Banking Customer Churn Prediction

Artificial Intelligence – Supervised Learning

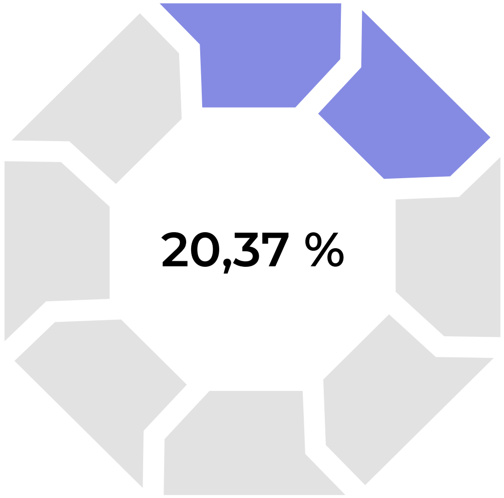
Banking Customer Churn Prediction

Dataset

Bank customers and their churn status, which indicates whether they have exited the bank or not.

Machine learning problem

Factors influencing customer churn in banking institutions and build predictive models to identify customers at risk of churning.



Churn rate : 2037 / 10000

Tools and algorithms to use

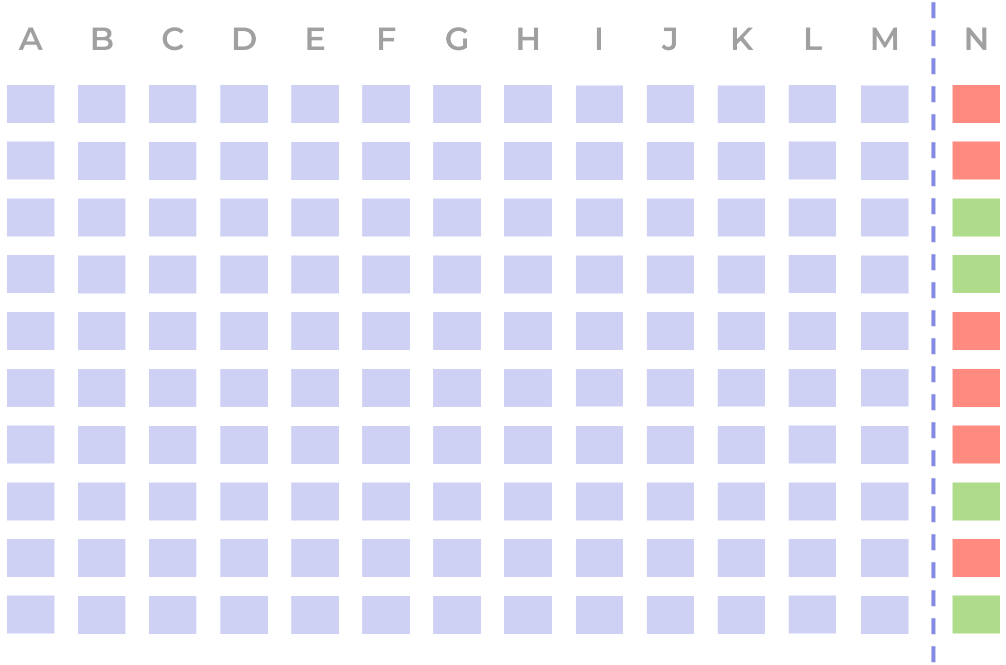
Tools

- Pandas
- Numpy
- MatPlot
- Seaborn
- Scikit-learn
- TensorFlow
- Imbalanced-learn

Algorithms

- Decision Trees
- Neural Networks
- K-Nearest Neighbors (K-NN)
- Support Vector Machines (SVM)
- Naïve Bayes

Dataset features



A : row number

B : customer id

C : surname

D : credit score

E : geography

F : gender

G : age

H : tenure

I : balance

J : numOfProducts

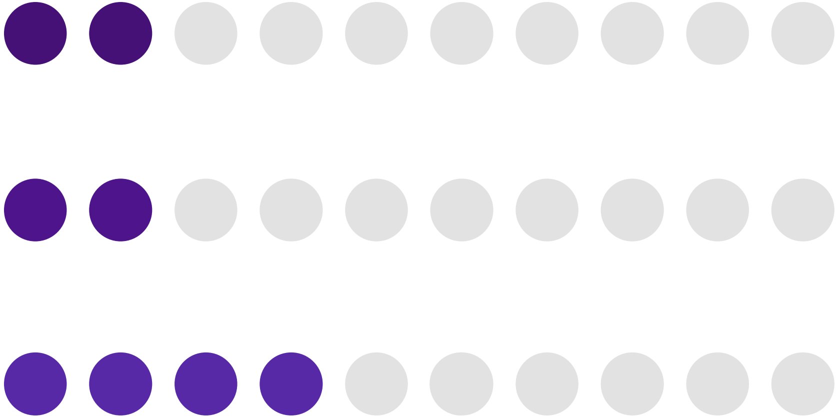
K : hasCreditCard

L : isActiveMember

M : estimatedSalary

N : exited

Exited customers per country



16% **Spain**

Churn rate : 413 / 2477

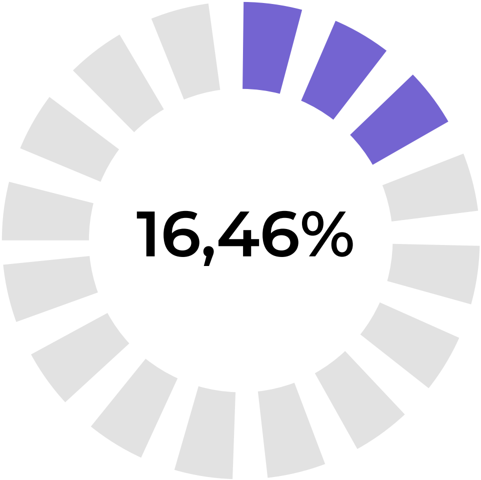
16% **France**

Churn rate : 810 / 5014

32% **Germany**

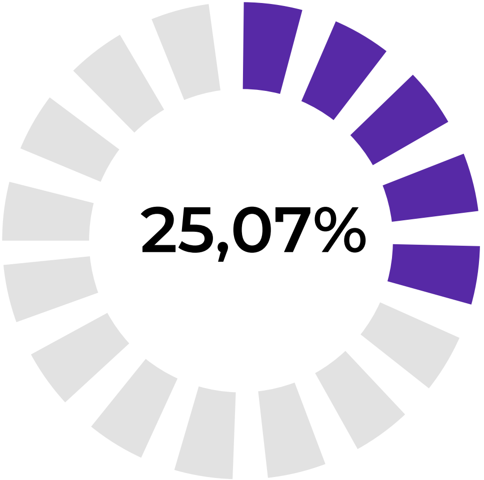
Churn rate : 814 / 2509

Exited customers per gender



Male

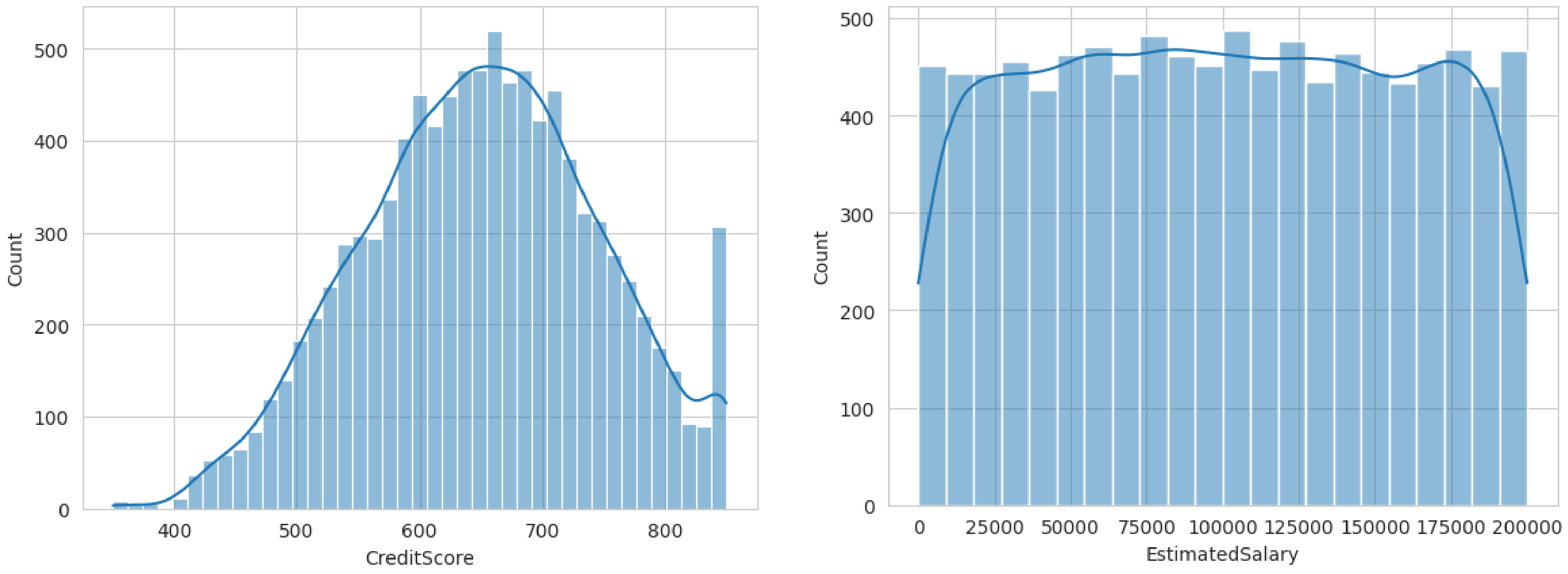
Churn rate : 898 / 5457

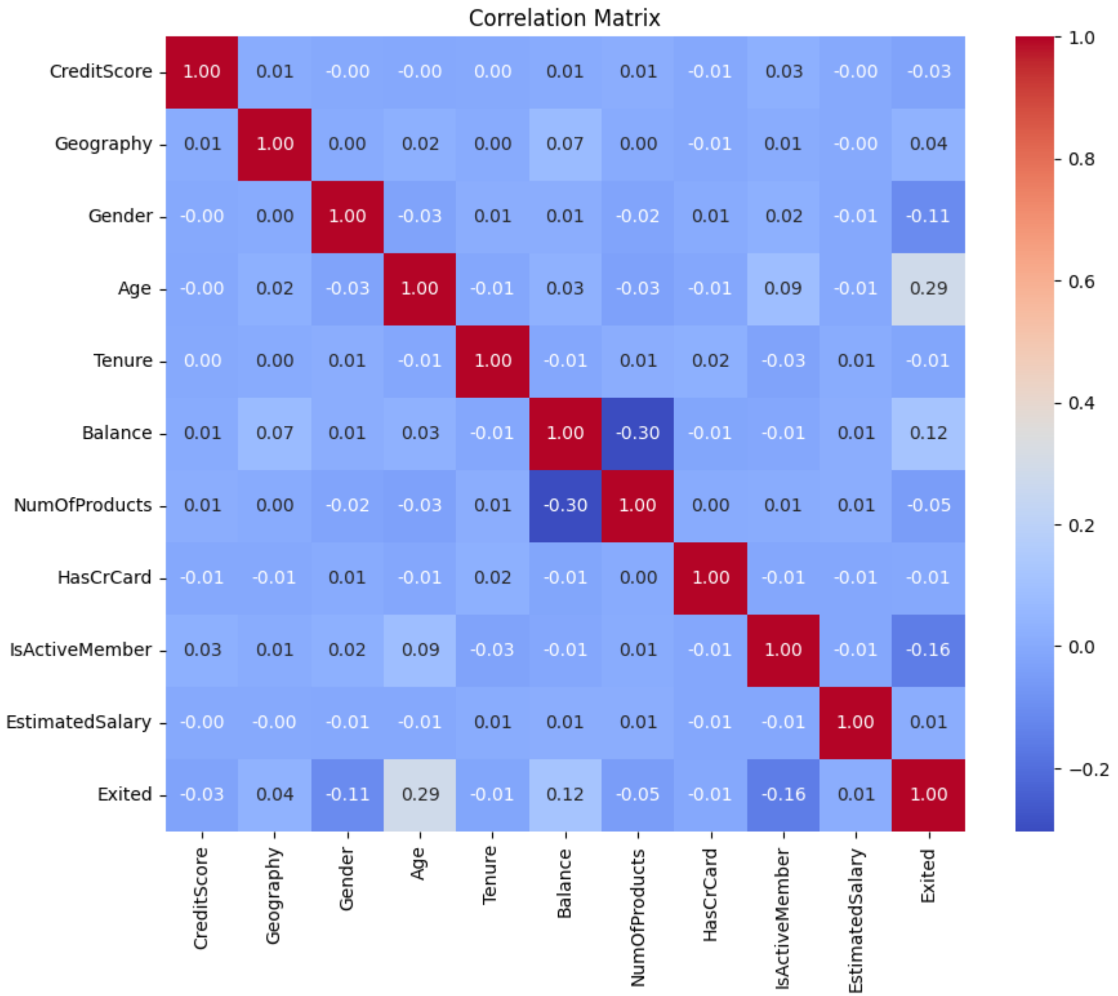


Female

Churn rate : 1139 / 4543

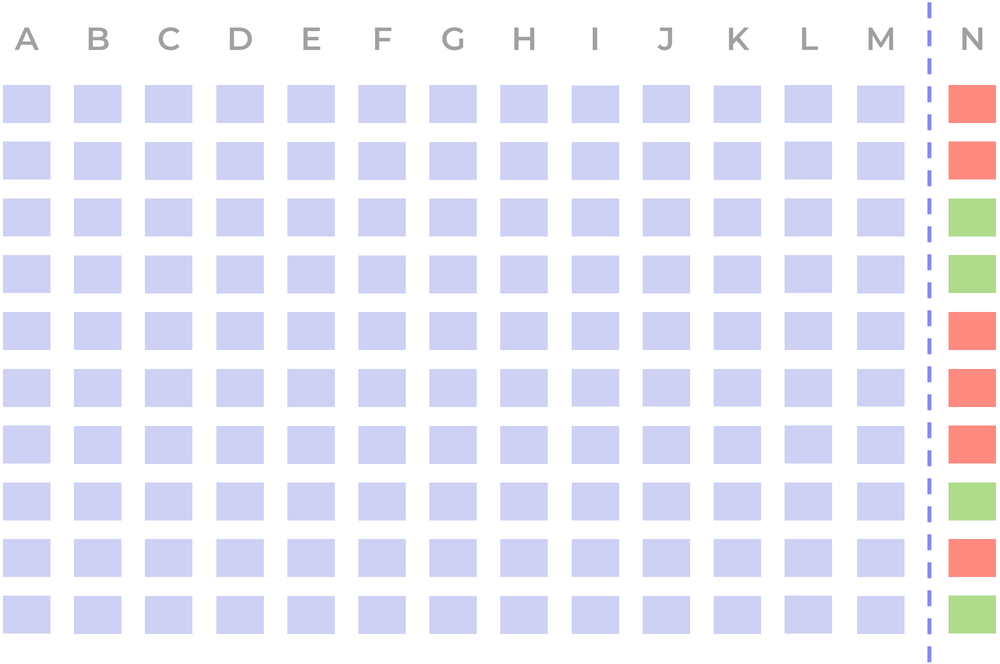
Credit score & estimated salary




Correlation Matrix
CreditScore
Geography
Gender
Age
Tenure
Balance
NumOfProducts
HasCrCard
IsActiveMember
EstimatedSalary
Exited
1.00 0.01 -0.00 -0.00 0.00 0.01 0.01 -0.01 0.03 -0.00 -0.03
0.01 1.00 0.00 0.02 0.00 0.07 0.00 -0.01 0.01 -0.00 0.04
-0.00 0.00 1.00 -0.03 0.01 0.01 -0.02 0.01 0.02 -0.01 -0.11
-0.00 0.02 -0.03 1.00 -0.01 0.03 -0.03 -0.01 0.09 -0.01 0.29
0.00 0.00 0.01 -0.01 1.00 -0.01 0.01 0.02 -0.03 0.01 -0.01
0.01 0.07 0.01 0.03 -0.01 1.00 -0.30 -0.01 -0.01 0.01 0.12
0.01 0.00 -0.02 -0.03 0.01 -0.30 1.00 0.00 0.01 0.01 -0.05
-0.01 -0.01 0.01 -0.01 0.02 -0.01 0.00 1.00 -0.01 -0.01 -0.01
0.03 0.01 0.02 0.09 -0.03 -0.01 0.01 -0.01 1.00 -0.01 -0.16
-0.00 -0.00 -0.01 -0.01 0.01 0.01 0.01 -0.01 -0.01 1.00 0.01
-0.03 0.04 -0.11 0.29 -0.01 0.12 -0.05 -0.01 -0.16 0.01 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Data preprocessing

Dropping features



A : row number

B : customer id

C : surname

D : credit score

E : geography

F : gender

G : age

H : tenure

I : balance

J : numOfProducts

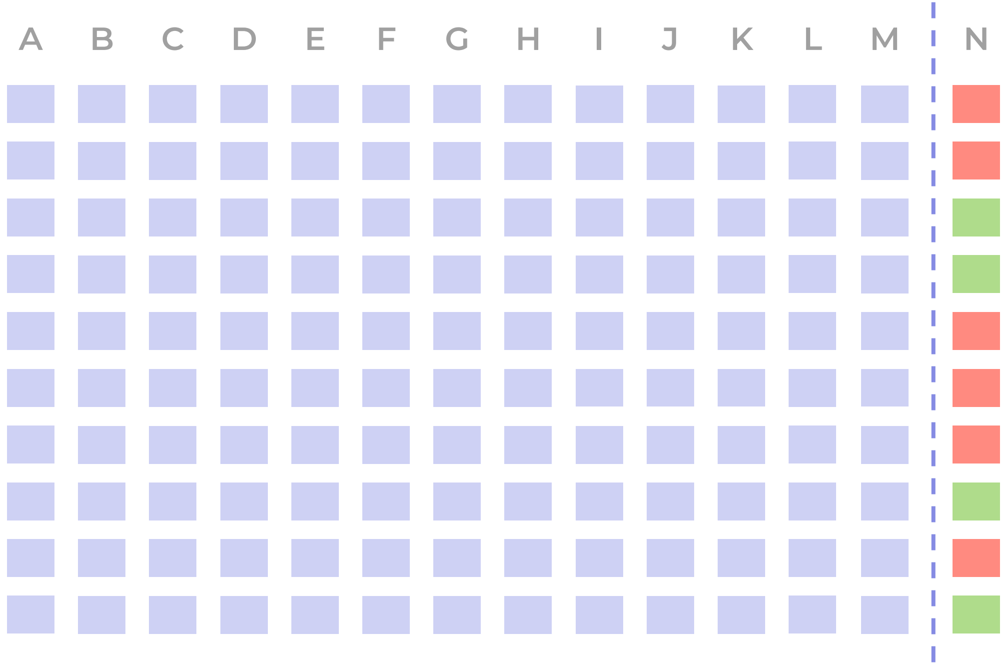
K : hasCreditCard

L : isActiveMember

M : estimatedSalary

N : exited

Dropping features



A : row number

B : customer id

C : surname

D : credit score

E : geography

F : gender

G : age

H : tenure

I : balance

J : numOfProducts

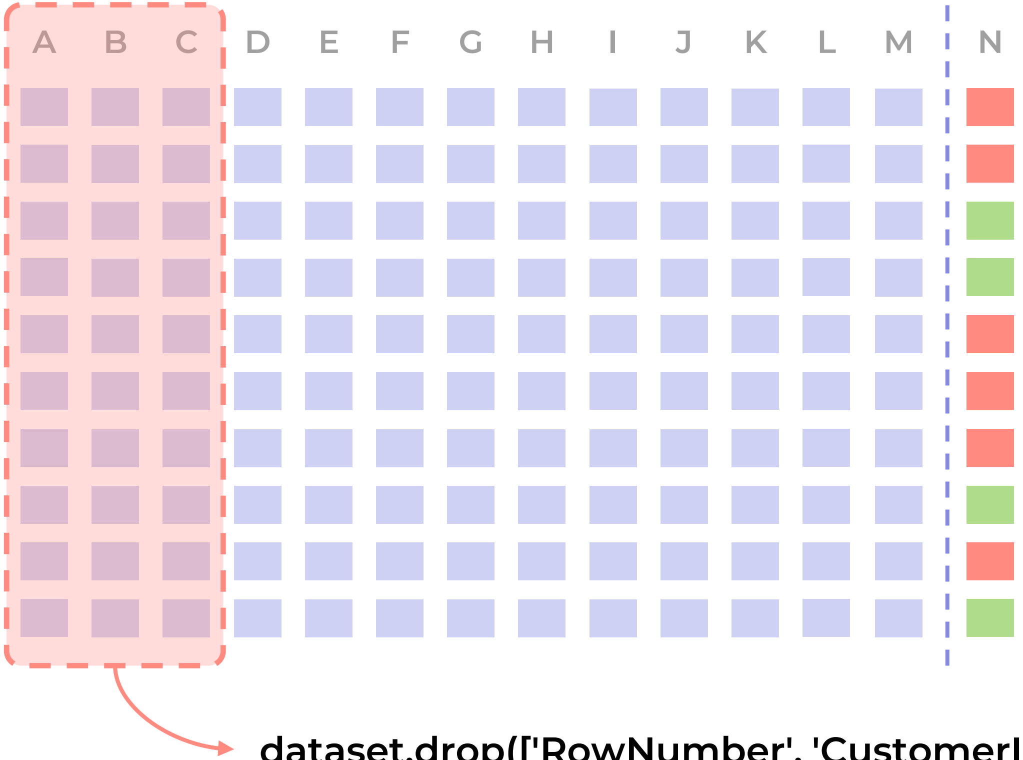
K : hasCreditCard

L : isActiveMember

M : estimatedSalary

N : exited

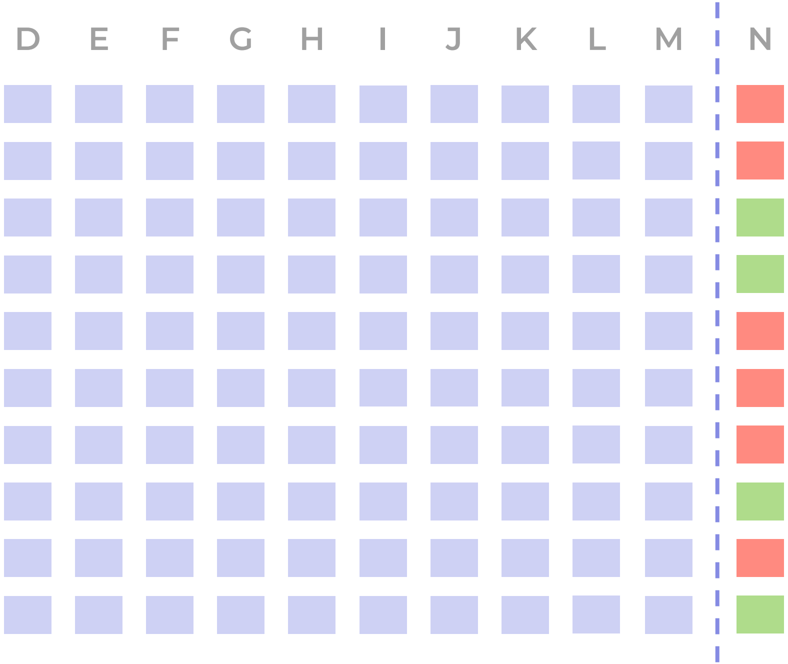
Dropping features



- A : row number
- B : customer id
- C : surname
- D : credit score
- E : geography
- F : gender
- G : age
- H : tenure
- I : balance
- J : numOfProducts
- K : hasCreditCard
- L : isActiveMember
- M : estimatedSalary
- N : exited

```
dataset.drop(['RowNumber', 'CustomerId', 'Surname’])
```

Dropping features



A : row number
B : customer id
C : surname
D : credit score
E : geography
F : gender
G : age
H : tenure
I : balance
J : numOfProducts
K : hasCreditCard
L : isActiveMember
M : estimatedSalary
N : exited

Converting input types

Gender	Gender	Geography	Geography
Male	1	Germany	2
Female	0	France	0
Female	0	Spain	1
Male	1	Germany	2
Male	0	Germany	2
Female	1	France	0

Normalizing salary

```
# Mapping the country to the minimum salary
country_salaries = {'France': 1540 * 12, 'Spain': 1050 * 12, 'Germany': 1580 * 12}

# Remove rows with 'EstimatedSalary' less than 1000
dataset = dataset[dataset['EstimatedSalary'] >= 1000]

# Multiply 'EstimatedSalary' by 12 if it's less than the corresponding country's salary
dataset.loc[dataset['EstimatedSalary'] < dataset['Geography'].map(country_salaries),
'EstimatedSalary'] *= 12
```

Handle imbalanced data

20%

Initial dataset

Imbalanced - 2037 / 7963

smt = SMOTE(random_state=5)

50%

Final dataset

Balanced - 7963 / 7963