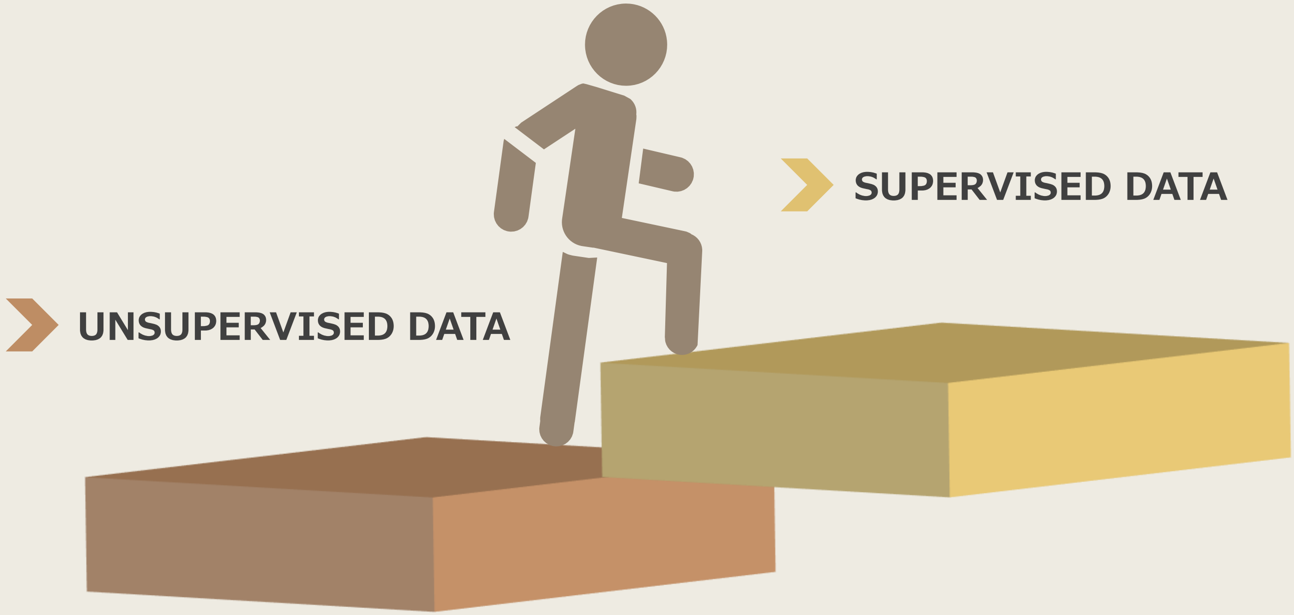


- "Occupation Type" column has 11,323 missing values.
- This represents around 31% of the total 36,457 filtered records.
- Retaining the "Occupation Type" column is crucial due to its potential significance in understanding and predicting process



- Each record is uniquely identified by the ID column, leading to the perception of no duplicate values upon initial observation.
- Excluding the ID column, reveals a presence of 26,729 duplicate values, constituting 73.3% of the dataset.

Target generation



Target production

Vintage Analysis

- Vintage analysis is a widely used method in credit risk management.
- Provides a dynamic understanding of credit portfolio performance.
- Identifies patterns in the emergence of bad customers over different periods.
- Facilitates proactive risk management strategies based on historical trends
- Evaluate the performance of customers in defined time intervals post loan or credit issuance.
- Aggregate the cumulative percentage of customers exhibiting unfavorable outcomes within each time window
- It assesses the performance of a portfolio over distinct periods post the issuance of a loan or credit card
- Calculate the cumulative percentage of bad customers within specific performance windows.
- Create a bad customer ratio based on historical data, offering insights into the evolving risk over time.

Target generation

Vintage Analysis

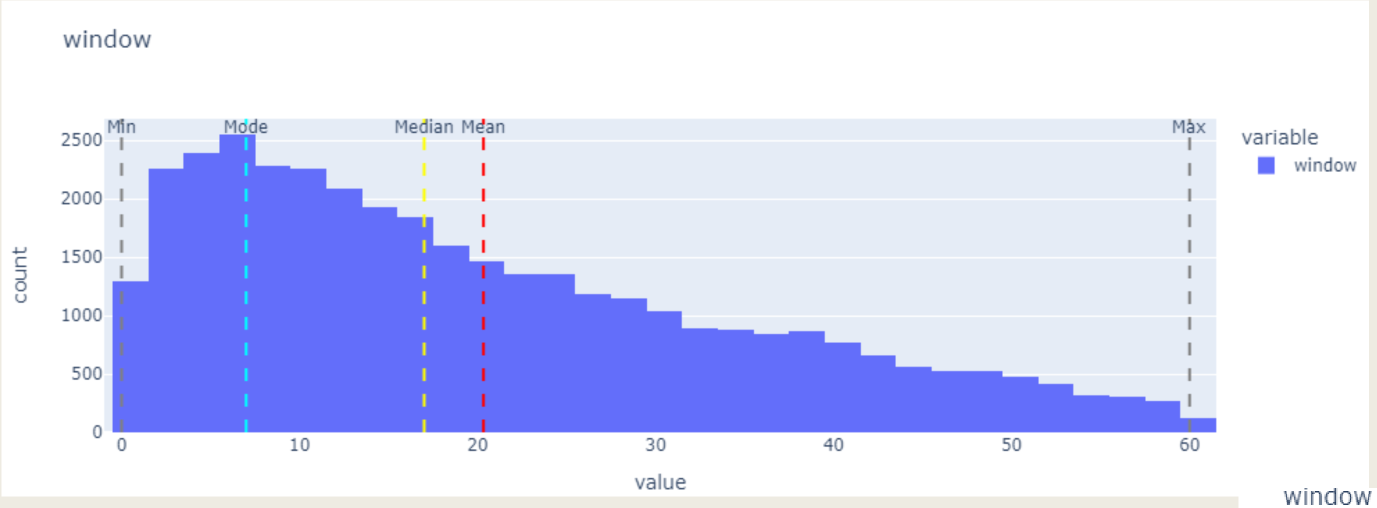
STEP 1

We want to keep the performance window most common in all cases. We don't simply look at the most recent data, because as we can see for certain clients, we may have last payment information from 2 years ago.

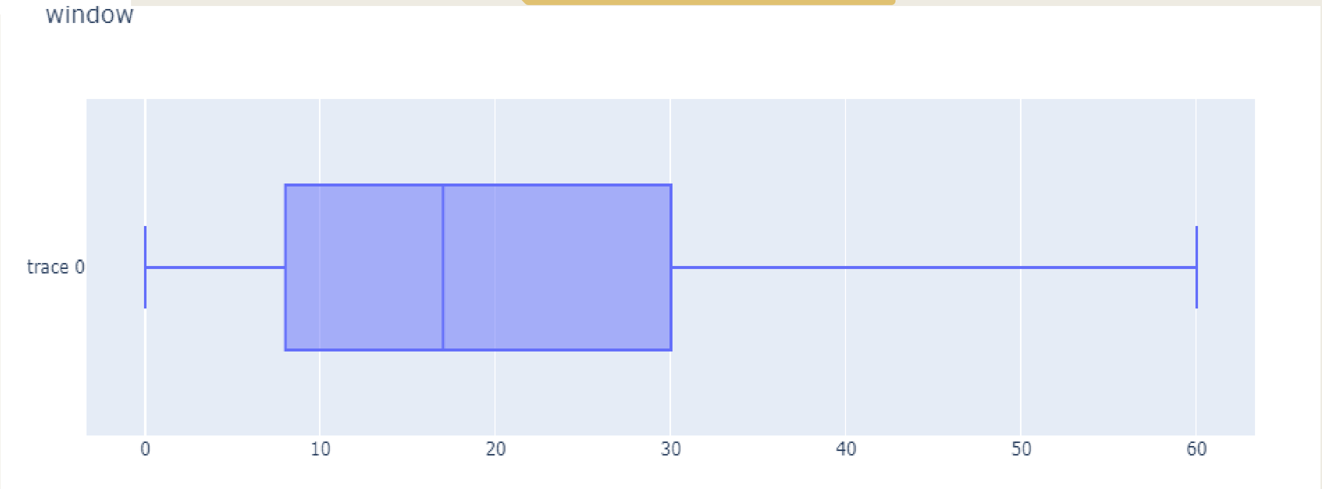
MONTHS_BALANCE	ID	open_month	end_month	window
0	5008804	-15	0	15
1	5008805	-14	0	14
2	5008806	-29	0	29
3	5008808	-4	0	4
4	5008809	-26	-22	4

Target generation

Let's visualize it and see the mode and median value



0	MIN
60	MAX
7	MODE
20	MEAN
17	MEDIAN



We can observe that applicants typically delay making their payments until after the 7-month period.

Target generation

Vintage Analysis

STEP 1

We want to keep the performance window most common in all cases. We don't simply look at the most recent data, because as we can see for certain clients, we may have last payment information from 2 years ago.

STEP 2

Calculate ratios

Target generation

Vintage Analysis

STEP 1 We want to keep the performance window most common in all cases. We don't simply look at the most recent data, because as we can see for certain clients, we may have last payment information from 2 years ago.

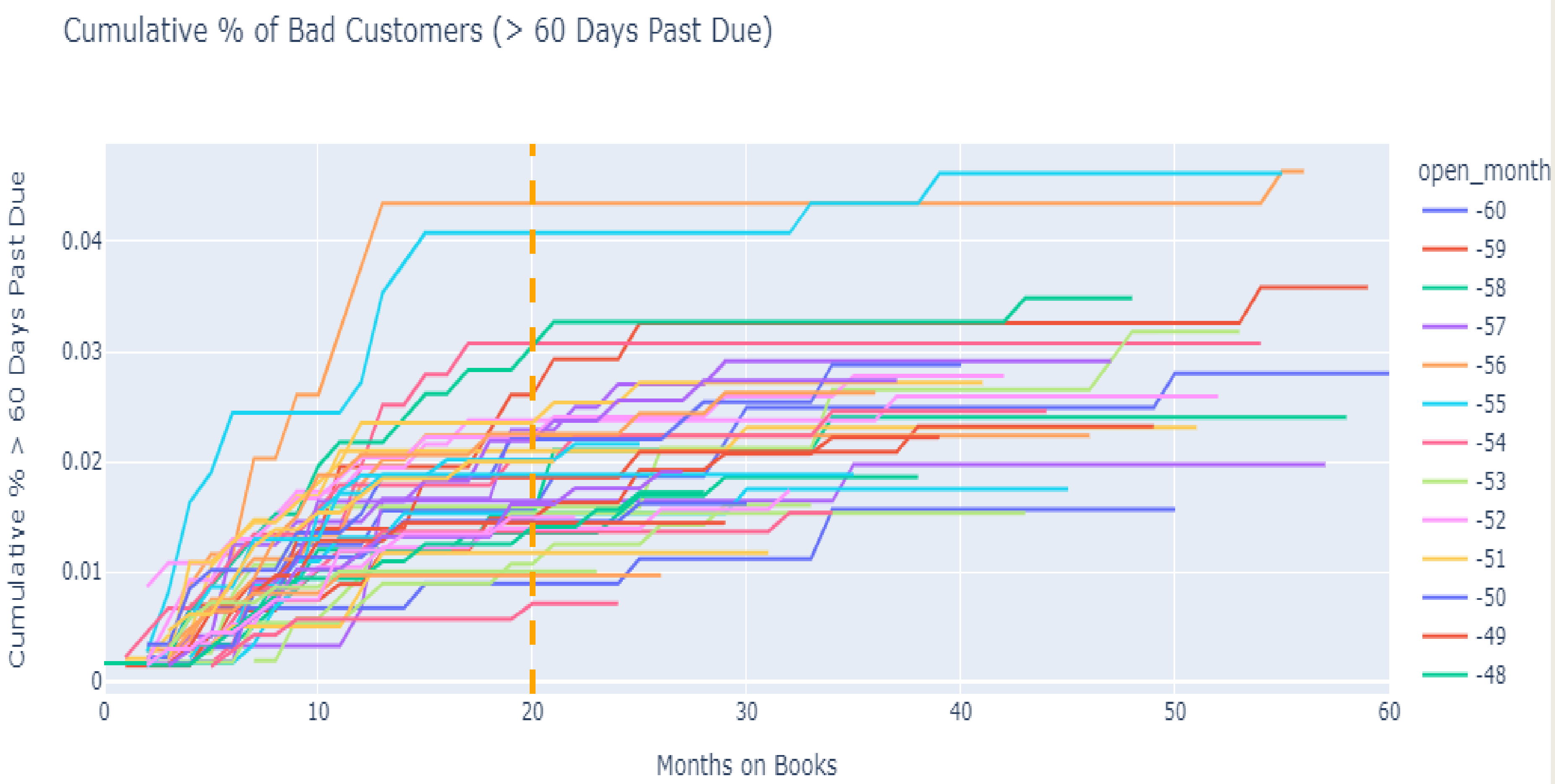
STEP 2 Calculate ratios

STEP 3 Analyzing Bad Customers

	ID	MONTHS_BALANCE	STATUS	open_month	end_month	window
0	5008804	0	C	-15	0	15
1	5008804	-1	C	-15	0	15
2	5008804	-2	C	-15	0	15
3	5008804	-3	C	-15	0	15
4	5008804	-4	C	-15	0	15
...	...	...	...	...	...	...
777710	5150487	-25	C	-29	0	29
777711	5150487	-26	C	-29	0	29
777712	5150487	-27	C	-29	0	29
777713	5150487	-28	C	-29	0	29
777714	5150487	-29	C	-29	0	29

Target generation

Let's say that we consider someone a bad client if they have a payment overdue more than 60 days



In this situation, if a client becomes high risk after 60 days of overdue payment, it's observed that things settle down after about 20 months. After this time, there's usually no significant new information, making a 20-month timeframe suitable for making a confident decision

Target generation

Vintage Analysis

STEP 1

We want to keep the performance window most common in all cases. We don't simply look at the most recent data, because as we can see for certain clients, we may have last payment information from 2 years ago.

STEP 2

Calculate ratios

STEP 3

Analyzing Bad Customers

STEP 4

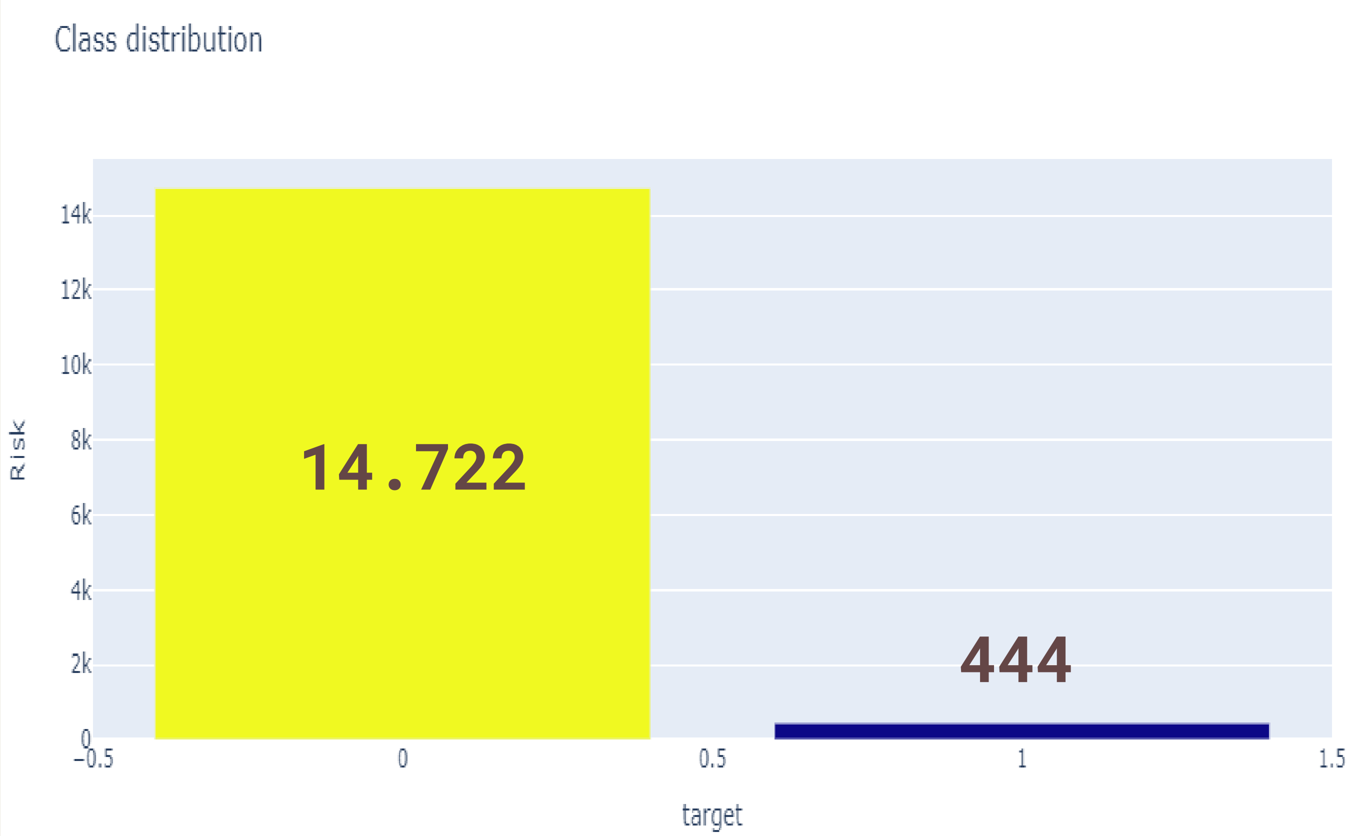
Target Column Creation

- **Low Risk vs. High Risk:** Research indicates that users late on payments by 30 days or more in any month are classified as 'high risk'. Those who do not exhibit this behavior are labeled as 'low risk' credit users.

- **Default Criteria:** A customer is considered 'bad' if they default by being 90 days or more past due within a 20-month performance window. The decision to use this timeframe is based on analysis and practical experience, focusing on identifying high risk through payments overdue by more than 60 days.

Target generation

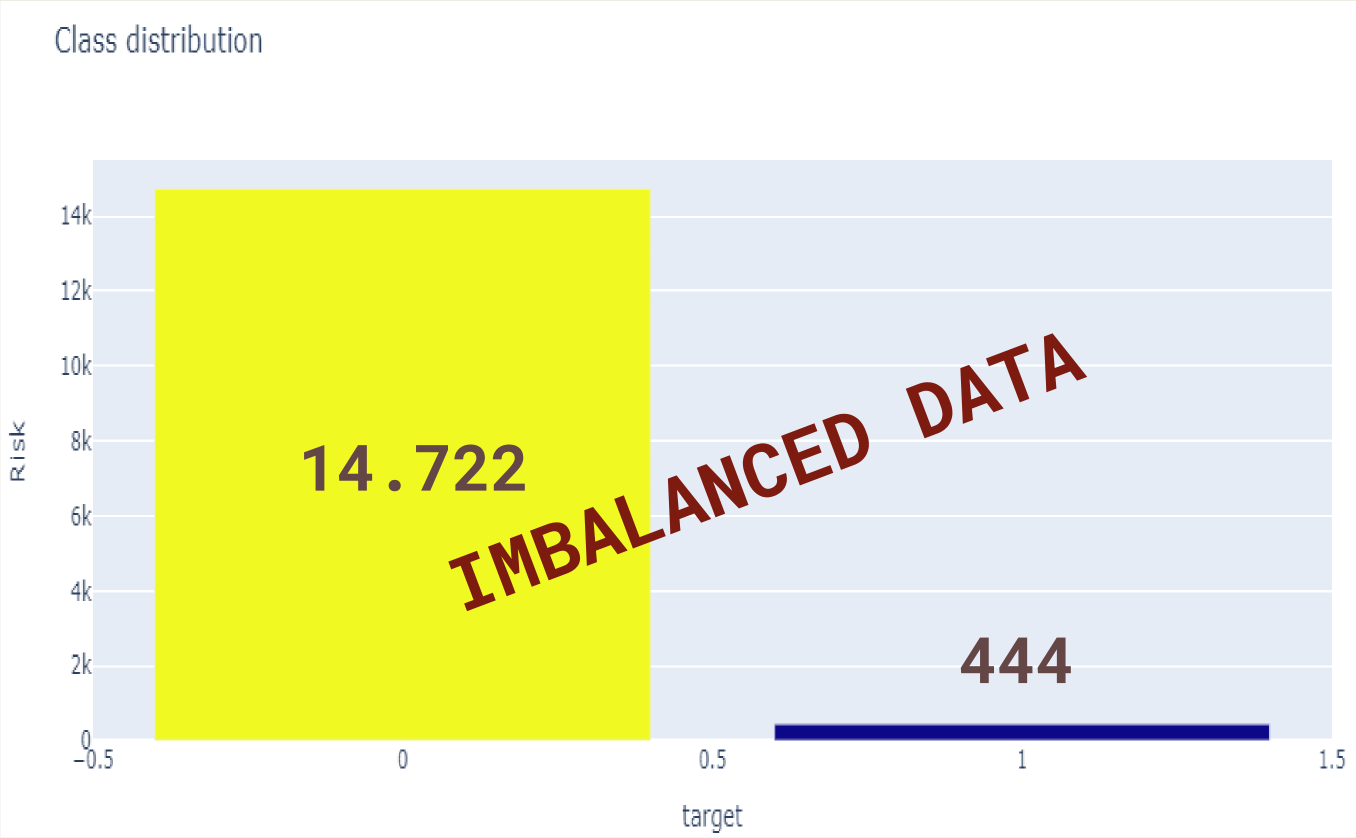
Now that we have the targets, let's check our class distribution



	0	1	2	3	4
ID	5008806	5008810	5008811	5112956	5008825
CODE_GENDER	M	F	F	M	F
FLAG_OWN_CAR	Y	N	N	Y	Y
FLAG_OWN_REALTY	Y	Y	Y	Y	N
CNT_CHILDREN	0	0	0	0	0
AMT_INCOME_TOTAL	112500.0	270000.0	270000.0	270000.0	130500.0
NAME_INCOME_TYPE	Working	Commercial associate	Commercial associate	Working	Working
NAME_EDUCATION_TYPE	Secondary / secondary special	Secondary / secondary special	Secondary / secondary special	Higher education	Incomplete higher
NAME_FAMILY_STATUS	Married	Single / not married	Single / not married	Married	Married
NAME_HOUSING_TYPE	House / apartment	House / apartment	House / apartment	House / apartment	House / apartment
DAYS_BIRTH	-21474	-19110	-19110	-16872	-10669
DAYS_EMPLOYED	-1134	-3051	-3051	-769	-1103
FLAG_MOBIL	1	1	1	1	1
FLAG_WORK_PHONE	0	0	0	1	0
FLAG_PHONE	0	1	1	1	0
FLAG_EMAIL	0	1	1	1	0
OCCUPATION_TYPE	Security staff	Sales staff	Sales staff	Accountants	Accountants
CNT_FAM_MEMBERS	2.0	1.0	1.0	2.0	2.0
target	0	0	0	0	0

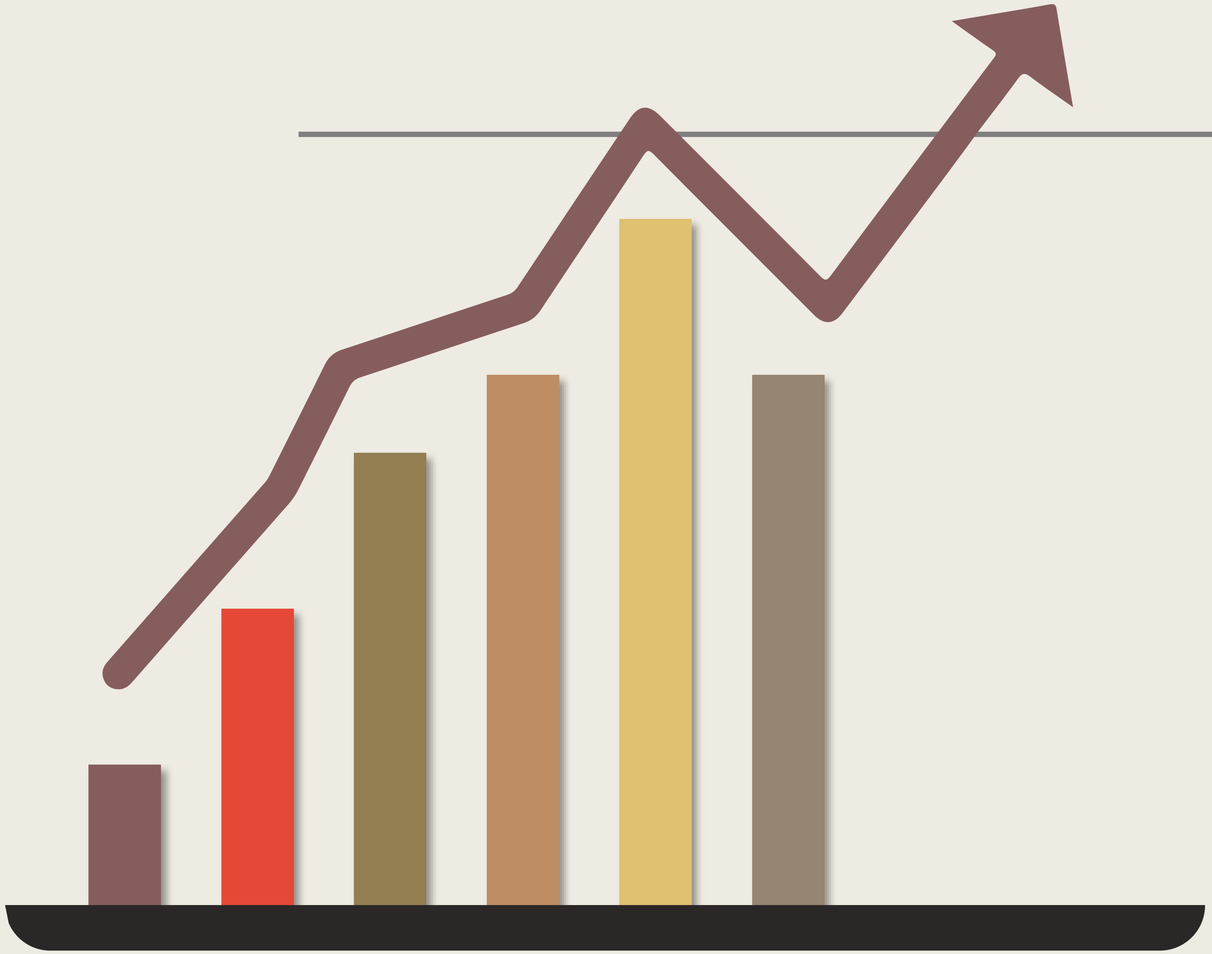
Target generation

Now that we have the targets, let's check our class distribution



	0	1	2	3	4
ID	5008806	5008810	5008811	5112956	5008825
CODE_GENDER	M	F	F	M	F
FLAG_OWN_CAR	Y	N	N	Y	Y
FLAG_OWN_REALTY	Y	Y	Y	Y	N
CNT_CHILDREN	0	0	0	0	0
AMT_INCOME_TOTAL	112500.0	270000.0	270000.0	270000.0	130500.0
NAME_INCOME_TYPE	Working	Commercial associate	Commercial associate	Working	Working
NAME_EDUCATION_TYPE	Secondary / secondary special	Secondary / secondary special	Secondary / secondary special	Higher education	Incomplete higher
NAME_FAMILY_STATUS	Married	Single / not married	Single / not married	Married	Married
NAME_HOUSING_TYPE	House / apartment	House / apartment	House / apartment	House / apartment	House / apartment
DAYS_BIRTH	-21474	-19110	-19110	-16872	-10669
DAYS_EMPLOYED	-1134	-3051	-3051	-769	-1103
FLAG_MOBIL	1	1	1	1	1
FLAG_WORK_PHONE	0	0	0	1	0
FLAG_PHONE	0	1	1	1	0
FLAG_EMAIL	0	1	1	1	0
OCCUPATION_TYPE	Security staff	Sales staff	Sales staff	Accountants	Accountants
CNT_FAM_MEMBERS	2.0	1.0	1.0	2.0	2.0
target	0	0	0	0	0

EXPLORATORY DATA ANALYSIS



Continuous Features

AMT_INCOME_TOTAL

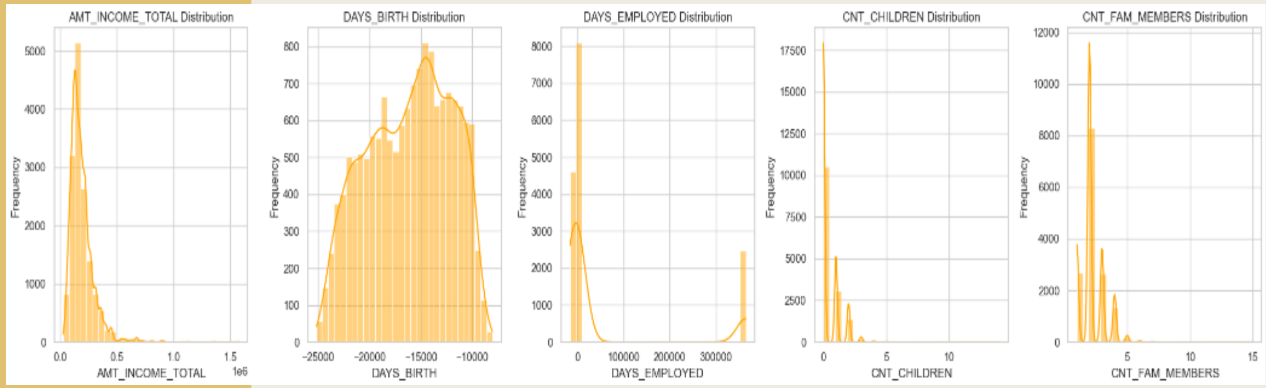
DAYS_BIRTH

DAYS_EMPLOYED

CNT_CHILDREN

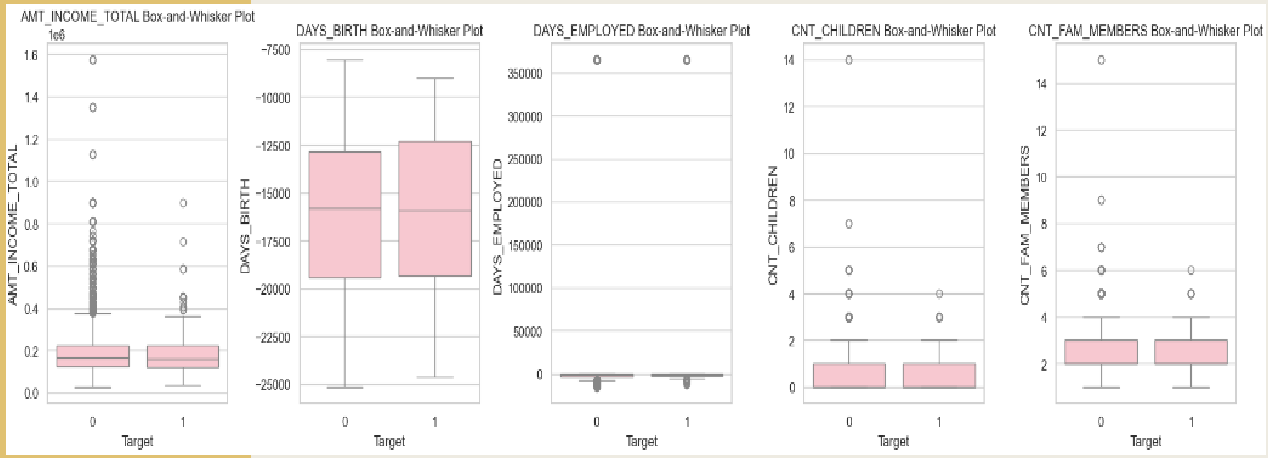
CNT_FAM_MEMBERS

Univariate feature plots



Histogram and Density plots showing each attribute's frequency

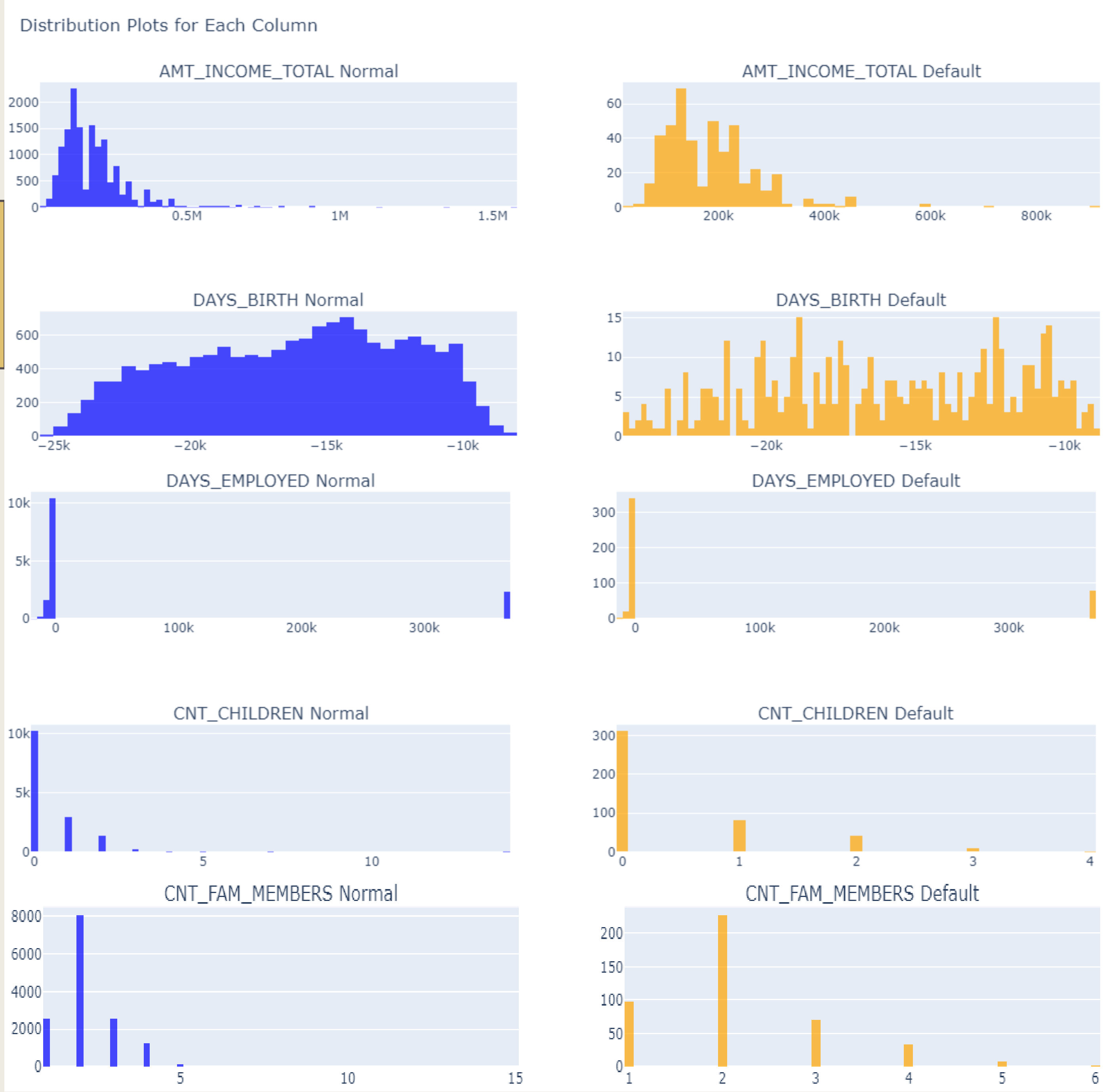
- Looking on the DAYS_EMPLOYED plot we can clearly see that there are some outliers
- It is difficult for us to simply remove those outliers due to their high frequency
- We will investigate later the way we will use them on the data cleaning process



Whisker plot

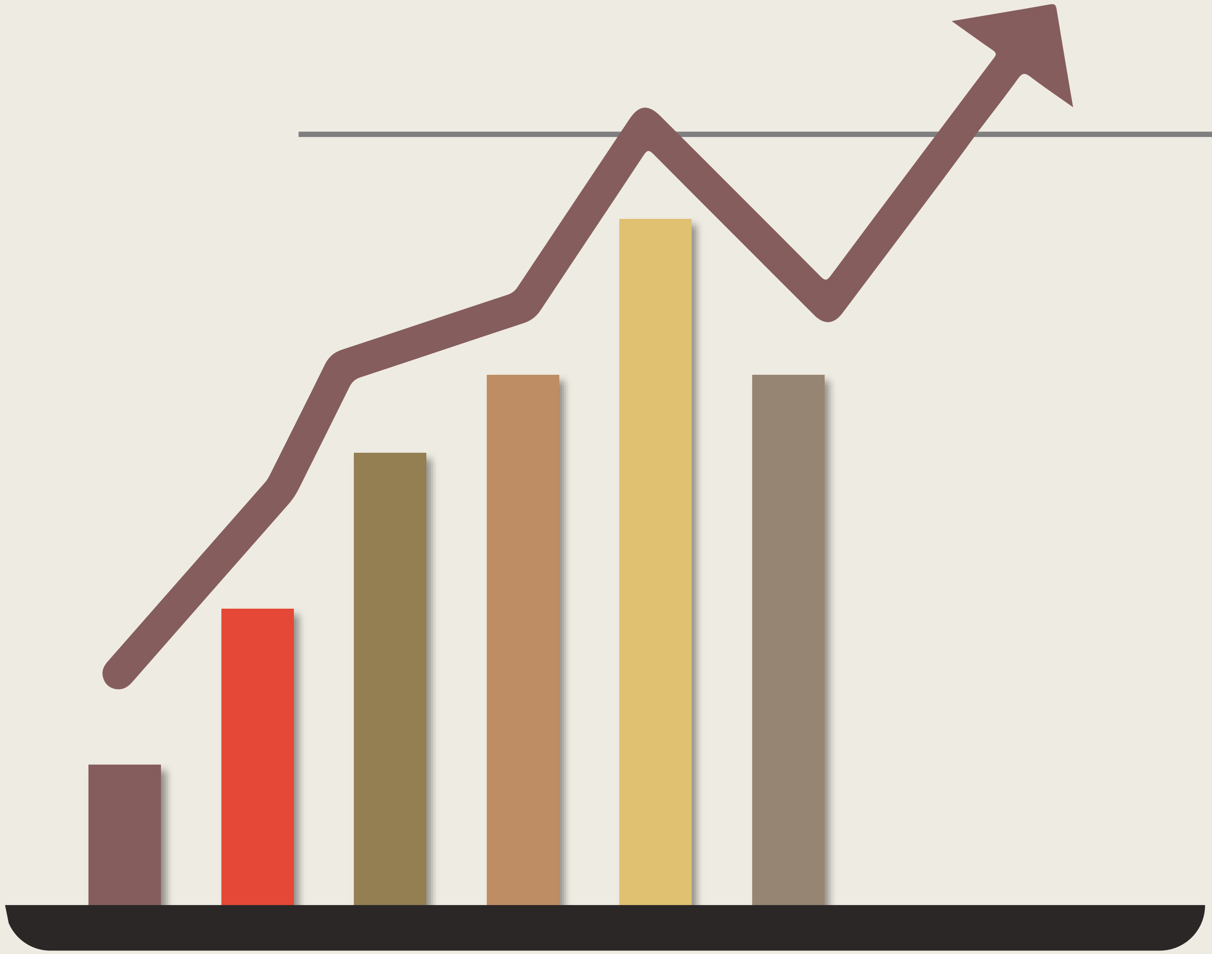
- We can also see the outliers on DATA_EMPLOYED.
- We observe that there are also outliers on the family members and children per family

Bar plot



Let's check each case now based on the target classes

EXPLORATORY DATA ANALYSIS



Categorical features

NAME_EDUCATION_TYPE
NAME_FAMILY_STATUS
NAME_HOUSING_TYPE
FLAG_WORK_PHONE
FLAG_PHONE
FLAG_EMAIL
OCCUPATION_TYPE
CODE_GENDER
FLAG_OWN_CAR
FLAG_OWN_REALTY
CNT_CHILDREN
NAME_INCOME_TYPE
CNT_FAM_MEMBERS

Bar plot

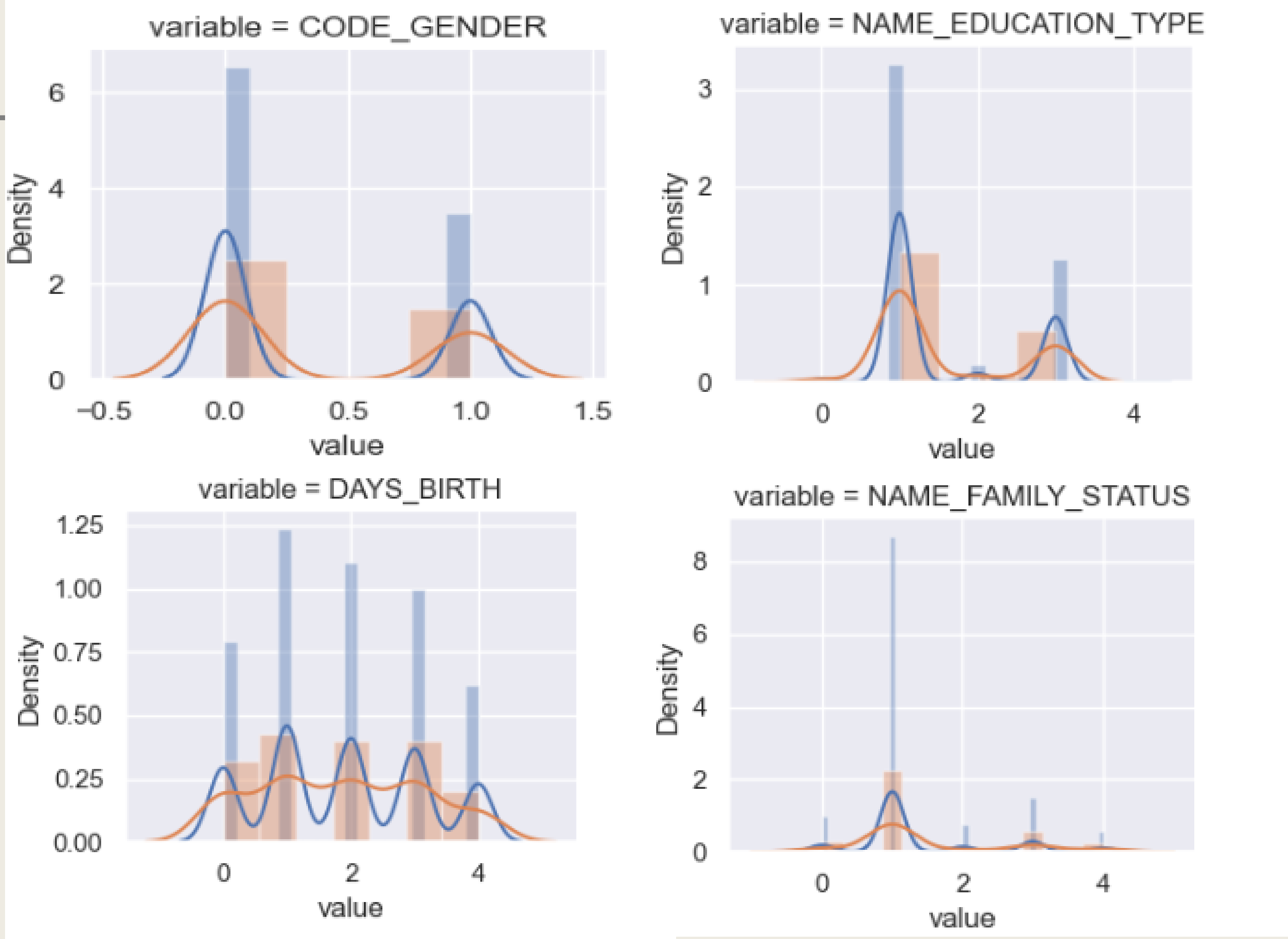
Let's check each case now based on the target classes without scaling so we can see how imbalanced our data are and make some observations

1 **0s have a higher proportion of More Educated**

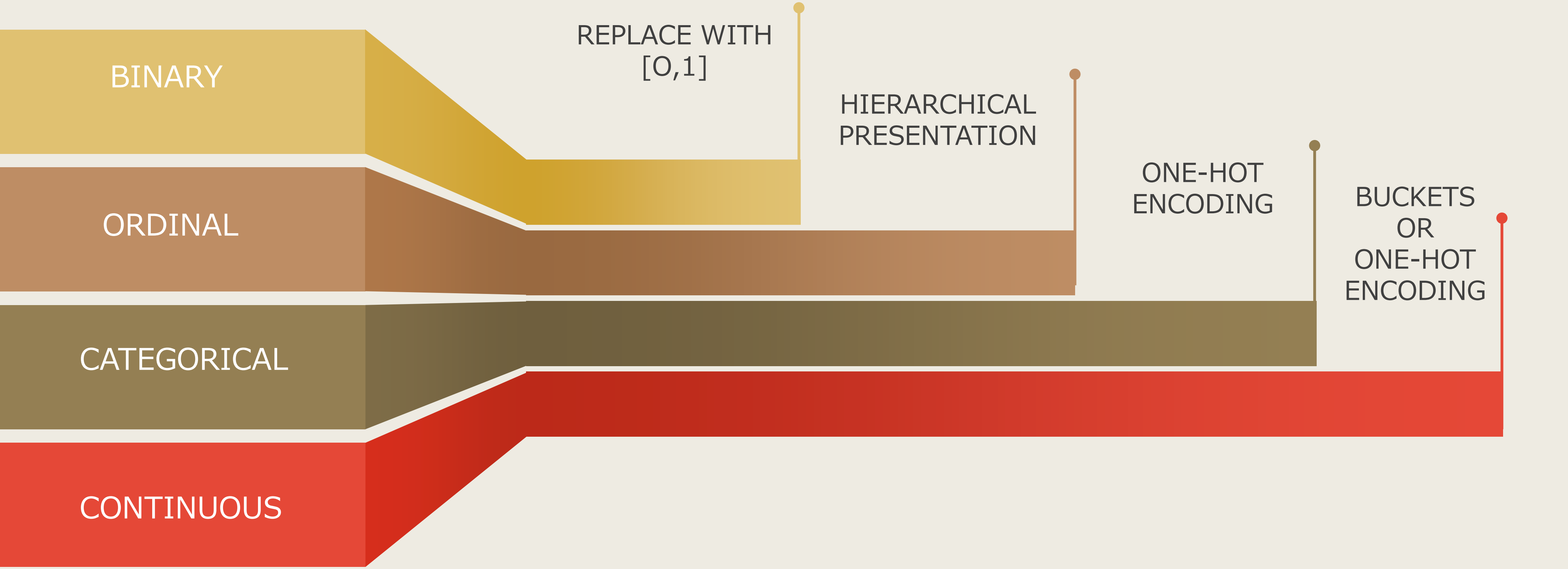
2 **0s have a higher proportion of Females**

3 **0s have a higher proportion of Singles**

4 **0s have a higher proportion of people 30-40years**



DATA CLEANING



DATA CLEANING

BINARY

At first, we will encode the binary features **CODE_GENDER, FLAG_OWN_CAR** and **FLAG_OWN_REALTY.**

In the attribute CODE_GENDER we will replace female 'F' to value 0 and male 'M' to value 1.

In the attribute FLAG_OWN_CAR we will replace yes 'Y' and no 'N' to 1 and 0 respectively.

In the attribute FLAG_OWN_REALTY we will replace as above yes 'Y' and no 'N' to 1 and 0 respectively.

	0	1	3	4	6
CODE_GENDER	M	F	M	F	F
FLAG_OWN_CAR	Y	N	Y	Y	N
FLAG_OWN_REALTY	Y	Y	Y	N	Y

Before

DATA CLEANING

BINARY

At first, we will encode the binary features **CODE_GENDER, FLAG_OWN_CAR** and **FLAG_OWN_REALTY.**

In the attribute CODE_GENDER we will replace female 'F' to value 0 and male 'M' to value 1.

In the attribute FLAG_OWN_CAR we will replace yes 'Y' and no 'N' to 1 and 0 respectively.

In the attribute FLAG_OWN_REALTY we will replace as above yes 'Y' and no 'N' to 1 and 0 respectively.

	0	1	3	4	6
CODE_GENDER	M	F	M	F	F
FLAG_OWN_CAR	Y	N	Y	Y	N
FLAG_OWN_REALTY	Y	Y	Y	N	Y

Before

After

	0	1	2	3	4
CODE_GENDER	1	0	1	0	0
FLAG_OWN_CAR	1	0	1	1	0
FLAG_OWN_REALTY	1	1	1	0	1

DATA CLEANING

ORDINAL

In the attribute **NAME_EDUCATION_TYPE** there are five unique values which are :

Because this column has a hierarchy, we are going to implement ordinal encoding in order to preserve the ordinal nature of our feature.

Label encoding should not be used with linear models where magnitude of features plays an important role

	NAME_EDUCATION_TYPE
0	Secondary / secondary special
1	Secondary / secondary special
2	Secondary / secondary special
3	Higher education
4	Incomplete higher

Before

DATA CLEANING

ORDINAL

In the attribute **NAME_EDUCATION_TYPE** there are five unique values which are :

Because this column has a hierarchy, we are going to implement ordinal encoding in order to preserve the ordinal nature of our feature.

Label encoding should not be used with linear models where magnitude of features plays an important role

	NAME_EDUCATION_TYPE
0	Secondary / secondary special
1	Secondary / secondary special
2	Secondary / secondary special
3	Higher education
4	Incomplete higher

Before

After

	NAME_EDUCATION_TYPE
0	1
1	1
2	1
3	3
4	2

DATA CLEANING

In this case, because our categorical variables **NAME_FAMILY_STATUS**, **NAME_HOUSING_TYPE** and **OCCUPATION_TYPE** have equal order, we are going to implement One-hot encoding.

In One-hot encoding our number of features will increase, which is not good for any tree based algorithm like Decision-trees, Random Forest etc

CATEGORICAL

	NAME_FAMILY_STATUS	NAME_HOUSING_TYPE	OCCUPATION_TYPE
0	Married	House / apartment	Security staff
1	Single / not married	House / apartment	Sales staff
2	Single / not married	House / apartment	Sales staff
3	Married	House / apartment	Accountants
4	Married	House / apartment	Accountants

Before

DATA CLEANING

CATEGORICAL

In this case, because our categorical variables **NAME_FAMILY_STATUS**, **NAME_HOUSING_TYPE** and **OCCUPATION_TYPE** have equal order, we are going to implement One-hot encoding.

In One-hot encoding our number of features will increase, which is not good for any tree based algorithm like Decision-trees, Random Forest etc

	NAME_FAMILY_STATUS	NAME_HOUSING_TYPE	OCCUPATION_TYPE
0	Married	House / apartment	Security staff
1	Single / not married	House / apartment	Sales staff
2	Single / not married	House / apartment	Sales staff
3	Married	House / apartment	Accountants
4	Married	House / apartment	Accountants

Before

After

	0	1	2	3	4
NAME_FAMILY_STATUS0	0.0	0.0	0.0	0.0	0.0
NAME_FAMILY_STATUS1	1.0	0.0	0.0	1.0	1.0
NAME_FAMILY_STATUS2	0.0	0.0	0.0	0.0	0.0
NAME_FAMILY_STATUS3	0.0	1.0	1.0	0.0	0.0
NAME_FAMILY_STATUS4	0.0	0.0	0.0	0.0	0.0
NAME_HOUSING_TYPE0	0.0	0.0	0.0	0.0	0.0
NAME_HOUSING_TYPE1	1.0	1.0	1.0	1.0	1.0
NAME_HOUSING_TYPE2	0.0	0.0	0.0	0.0	0.0
NAME_HOUSING_TYPE3	0.0	0.0	0.0	0.0	0.0
NAME_HOUSING_TYPE4	0.0	0.0	0.0	0.0	0.0
NAME_HOUSING_TYPE5	0.0	0.0	0.0	0.0	0.0
OCCUPATION_TYPE0	0.0	0.0	0.0	0.0	0.0
OCCUPATION_TYPE1	1.0	0.0	0.0	0.0	0.0
OCCUPATION_TYPE2	0.0	1.0	1.0	1.0	1.0
OCCUPATION_TYPE3	0.0	0.0	0.0	0.0	0.0

DATA CLEANING

CONTINUOUS

For the case of **CNT_FAM_MEMBERS** and **CNT_CHILDREN**, we observe some outliers.

In order to deal with those edge cases and have a consistency across our dataset, we choose to group these columns into buckets

Before

	CNT_FAM_MEMBERS	CNT_CHILDREN
0	2.0	0
1	1.0	0
2	1.0	0
3	2.0	0
4	2.0	0

DATA CLEANING

For the case of **CNT_FAM_MEMBERS** and **CNT_CHILDREN**, we observe some outliers.

In order to deal with those edge cases and have a consistency across our dataset, we choose to group these columns into buckets

CONTINUOUS

After

	CNT_FAM_MEMBERS	CNT_CHILDREN
0	1	0
1	0	0
2	1	0
3	1	0
4	1	0

DATA CLEANING

OTHER ways of data cleaning

INITIAL DATA

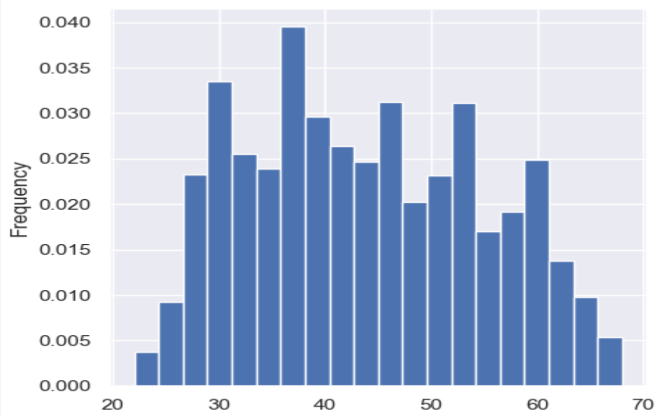
	DAYS_BIRTH
0	-21474
1	-19110
2	-19110
3	-16872
4	-10669

YEARS

	DAYS_BIRTH
0	58
1	52
2	52
3	46
4	29

AFTER AGE-CLUSTRERING

	DAYS_BIRTH
0	3
1	3
2	2
3	0
4	0

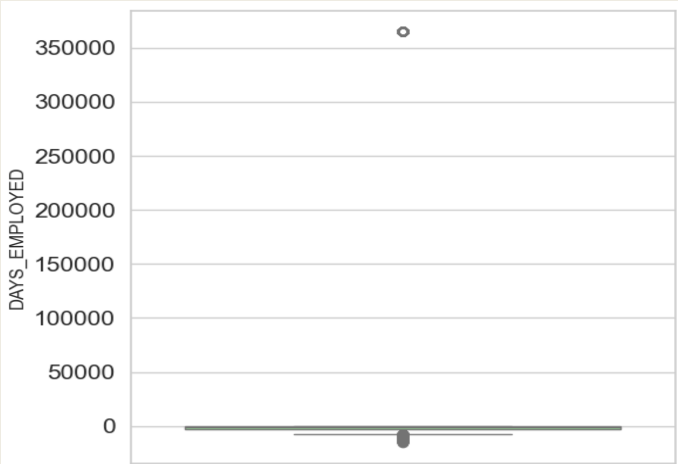


DAYS_BIRTH

DATA CLEANING

OTHER ways of data cleaning

DAYS_EMPLOYED



DAYS_BIRTH

BEFORE

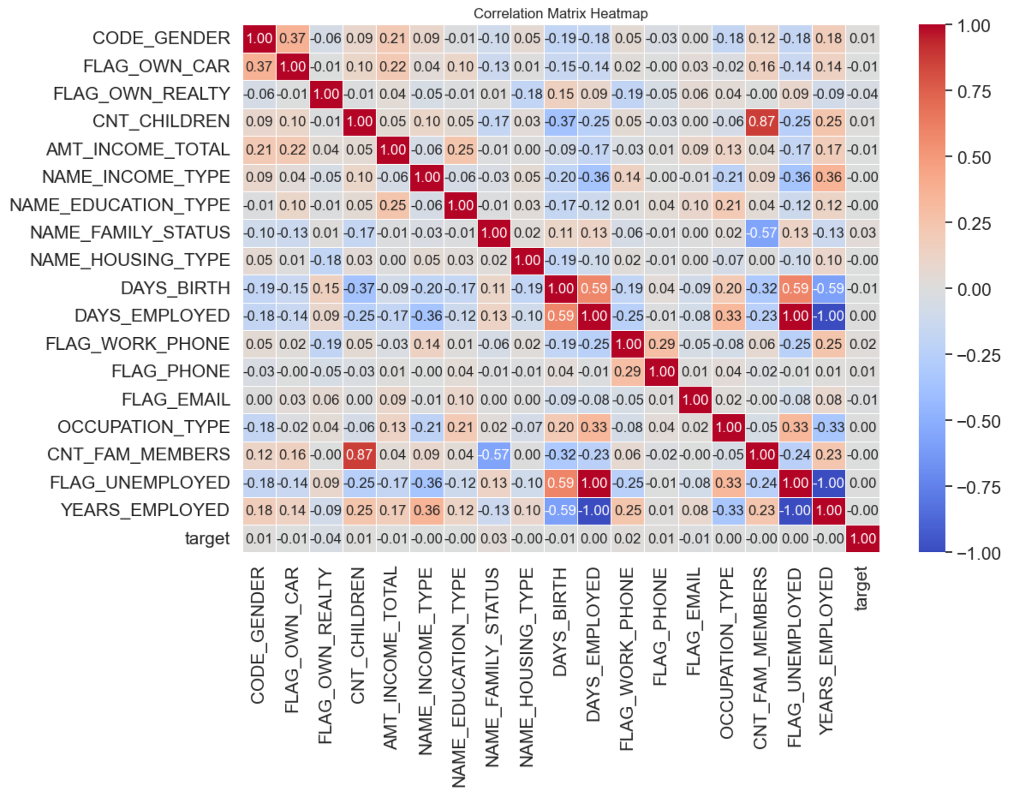
	DAYS_EMPLOYED
0	-1134
1	-3051
2	-769

AFTER

	FLAG_UNEMPLOYED	YEARS_EMPLOYED
0	0	3.106849
1	0	8.358904
2	0	2.106849

DATA CLEANING

OTHER ways of data cleaning



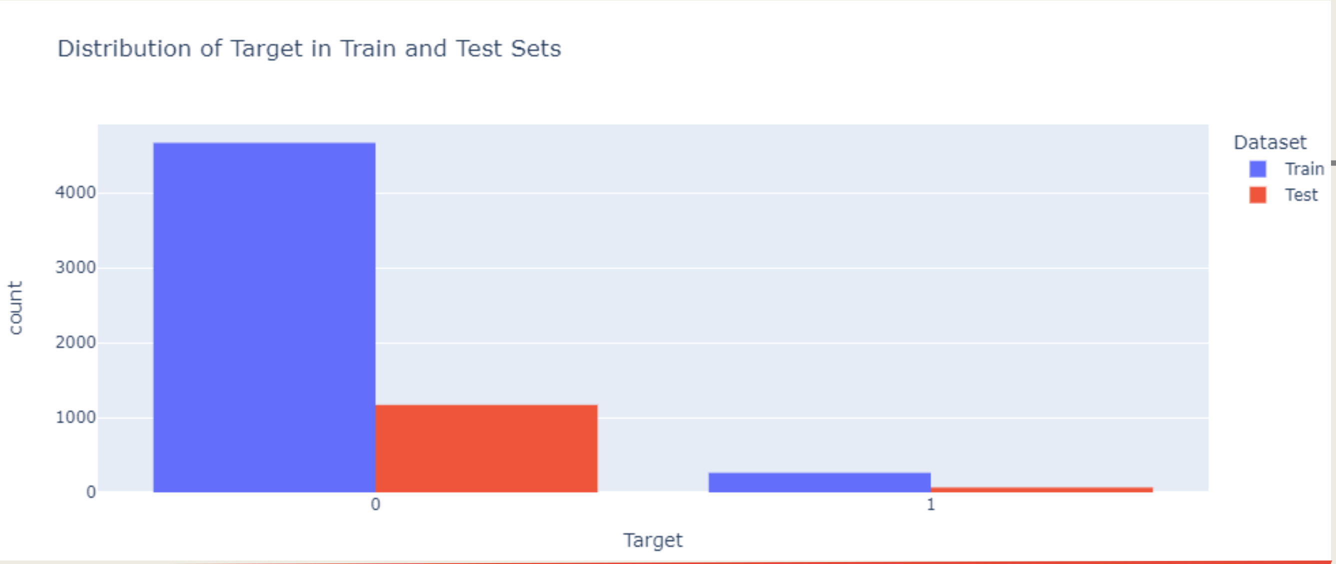
CNT_FAM_MEMBERS

DAYS_EMPLOYED

DAYS_BIRTH

An important step for our project

Split to Train and Test




```
G_OWN_CAR','FLAG_OWN_REALTY','CNT_CHILDRE
', 'NAME_INCOME_TYPE2', 'NAME_INCOME_TYPE3
4', 'NAME_FAMILY_STATUS0', 'NAME_FAMILY_STA
TUS2', 'NAME_FAMILY_STATUS3', 'NAME_FAMILY_S
YPE0', 'NAME_HOUSING_TYPE1', 'NAME_HOUSING_T
TYPE3', 'NAME_HOUSING_TYPE4', 'NAME_HOUSING_
YPE0', 'OCCUPATION_TYPE1', 'OCCUPATION_TYPE2
TYPE3']]
stype(int)

st, y_train, y_test = train_test_split(X, Y,
aFrames for y_train and y_test
pd.DataFrame({'target': y_train, 'dataset':
pd.DataFrame({'target': y_test, 'dataset': '
ed = pd.concat([df_train, df_test])

istributions using Plotly Express
x.histogram(df_combined, x='target', color='
            labels={'target': 'Target', 'da
            title='Distribution of Target
update_layout(width = 1000)
.show()
```

From now on, we are going to continue with the training data only

Normalization of features

- Normalizing our features before oversampling depends on the specific oversampling technique we are using and the nature of our data.

FEATURE SELECTION

01

Select From Model

- For **Linear** cases
- Ex. Logistic Regression
- Feature importance through a coef_ attribute

FEATURE SELECTION

01

Select From Model

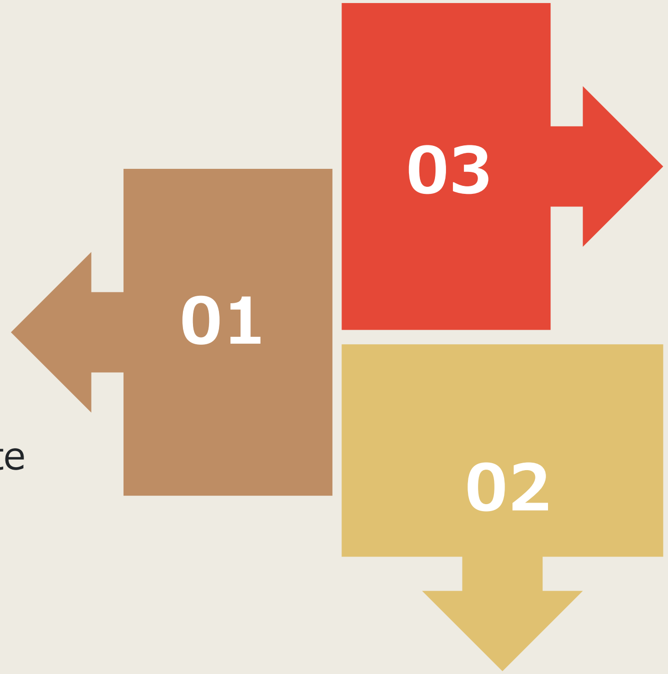
- For **Linear** cases
- Ex. Logistic Regression
- Feature importance through a coef_ attribute

02

Select From Model

- For **Tree-based** cases
- Ex. Random Forest
- feature_importances attribute

FEATURE SELECTION



Select From Model

- For **Linear** cases
- Ex. Logistic Regression
- Feature importance through a coef_ attribute

Select From Model

- For **Tree-based** cases
- Ex. Random Forest
- feature_importances attribute

RFECV using Stratified k-fold and f1 scoring

FEATURE SELECTION

Select From Model	Select From Model	RFECV
Extra Trees Classifier	Random Forest	Using Stratified k-fold and f1 scoring
CODE_GENDER FLAG_PHONE	CODE_GENDER CNT_FAM_MEMBERS	CODE_GENDER
FLAG_OWN_CAR CNT_FAM_MEMBERS	FLAG_OWN_CAR YEARS_EMPLOYED	AMT_INCOME_TOTAL
CNT_CHILDREN YEARS_EMPLOYED	AMT_INCOME_TOTAL	NAME_EDUCATION_TYPE
AMT_INCOME_TOTAL	NAME_EDUCATION_TYPE	DAYS_BIRTH
NAME_EDUCATION_TYPE	DAYS_BIRTH	CNT_FAM_MEMBERS
DAYS_BIRTH	FLAG_PHONE	YEARS_EMPLOYED

OVERSAMPLING

When working on a dataset with class imbalance problem, one needs to oversample or under sample only the train set and not the test set

SMOTE

- A subset of minority class is taken and new synthetic data points are generated based on it.
- SMOTE, is a clever way to perform over-sampling over the minority class to avoid overfitting(unlike random over-sampling that has overfitting problems)

```
Training Set Class Balance before:target
0     4669
1      261
```

SMOTE-NC

- Creates synthetic data for categorical as well as quantitative features in the data set.

```
Training Set Class Balance before:target
0     4669
1      261
```

OVERSAMPLING

When working on a dataset with class imbalance problem, one needs to oversample or under sample only the train set and not the test set

SMOTE

- A subset of minority class is taken and new synthetic data points are generated based on it.
- SMOTE, is a clever way to perform over-sampling over the minority class to avoid overfitting(unlike random over-sampling that has overfitting problems)

```
Training Set Class Balance before:target
0    4669
1     261
```

```
Training Set Class Balance now:target
0    4669
1    4669
```

SMOTE-NC

- Creates synthetic data for categorical as well as quantitative features in the data set.

```
Training Set Class Balance before:target
0    4669
1     261
```

```
Training Set Class Balance now:target
0    4669
1    4669
```

BUILD A MACHINE

LEARNING MODEL FOR PREDICTION

MODEL SELECTION

WHY?

- Best metric for imbalanced data
- F1 near 1 means that we have a good model

F1

METRICS SLECTION

01

ALGORITHMS FOR IMBALANCED DATA

- Logistic Regression
- Random Forest Classifier
- SVC
- Naïve Bayes
- XGB Classifier
- LGBM Classifier

02

ALGORITHMS

STRATIFIED-KFOLD

Ensures that each fold maintains the same proportion of target classes as the original dataset

CROS-VALIDATION

03

04

HYPERPARAMETER TUNNING

Tuning methods

- GridSearchCV
- RandomizedSearchCV

EVALUATION METRICS

1

F1-score

is a metric that combines precision and recall into a single value. It is particularly useful in binary classification settings where there is an imbalance between the classes

2

Precision Recall Curve

Provide insights into the model's performance

3

Confusion Matrix

Represents classifier's performance

4

Classification Report

With all the above now we can have a classification report

F1-score

Training set

Algorithm	Baseline	Normalization	Oversampling	Feature Selection
Logistic Regression	0.000000	0.007547	0.658003	0.664809
Random Forest	0.022884	0.011775	0.960154	0.959742
SVC	0.000000	0.000000	0.832008	0.857495
Naive Bayes	0.000000	0.095734	0.506672	0.694134
XGBoost	0.034447	0.034447	0.961965	0.960625
LightGBM	0.024964	0.025871	0.966350	0.965120

F1-score

Training set

Algorithm	Baseline	Normalization	Oversampling	Feature Selection
Logistic Regression	0.000000	0.007547	0.658003	0.664809
Random Forest	0.022884	0.011775	0.960154	0.959742
SVC	0.000000	0.000000	0.832008	0.857495
Naive Bayes	0.000000	0.095734	0.506672	0.694134
XGBoost	0.034447	0.034447	0.961965	0.960625
LightGBM	0.024964	0.025871	0.966350	0.965120

Our Path Forward

Perform Predictions

Finalize model

For three winning algorithms

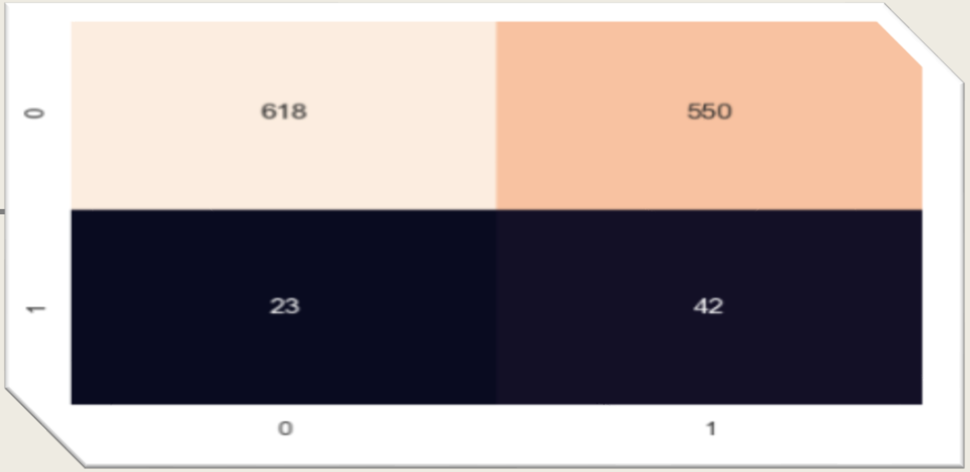
Random Forest
XGBoost
LightGBM

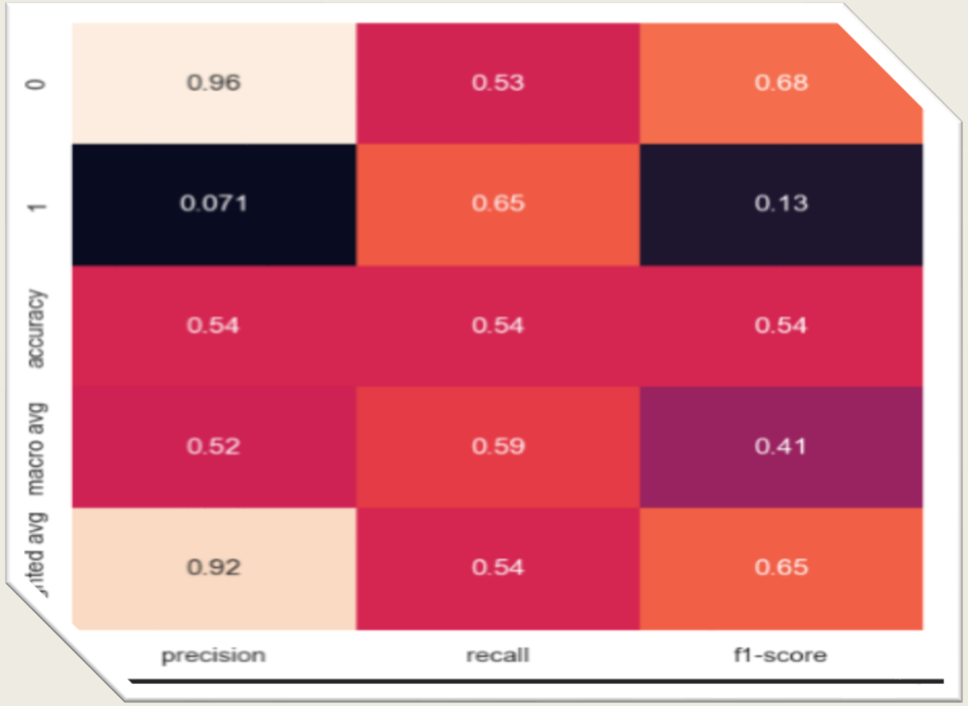
Logistic Regression

Select the final model

LGBM CLASSIFIER

LOGISTIC REGRESSION





Conclusions

Training Data

The Impact of Oversampling in Training

Oversampling data performs better

Best Algorithms

- XGBoost
- LightGBM
- Random Forest
- !!!Logistic Regression

Test Data

Test Data without oversampling

To evaluate True Model Performance

Differces in f1 score

Best f1-score Logistic Regression but the performance is worst than others

Imbalanced data

Numerous Attempts (normalization, oversampling,feature selection,stratifiedkfold cross validation ,hyperparameter tuning) ,but the Data Imbalance Remained a Challenge

ALTERNATIVE APPROACHES

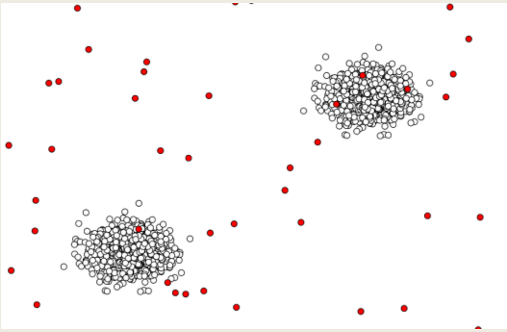
We can use the **unsupervised** data and make clusters

☑ Outlier Detection

-Isolation Forest

Robust to high-dimensional data

and can efficiently isolate outliers.



-One-Class SVM

Handles high-dimensional data and
can effectively identify outliers by learning
the normal patterns in the feature space.

Thank you
for your time !!!