Bank Churn Customer

10/03/2024

Cao Thanh Phat

01 - Introduction

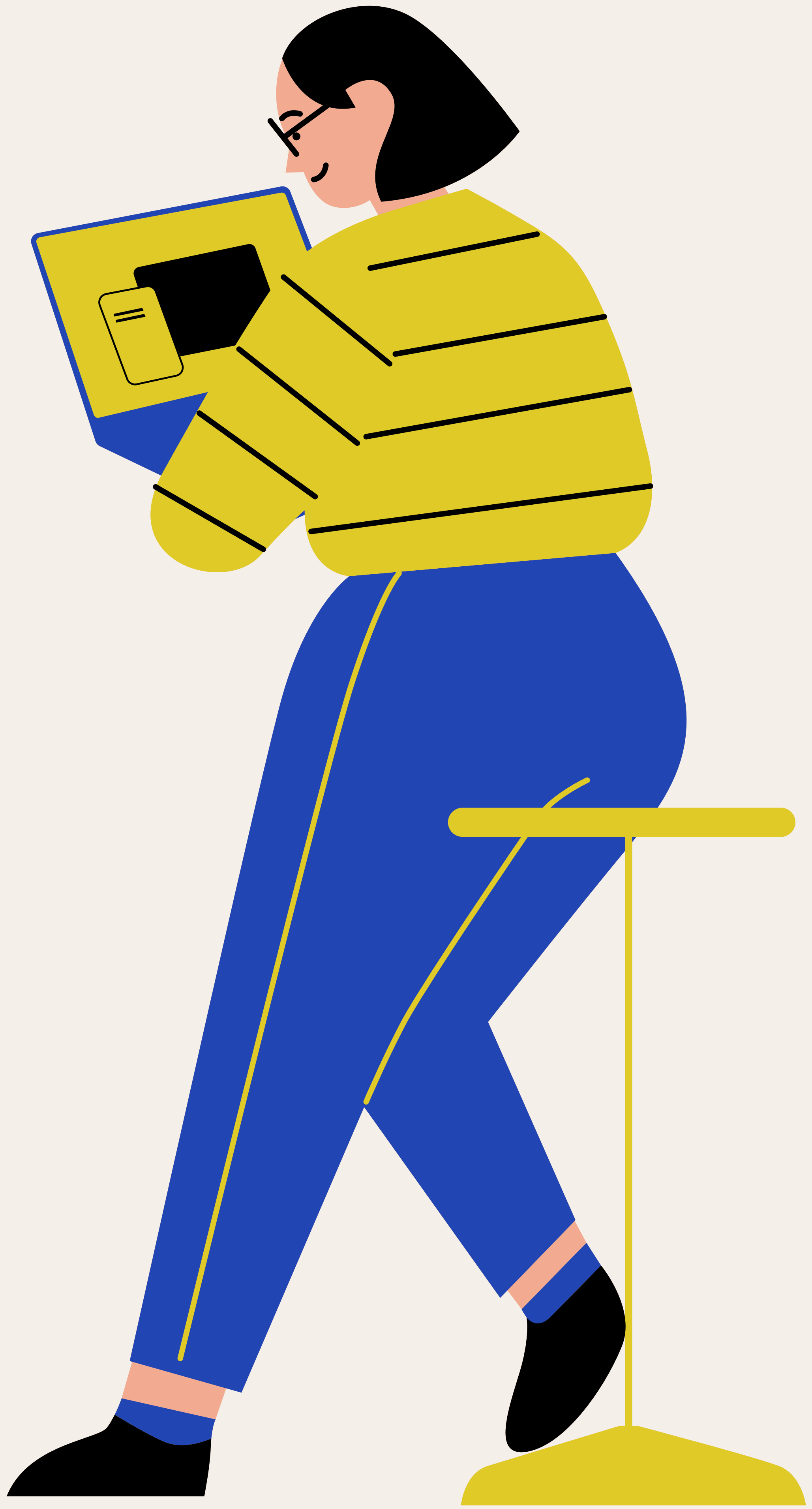
02 - Explore Data Analysis

03 - Build Model ML

04 - Conclusions

Data

Visualization



Link

phatcao285/**Project-Data-Analyst**

1 Contributor · 0 Issues · 0 Stars · 0 Forks

Project-Data-Analyst/Churn_customer_Classifier.ipynb at main · phatcao285/Project-Data-Analyst

Contribute to phatcao285/Project-Data-Analyst development by creating an account on GitHub.

GitHub

Cao Thanh Phat

01 - Introduction

Problem Statement

Data

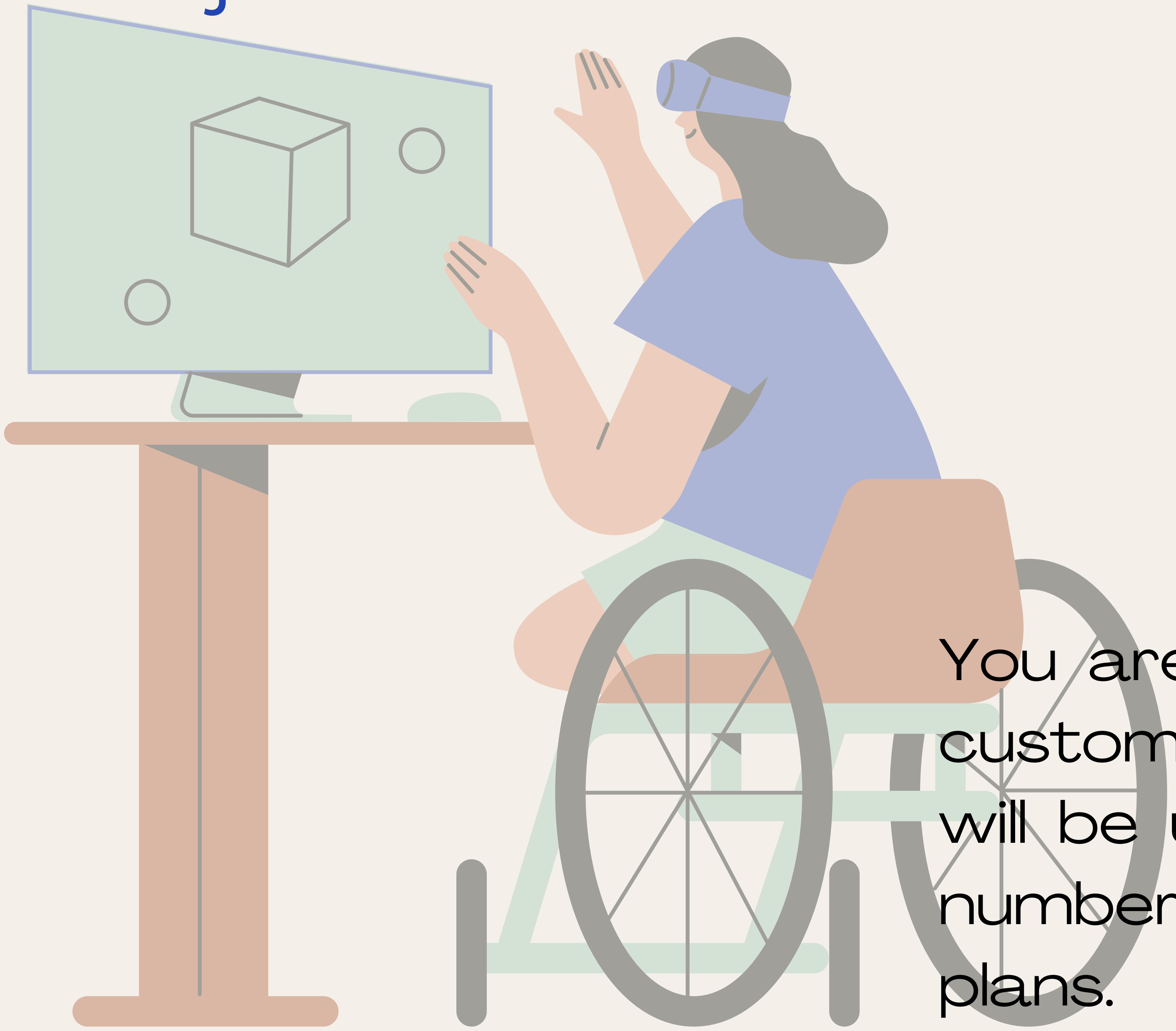
Visualization

Banks all want to retain their customers to maintain business operations and ABC Multinational Bank also wants that. Below is the customer data of customers at ABC Multinational Bank that has been discovered. transaction generation and the purpose of the data will be to predict Customer Churn Rate.

Suppose you are a Data Analyst for ABC bank. BOD is trying to find out why the above problem occurs and whether users of the services will leave ABC (cancel the service) in the next few days.

01 - Introduction

Objective



Data

Visualization

Machine Learning

You are asked to build a model to predict whether customers will leave or continue to use the service. It will be used by the Strategy team to estimate the number of customer churn and develop improvement plans.

02 - Exploring Data Analysis

This dataset have 10000 columns and 12 rows

```
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   customer_id      10000 non-null  int64
 1   credit_score     10000 non-null  int64
 2   country          10000 non-null  object
 3   gender           10000 non-null  object
 4   age              10000 non-null  int64
 5   tenure           10000 non-null  int64
 6   balance          10000 non-null  float64
 7   products_number  10000 non-null  int64
 8   credit_card      10000 non-null  int64
 9   active_member    10000 non-null  int64
 10  estimated_salary 10000 non-null  float64
 11  churn            10000 non-null  int64
```

02 - Exploring Data Analysis

```
df.duplicated().sum() # kiểm tra dữ liệu bị trùng lặp
```

```
0
```

```
import numpy as np
df.isin([np.inf, -np.inf]).any() # kiểm tra infinity của dữ liệu
```

```
customer_id         False
credit_score        False
country             False
gender              False
age                 False
tenure              False
balance             False
products_number     False
credit_card         False
active_member       False
estimated_salary    False
churn               False
```

1. This dataset consists of 10000 rows and 12 columns
2. Haven't seen any empty data column "null"
3. There is no data column containing infinity form.
4. Most data types are quite clean, are float or integer. Only left:
   Columns "country" and "gender" are strings

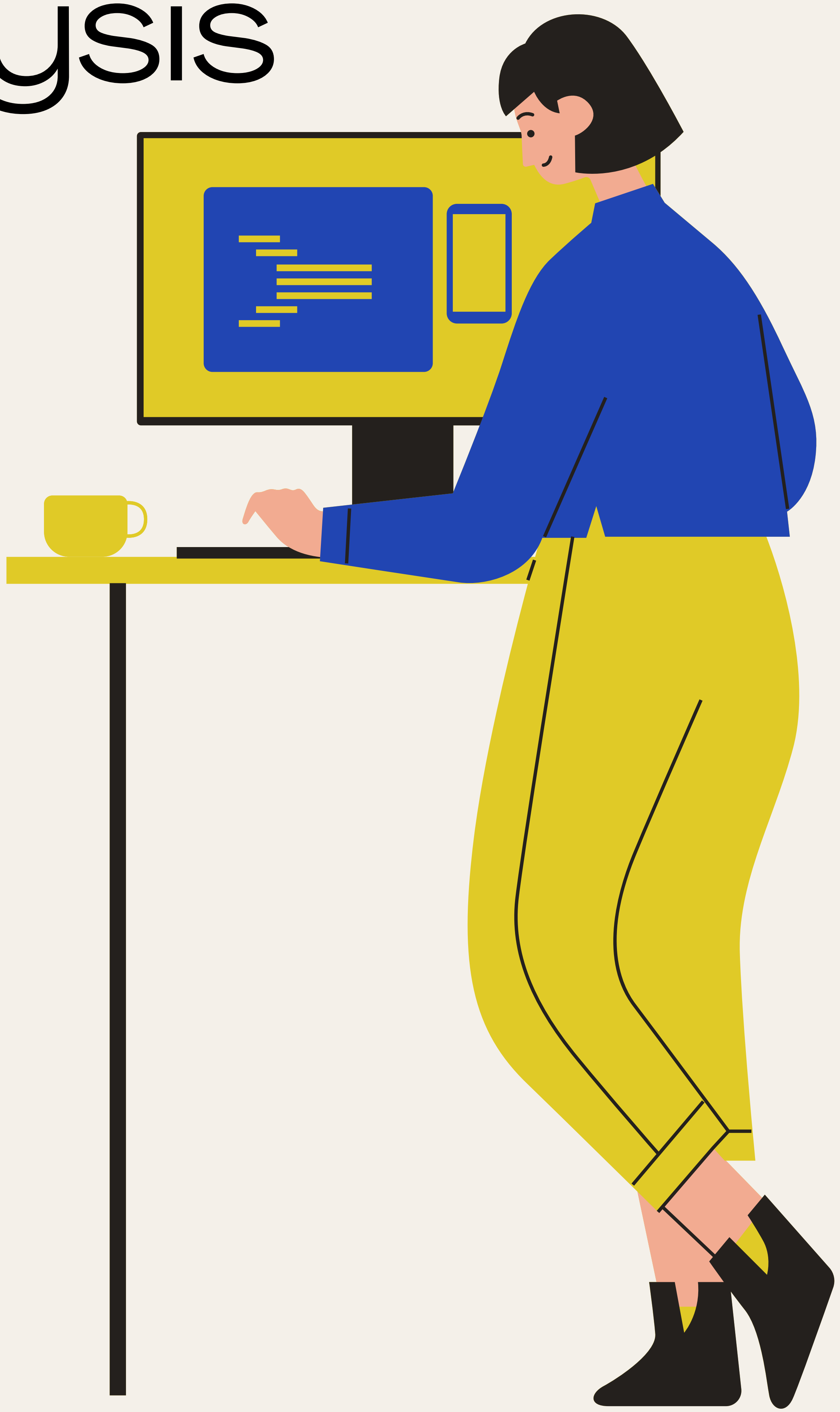
=>We need to modify these 2 columns

02 - Explore Data Analysis

Customer_idcolumn unnecessary for ML

```python
#Xóa cột không cần thiết
df_01=df.drop(columns='customer_id',axis=True)
df_01.head()
```

	credit_score	country	gender	age	tenure	balance	products_number	credit_card	active_member	estimated_salary	churn
0	619	France	Female	42	2	0.0	1	1	1	10134888.0	1
1	608	Spain	Female	41	1	8380786.0	1	0	1	11254258.0	0
2	502	France	Female	42	8	1596608.0	3	1	0	11393157.0	1
3	699	France	Female	39	1	0.0	2	0	0	9382663.0	0
4	850	Spain	Female	43	2	12551082.0	1	1	1	790841.0	0



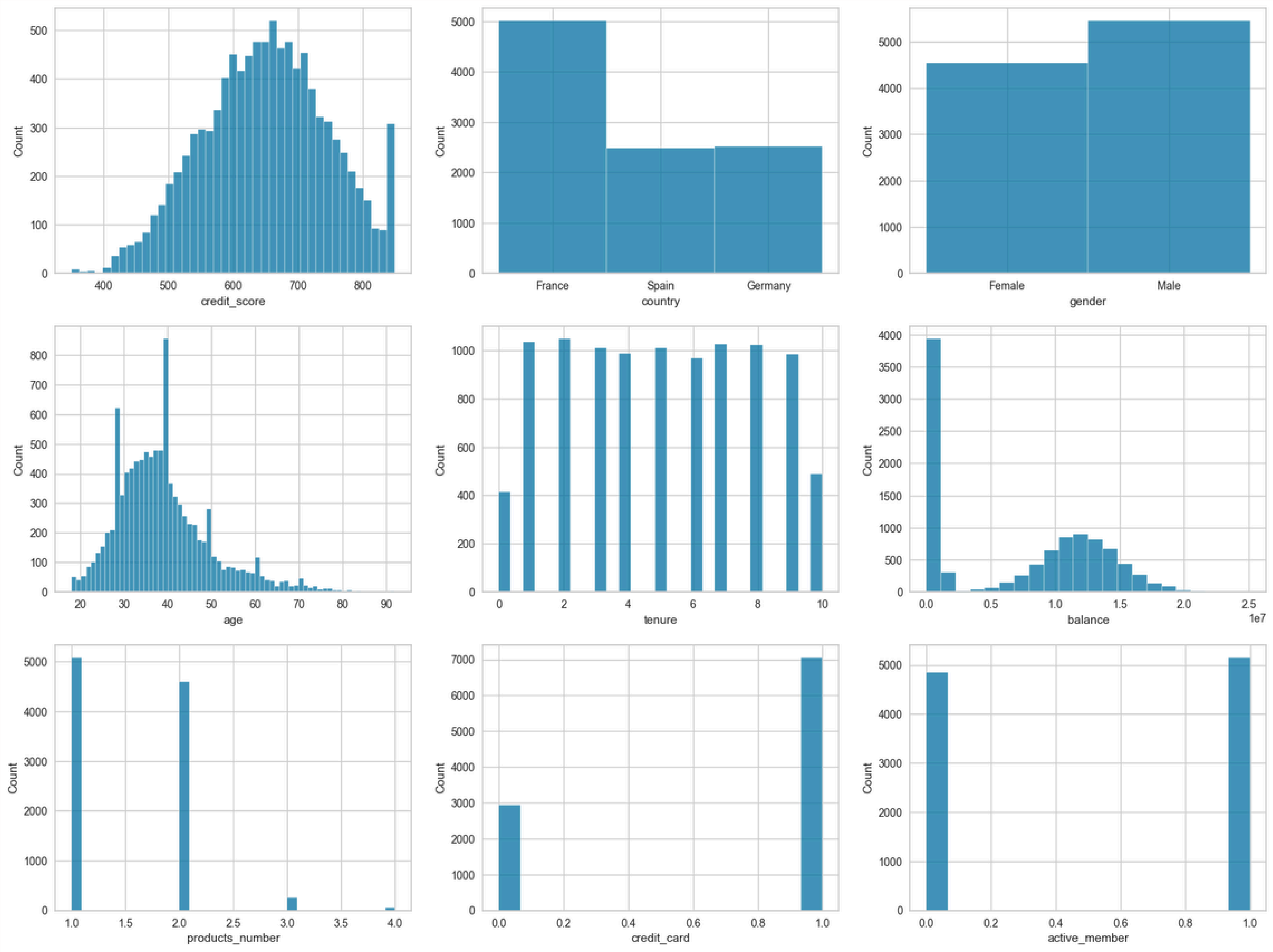
02 - Explore Data Analysis

Customer_idcolumn unnecessary for ML

```python
#Xóa cột không cần thiết
df_01=df.drop(columns='customer_id',axis=True)
df_01.head()
```

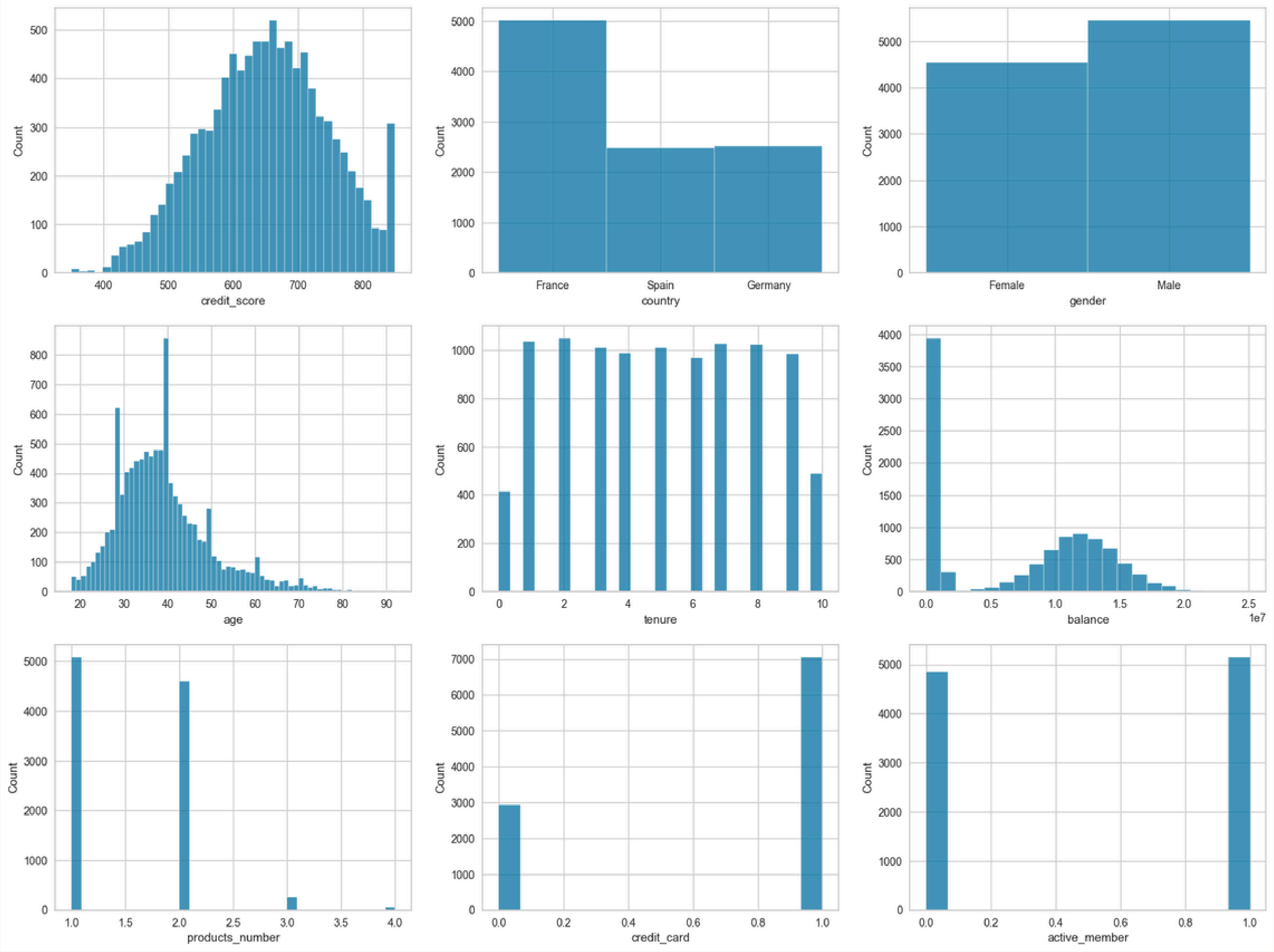
	credit_score	country	gender	age	tenure	balance	products_number	credit_card	active_member	estimated_salary	churn
0	619	France	Female	42	2	0.0	1	1	1	10134888.0	1
1	608	Spain	Female	41	1	8380786.0	1	0	1	11254258.0	0
2	502	France	Female	42	8	1596608.0	3	1	0	11393157.0	1
3	699	France	Female	39	1	0.0	2	0	0	9382663.0	0
4	850	Spain	Female	43	2	12551082.0	1	1	1	790841.0	0

02 - Explore Data Analysis



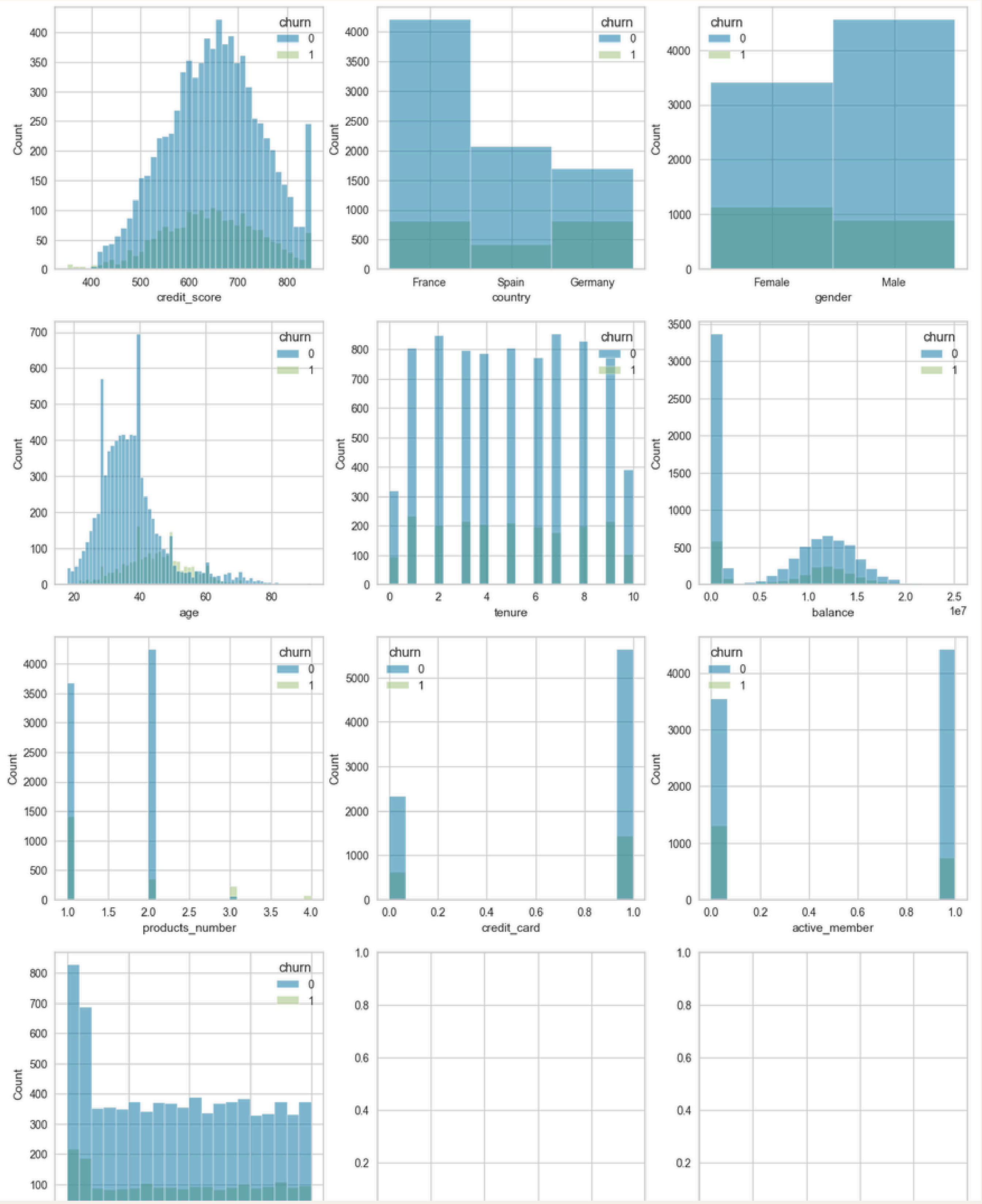
1. The Credit score column has an evenly distributed normal distribution. However, the credit score suddenly increased above 800
2. France has twice as many people as the other two countries
3. Female has approximately equal proportions as Male
4. The average age in this data set is 37.44 with the highest number of people, gradually decreasing from 60 to 92

02 - Explore Data Analysis



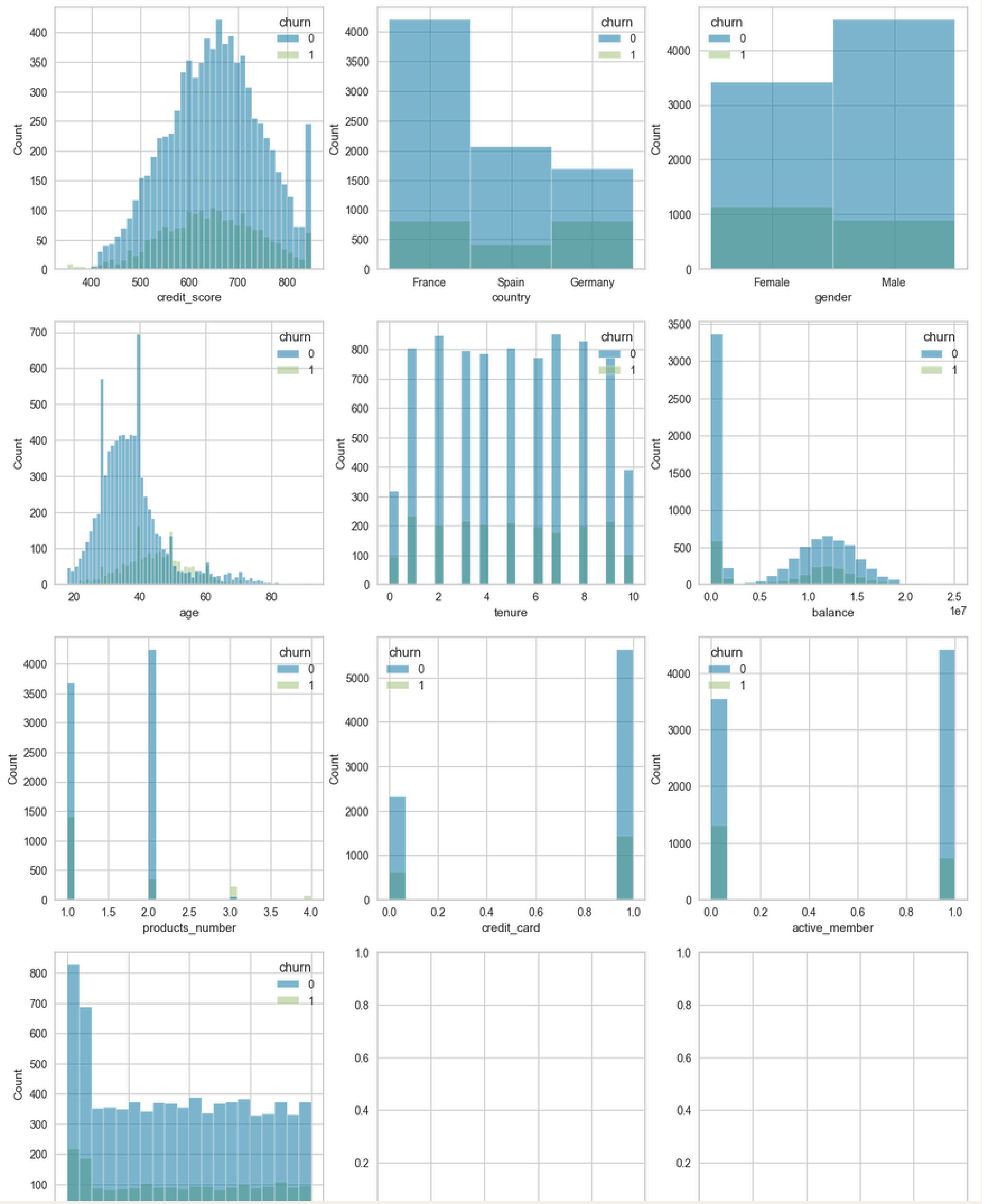
5. The number of years with an account is evenly spread, only years 0 and 10 have a low number of people
6. Many people have an account balance of 0? Does this affect the churn column?
7. The maximum number of people using the bank's products is 1 and 2 products. However, the number of people buying 3 to 4 products is very low? Does it affect customer churn?
8. The proportion of people who have credit cards is higher than those who do not use credit cards.
9. The ratio of people who are members of the bank is slightly higher than those who are not members? Does it affect customers leaving?

02 - Explore Data Analysis



1. Customer churn rate is spread evenly along with credit_score
2. Customer churn rate is higher in Germany than in other countries in the data set
3. The churn rate of women is higher than that of men
4. The age group with the number of people leaving the service is between 40 and 50 years old
5. The group of people with 0 years of service has the highest churn rate compared to the other groups

02 - Explore Data Analysis



6. Balance = 0 does not necessarily lead to customers leaving the service
7. The number of people leaving with a credit card is higher than the number of people without a card (this does not necessarily require a ratio calculation).
8. More non-members leave the service than members
9. The clear influence of salary on customer churn has not been seen

03 - Build Model ML

DATA PREPARATION

```
#Encoding Column Country
from sklearn.preprocessing import OneHotEncoder
df_01 = pd.get_dummies(df_01, columns=["country"],drop_first=False)
df_01
```

	credit_score	gender	age	tenure	balance	products_number	credit_card	active_member	estimated_salary	churn	country_France	country_Germany	country_Spain
0	619	Female	42	2	0.0	1	1	1	10134888.0	1	True	False	False
1	608	Female	41	1	8380786.0	1	0	1	11254258.0	0	False	False	True
2	502	Female	42	8	1596608.0	3	1	0	11393157.0	1	True	False	False
3	699	Female	39	1	0.0	2	0	0	9382663.0	0	True	False	False
4	850	Female	43	2	12551082.0	1	1	1	790841.0	0	False	False	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	771	Male	39	5	0.0	2	1	0	9627064.0	0	True	False	False
9996	516	Male	35	10	5736961.0	1	1	1	10169977.0	0	True	False	False
9997	709	Female	36	7	0.0	1	0	1	4208558.0	1	True	False	False
9998	772	Male	42	3	7507531.0	2	1	0	9288852.0	1	False	True	False
9999	792	Female	28	4	13014279.0	1	1	0	3819078.0	0	True	False	False

10000 rows × 13 columns

Using one-hot Encoding for the Country column to transform the text to numeric and make independence variables

03 - Build Model ML

DATA PREPARATION

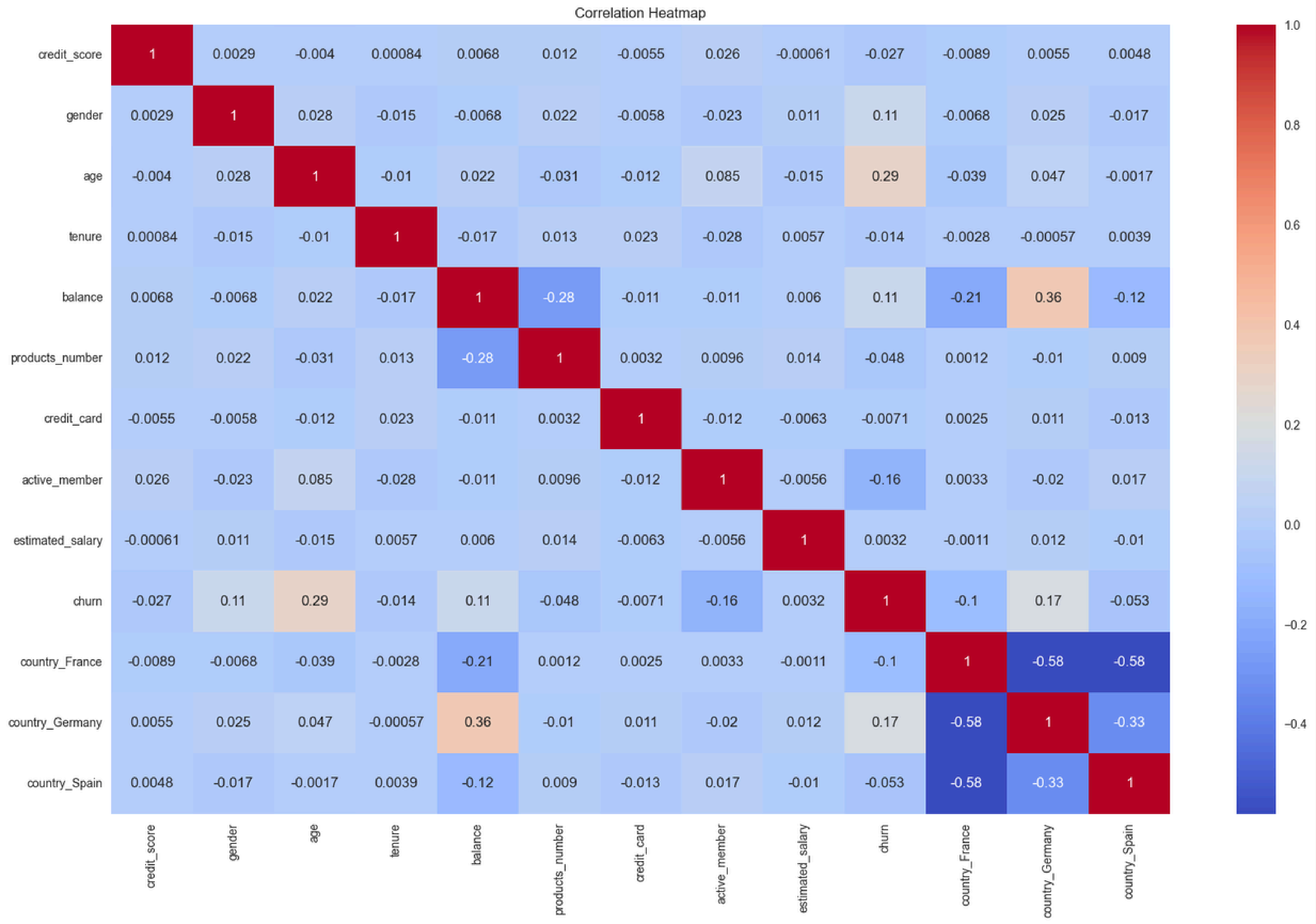
```python
df_01['gender'] = df['gender'].map({'Male':0, 'Female':1})
df_01
```

	credit_score	gender	age	tenure	balance	products_number	credit_card	active_member	estimated_salary	churn	country_France	country_Germany	country_Spain
0	619	1	42	2	0.0	1	1	1	10134888.0	1	True	False	False
1	608	1	41	1	8380786.0	1	0	1	11254258.0	0	False	False	True
2	502	1	42	8	1596608.0	3	1	0	11393157.0	1	True	False	False
3	699	1	39	1	0.0	2	0	0	9382663.0	0	True	False	False
4	850	1	43	2	12551082.0	1	1	1	790841.0	0	False	False	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	771	0	39	5	0.0	2	1	0	9627064.0	0	True	False	False
9996	516	0	35	10	5736961.0	1	1	1	10169977.0	0	True	False	False
9997	709	1	36	7	0.0	1	0	1	4208558.0	1	True	False	False
9998	772	0	42	3	7507531.0	2	1	0	9288852.0	1	False	True	False
9999	792	1	28	4	13014279.0	1	1	0	3819078.0	0	True	False	False

Using map() to transform string data to numeric the Gender colums

03 - Build Model ML

DATA PREPARATION



The variable 'estimated_salary' has a moderate negative correlation with churn (-0.16), meaning that customers with higher salaries are less likely to churn.
The variables 'country_France' and 'country_Germany' have a fairly high positive correlation (0.17) with churn, which suggests that customers in France and Germany have a higher tendency to churn.
The variable 'balance' has a weak negative correlation (-0.21) with churn, customers with higher balances are less likely to churn.

03 - Build Model ML

DATA PREPARATION

```
X = df_01.drop(columns=['churn'])
Y = df_01['churn']
```

In machine learning, the "bank churn customer" problem is often considered a classification problem. The goal of the problem is to predict whether a customer will convert or not based on customer characteristics and information.
The label is the “churn” columns and all another columns is the feature

03 - Build Model ML

DATA PREPARATION

```
churn
0    7963
1    2037
Name: count, dtype: int64
```

```python
from imblearn.over_sampling import SMOTE

smote = SMOTE()

X, Y = smote.fit_resample(X, Y)
```

Churn column is imbalance data.To balance data using over sampling for balance this label

03 - Build Model ML

```
trained Logistic Regression in 0.02 s
trained Nearest Neighbors in 0.03 s
trained Linear SVM in 2.12 s
trained Gradient Boosting Classifier in 1.42 s
trained Decision Tree in 0.05 s
trained Random Forest in 0.20 s
trained Neural Net in 2.44 s
trained Naive Bayes in 0.00 s
                    classifier  train_score  training_time
4                Decision Tree     1.000000       0.046875
5                Random Forest     0.996591       0.203125
1            Nearest Neighbors     0.888500       0.031250
2                   Linear SVM     0.870022       2.125000
3  Gradient Boosting Classifier     0.869752       1.421875
6                   Neural Net     0.858809       2.437500
0          Logistic Regression     0.808576       0.015625
7                  Naive Bayes     0.786598       0.000000
```

-Model type: Different models were trained, including Logistic Regression, Nearest Neighbors, Linear SVM, Gradient Boosting Classifier, Decision Tree, Random Forest, Neural Net, and Naive Bayes.
-Training time: The time (in seconds) it took each model to be trained on the dataset.
-Training score (train_score): Score that evaluates the performance of each model on the training data set.

03 - Build Model ML

```
trained Logistic Regression in 0.02 s
trained Nearest Neighbors in 0.03 s
trained Linear SVM in 2.12 s
trained Gradient Boosting Classifier in 1.42 s
trained Decision Tree in 0.05 s
trained Random Forest in 0.20 s
trained Neural Net in 2.44 s
trained Naive Bayes in 0.00 s
                   classifier  train_score  training_time
4               Decision Tree     1.000000       0.046875
5               Random Forest     0.996591       0.203125
1           Nearest Neighbors     0.888500       0.031250
2                  Linear SVM     0.870022       2.125000
3  Gradient Boosting Classifier     0.869752       1.421875
6                  Neural Net     0.858809       2.437500
0         Logistic Regression     0.808576       0.015625
7                 Naive Bayes     0.786598       0.000000
```

-Model type: Different models were trained, including Logistic Regression, Nearest Neighbors, Linear SVM, Gradient Boosting Classifier, Decision Tree, Random Forest, Neural Net, and Naive Bayes.
-Training time: The time (in seconds) it took each model to be trained on the dataset.
-Training score (train_score): Score that evaluates the performance of each model on the training data set.

03 - Build Model ML

Overfitting

```
Model: Desicion Tree
Độ chính xác của mô hình Decision tree trên tập Train:  100.0 %
Độ chính xác của mô hình Decision tree trên tập Test: 83.01
```

```
Model: Random Forest
Độ chính xác của mô hình Random Forest trên tập Train:  100.0 %
Độ chính xác của mô hình Random Forest trên tập Test: 87.88
```

As we can see above: With the Decision Tree model when training TRAIN, the accuracy reaches 100% (very high), however that model when used for the TEST set, the accuracy only reaches 82.46% (very low). Similar to the Random Forest model, the training accuracy is ~100%, however the accuracy on the TEST set is only 87.71%

==> OVERFITTING (The phenomenon where the model has high accuracy when TRAIN is high (small error) but when run with TEST data, the accuracy is low (high error)

03 - Build Model ML

```python
from sklearn.model_selection import cross_val_score

# Logistic Regression
log_reg = LogisticRegression(solver='lbfgs', max_iter=5000)
log_scores = cross_val_score(log_reg, X_train_sc, y_train, cv=5)
log_reg_mean = log_scores.mean()

# SVC
svc_clf = SVC(gamma='auto')
svc_scores = cross_val_score(svc_clf, X_train_sc, y_train, cv=5)
svc_mean = svc_scores.mean()

# KNearestNeighbors
knn_clf = KNeighborsClassifier()
knn_scores = cross_val_score(knn_clf, X_train_sc, y_train, cv=5)
knn_mean = knn_scores.mean()

# Decision Tree
tree_clf = tree.DecisionTreeClassifier()
tree_scores = cross_val_score(tree_clf, X_train_sc, y_train, cv=5)
tree_mean = tree_scores.mean()

# Gradient Boosting Classifier
grad_clf = GradientBoostingClassifier()
grad_scores = cross_val_score(grad_clf, X_train_sc, y_train, cv=5)
grad_mean = grad_scores.mean()

# Random Forest Classifier
rand_clf = RandomForestClassifier(n_estimators=18)
rand_scores = cross_val_score(rand_clf, X_train_sc, y_train, cv=5)
rand_mean = rand_scores.mean()

# NeuralNet Classifier
neural_clf = MLPClassifier(alpha=1)
neural_scores = cross_val_score(neural_clf, X_train_sc, y_train, cv=5)
neural_mean = neural_scores.mean()

# Naives Bayes
nav_clf = GaussianNB()
nav_scores = cross_val_score(nav_clf, X_train_sc, y_train, cv=5)
nav_mean = neural_scores.mean()

# Create a Dataframe with the results.
d = {'Classifiers': ['Logistic Reg.', 'SVC', 'KNN', 'Dec Tree', 'Grad B CLF', 'Rand FC', 'Neural Classifier', 'Naives Bayes'],
    'Crossval Mean Scores': [log_reg_mean, svc_mean, knn_mean, tree_mean, grad_mean, rand_mean, neural_mean, nav_mean]}
```

	Classifiers	Crossval Mean Scores
5	Rand FC	0.868586
4	Grad B CLF	0.861769
1	SVC	0.858091
6	Neural Classifier	0.856118
7	Naives Bayes	0.856118
2	KNN	0.837998
3	Dec Tree	0.815391
0	Logistic Reg.	0.807499

Using Cross-Validiation to repair overfitting model

03 - Build Model ML

Choose model

```
TRAIN: Gradient Boost Classifier accuracy is 0.8618
```

```
TRAIN: Neural classifier accuracy is 0.8556
```

```
TRAIN: Navie Bayes accuracy is 0.7862
```

```
TRAIN: Random Forest Classifer accuracy is 0.8662
```

Select Random Forest is the best model with high accuaracy about 86.62%

03 - Build Model ML

Evaluate

```
from sklearn.metrics import accuracy_score
print("TEST: Random Forest Classifier accuracy is %2.4f" % accuracy_score(y_test,y_predicted))
```

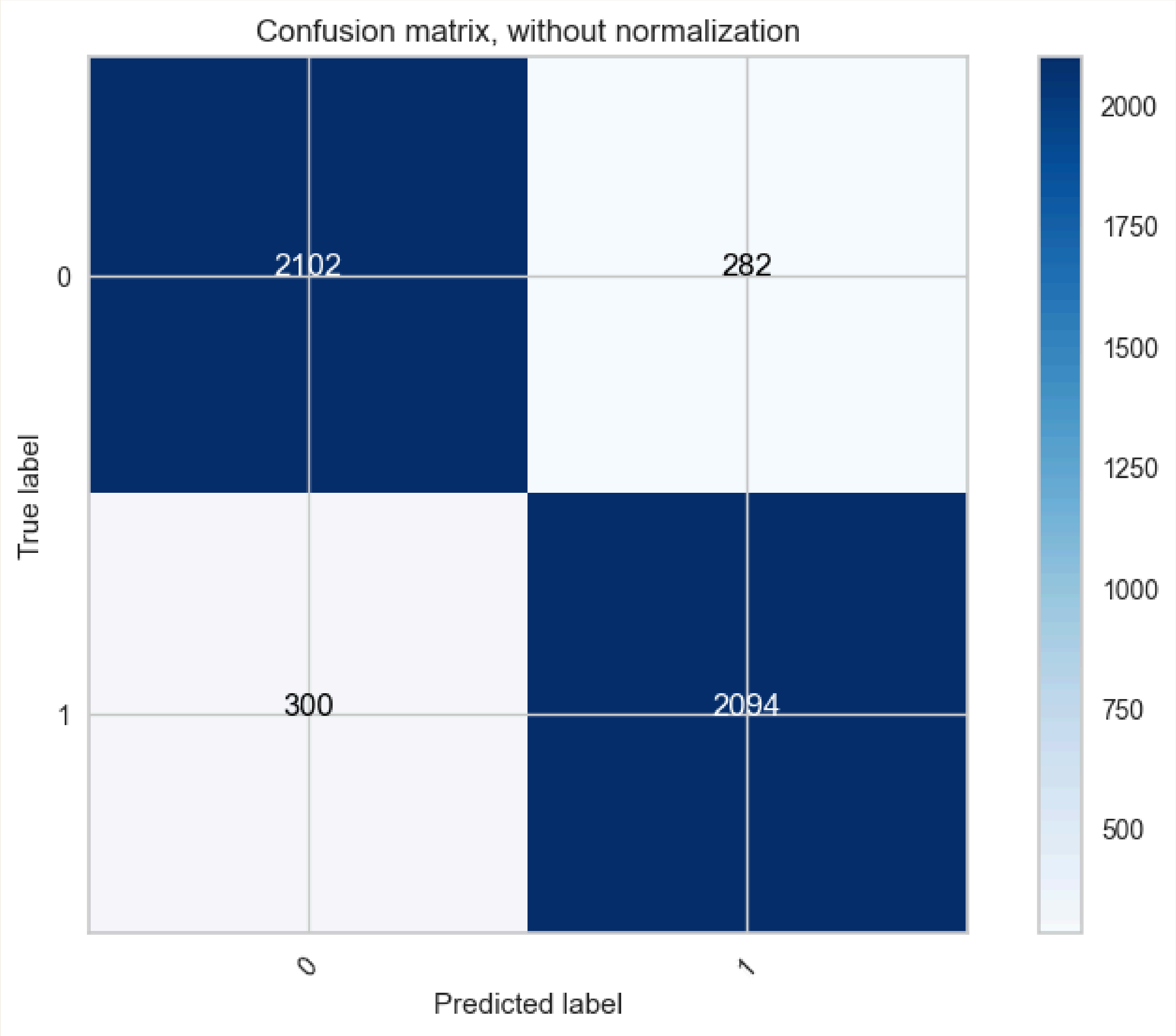
```
TEST: Random Forest Classifier accuracy is 0.8782
```

In the Test File, Random forest have 87.82% higher than the train file but no difference.

Random forest is a supervised learning algorithm that can solve both regression and classification problems.

03 - Build Model ML

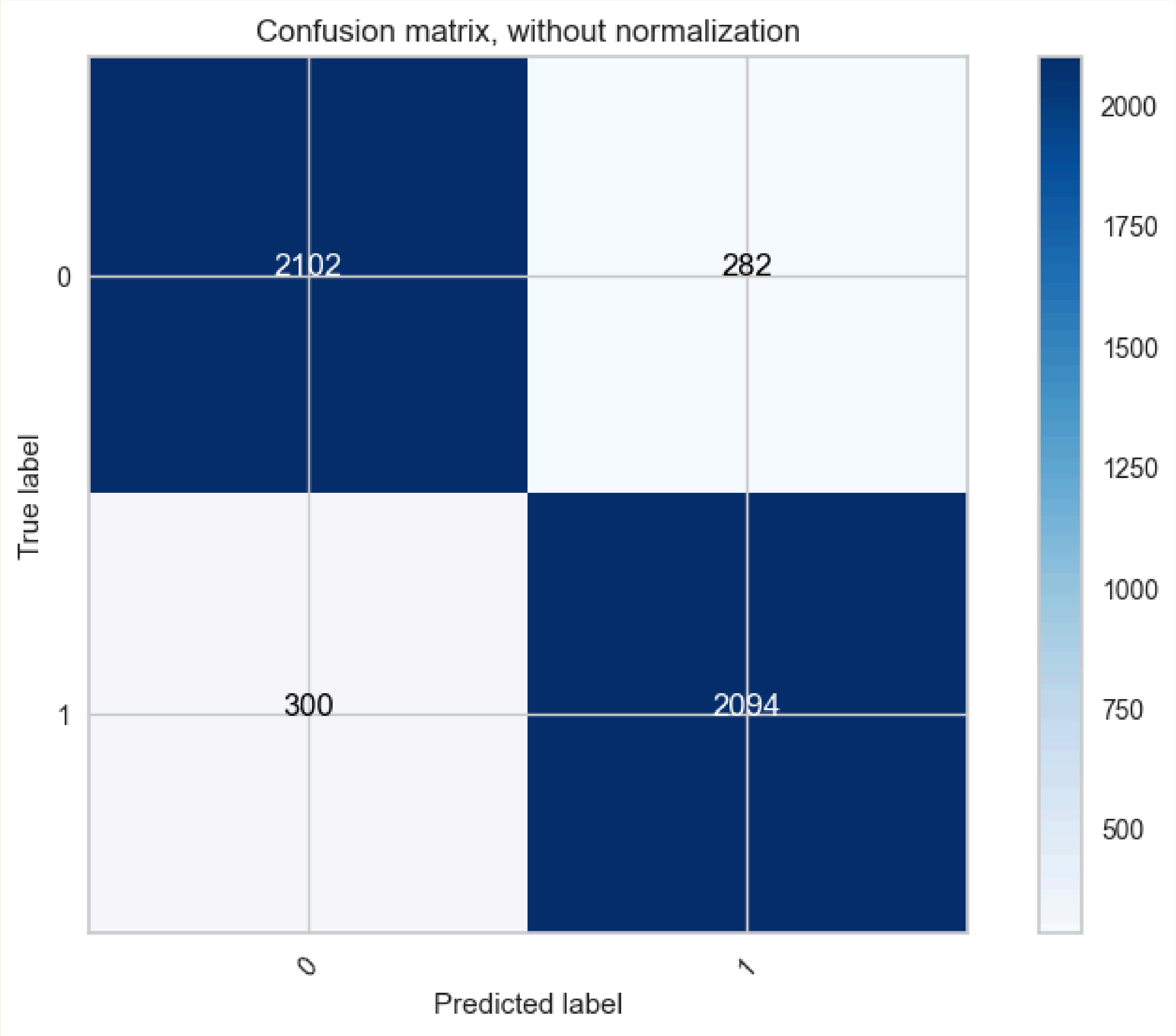
Evaluate



Confusion matrix is a square matrix with each dimension equal to the number of data layers. The value in the ith row and jth column is the number of points that should belong to class i but are predicted to belong to class j.

03 - Build Model ML

Evaluate

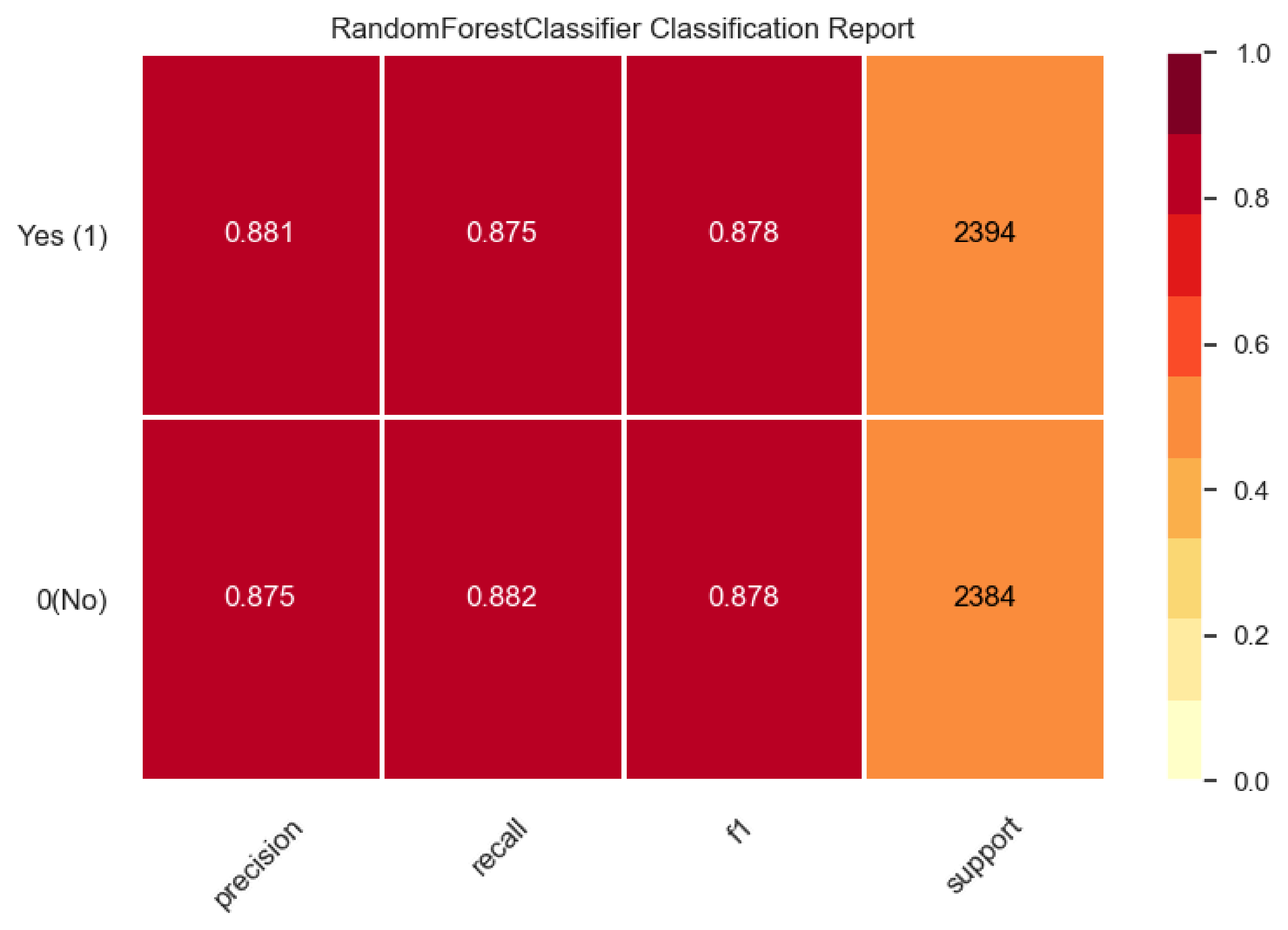


Thus, looking at the confusion matrix (without normalization):
* Row (0), column (0): indicates the number of points in class 0 that are correctly classified into class 0 (2102 points.
* Row (0), column (1): Number of points belonging to class 0 but assigned to class 1 (wrong class) 282 points.
* Row (1), Column (0): Number of points belonging to class 1 but assigned to class 0 (wrong class) 300 points
* Row (1), column (1): Number of points in class 1 correctly classified into class 1 (2094 points)

We can immediately deduce that the sum of the elements in this entire matrix is the number of points in the TEST set. The elements on the diagonal of the matrix are the number of correctly classified points of each data class. From here, it can be deduced that the accuracy is equal to the sum of the elements on the diagonal divided by the sum of the elements of the entire matrix.

03 - Build Model ML

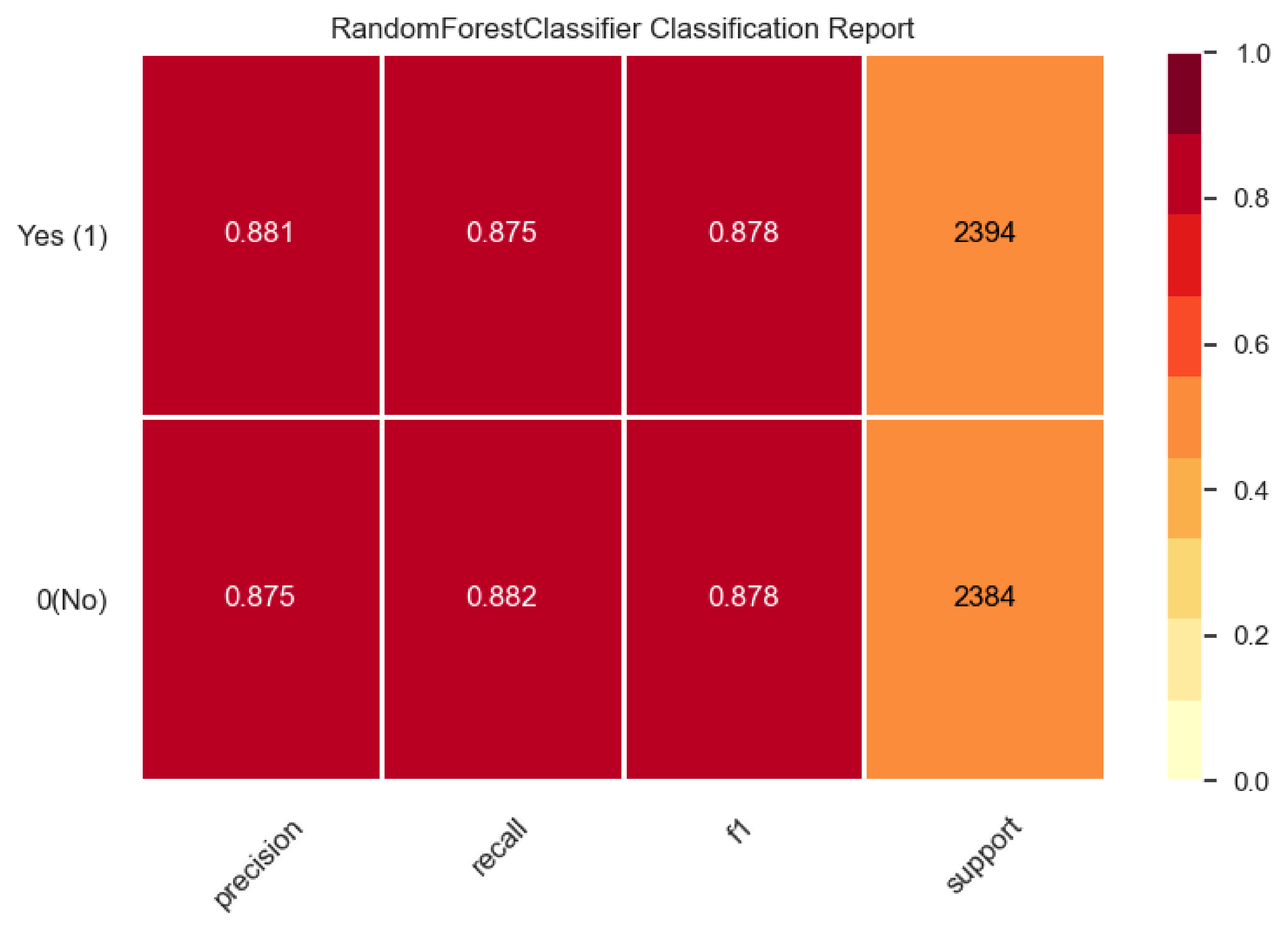
Evaluate



This is a classification report for the Random Forest Classifier model. This report provides key performance metrics for model evaluation on a test data set for a binary classification problem.

03 - Build Model ML

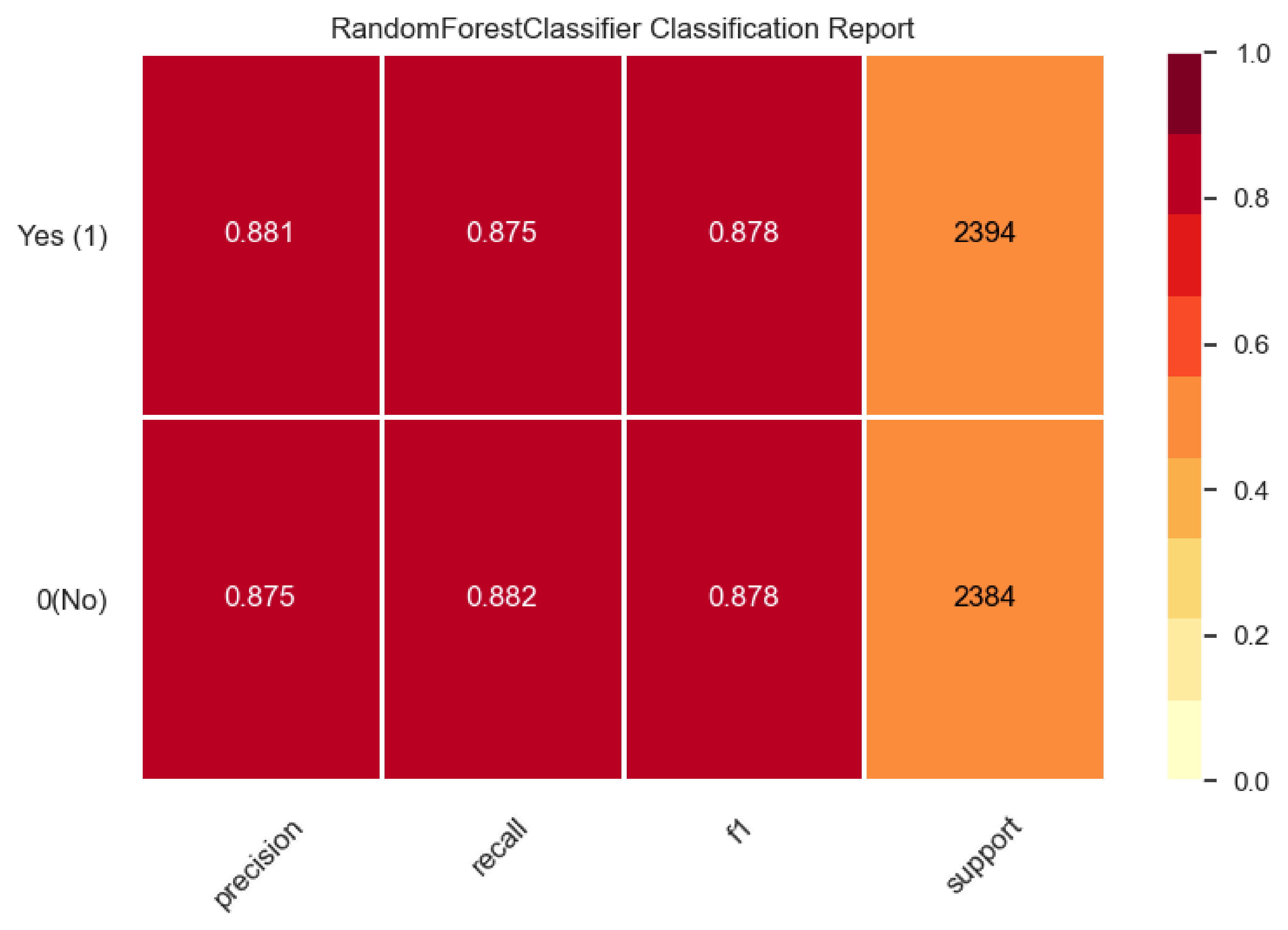
Evaluate



Measures presented include:
Precision: Measures the percentage of samples predicted to be class "Yes (1)" that actually belong to this class. The model's precision is 0.881, relatively high.
Recall: Measures the percentage of samples belonging to class "Yes (1)" that are correctly predicted. The recall of the model is 0.875.
F1-score: Harmonic average of Precision and Recall. The model's F1-score is 0.878.
Support: Number of samples in the test set for each class. There are 2394 samples belonging to class "Yes (1)" and 2384 samples belonging to class "No (0)".

03 - Build Model ML

Evaluate



From this report, we can see that Random Forest Classifier has good performance in data classification, with Precision, Recall and F1-score measures all at high levels, 0.881, 0.875 and 0.878 respectively for the class. "Yes (1)". This shows that the model is able to accurately predict samples of the "Yes (1)" class as well as not miss too many samples of this class.

The problem of customer churn is a major challenge for banks and financial institutions. Losing customers not only leads to reduced revenue but also increases marketing costs to attract new customers, affecting profits and sustainable development of the business.
By building a churn prediction model, banks can identify customers at risk of leaving and devise appropriate retention strategies and programs such as improving services and offering attractive incentives. , thereby minimizing churn rate.

Data analysis allows for identifying trends and patterns within datasets.

04 - Conclusions

04 - Conclusions

The Random Forest model was selected with a high accuracy of 86.62% on the test set. This model uses ensemble learning by combining multiple decision trees to improve prediction performance and avoid overfitting.
Important features such as estimated salary, country, account balance have a significant influence on churn rate, and are used in the Random Forest model for more accurate predictions.

Data analysis helps in identifying outliers or anomalies in the data

Data

Visualization

Data analysis facilitates predictive modeling and forecasting

04 - Conclusions

Data analysis helps in identifying outliers or anomalies in the data

In addition to high accuracy, this model also achieves precision of 0.881, recall of 0.875 and F1-score of 0.878 which is good for the customer churn class. This helps banks have more accurate predictions for customers at risk of leaving, thereby offering appropriate retention solutions. With an effective churn prediction model, banks can focus on improving customer experience, recommending products and services suitable for each audience, thereby improving customer satisfaction and loyalty. customers, reduce marketing costs and maintain sustainable growth rates.

Data

Visualization

Data analysis facilitates predictive modeling and forecasting

Thanks

Cao Thanh Phat