Churn Iran

05-06-2024

CHURN IRAN REPORT

TABLE OF CONTENS

Contents

ABOUT THE DATA

This dataset is randomly collected from an Iranian telecom company database over a period of 12 months

LETS GET STARTED

DATASET
ADDITIONAL INFORMATION

This dataset is randomly collected from an Iranian telecom company database over a period of 12 months. A total of 3150 rows of data, each representing a customer, bear information for 13 columns. The attributes that are in this dataset are call failures, frequency of SMS, number of complaints, number of distinct calls, subscription length, age group, the charge amount, type of service, seconds of use, status, frequency of use, and Customer Value.

All of the attributes except for attribute churn is the aggregated data of the first 9 months. The churn labels are the state of the customers at the end of 12 months. The three months is the designated planning gap.

ANONYMOUS CUSTOMER ID

1. **Call Failures:** number of call failures
2. **Complains:** binary (0: No complaint, 1: complaint)
3. **Subscription Length:** total months of subscription
4. **Charge Amount:** Ordinal attribute (0: lowest amount, 9: highest amount)
5. **Seconds of Use:** total seconds of calls
6. **Frequency of use:** total number of calls
7. **Frequency of SMS:** total number of text messages
8. **Distinct Called Numbers:** total number of distinct phone calls
9. **Age Group:** ordinal attribute (1: younger age, 5: older age)
10. **Tariff Plan:** binary (1: Pay as you go, 2: contractual)
11. **Status:** binary (1: active, 2: non-active)
12. **Customer Value:** The calculated value of customer
13. **Churn:** binary (1: churn, 0: non-churn) - Class label

Churn Iran

05-06-2024

PRE-PROCESSING DATA

Step 1: EDA

Step 2: T-C-R

Churn Iran

05-06-2024

EXPLORE DATA ANALYSIS

PRE-PROCESSING DATA EDA

LOYAL CUSTOMERS

The majority of customers have no complaints and do not churn. Moderate service usage: The duration and frequency of service usage are concentrated at an average level.

LOW SERVICE FEES

Most customers pay low service fees

LOW SMS USAGE

The frequency of SMS usage is low

YOUNG AGE GROUP

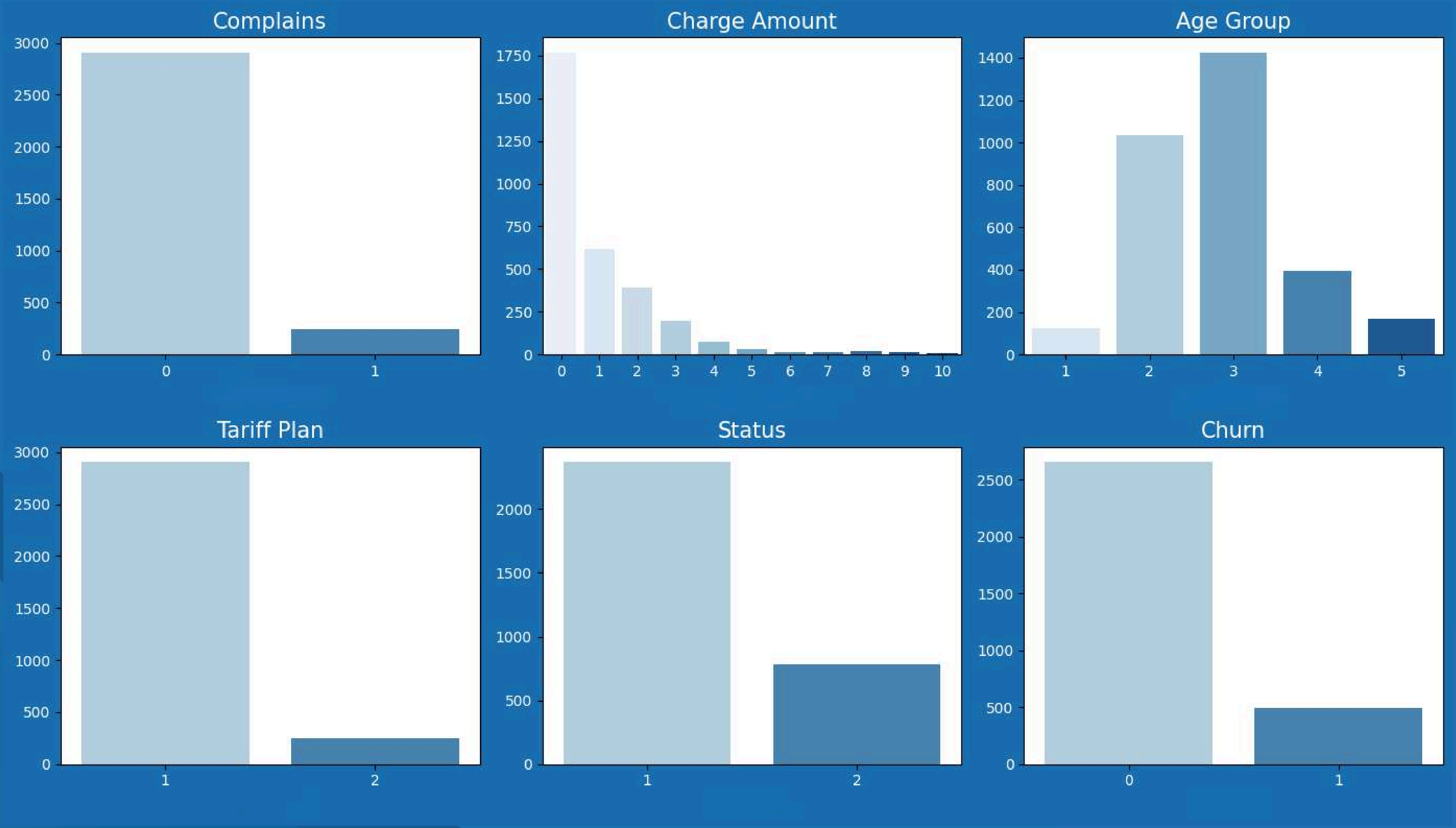
Customers are mainly in the young age group

POPULAR PACKAGE 1

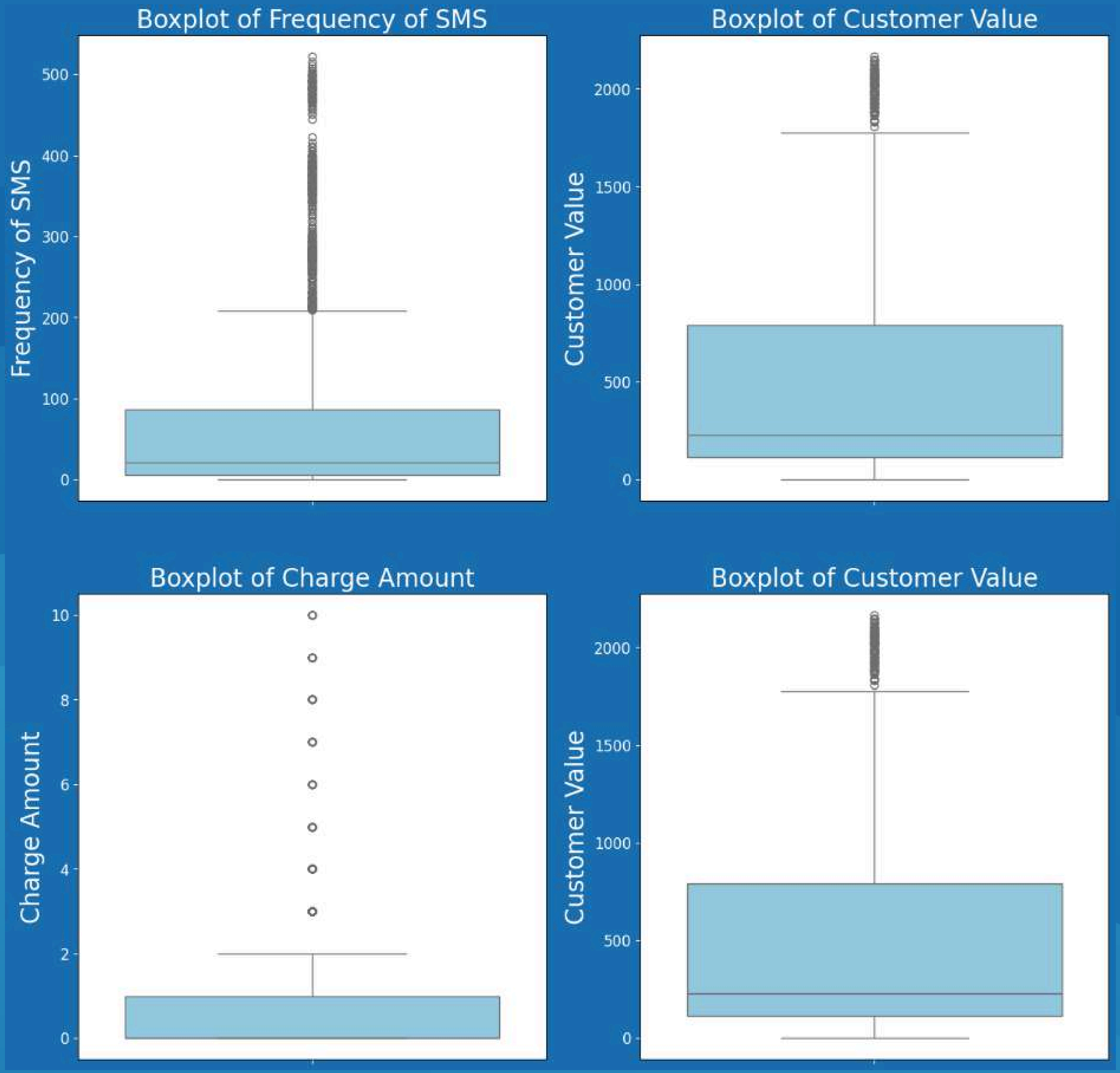
Most customers use package 1

	Call Failure	Complains	Subscription Length	Charge Amount	Seconds of Use	Frequency of use	Frequency of SMS	Distinct Called Numbers	Age Group	Tariff Plan	Status	Age	Customer Value	Churn
count	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000	3150.000000
mean	7.627937	0.076508	32.541905	0.942857	4472.459683	69.460635	73.174921	23.509841	2.826032	1.077778	1.248254	30.998413	470.972916	0.157143
std	7.263886	0.265851	8.573482	1.521072	4197.908687	57.413308	112.237560	17.217337	0.892555	0.267864	0.432069	8.831095	517.015433	0.363993
min	0.000000	0.000000	3.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	15.000000	0.000000	0.000000
25%	1.000000	0.000000	30.000000	0.000000	1391.250000	27.000000	6.000000	10.000000	2.000000	1.000000	1.000000	25.000000	113.801250	0.000000
50%	6.000000	0.000000	35.000000	0.000000	2990.000000	54.000000	21.000000	21.000000	3.000000	1.000000	1.000000	30.000000	228.480000	0.000000
75%	12.000000	0.000000	38.000000	1.000000	6478.250000	95.000000	87.000000	34.000000	3.000000	1.000000	1.000000	30.000000	788.388750	0.000000
max	36.000000	1.000000	47.000000	10.000000	17090.000000	255.000000	522.000000	97.000000	5.000000	2.000000	2.000000	55.000000	2165.280000	1.000000

CUSTOMER INSIGHT



PRE-PROCESSING DATA EDA



FREQUENCY OF SMS

The majority of customers have a low frequency of SMS messages (under 100 messages). There are some customers with a higher frequency of SMS messages, forming outliers on the chart.

CHARGE AMOUNT

The majority of customers have a charge amount concentrated under 3 units. There are a few customers with significantly higher charge amounts, forming outliers.

CUSTOMER VALUE

Customer value has a relatively even distribution, ranging from 0 to nearly 2000. There are a few customers with very high value, forming outliers

DATA PREPARATION BEFORE RUNNING ALGORITHMS

0.53

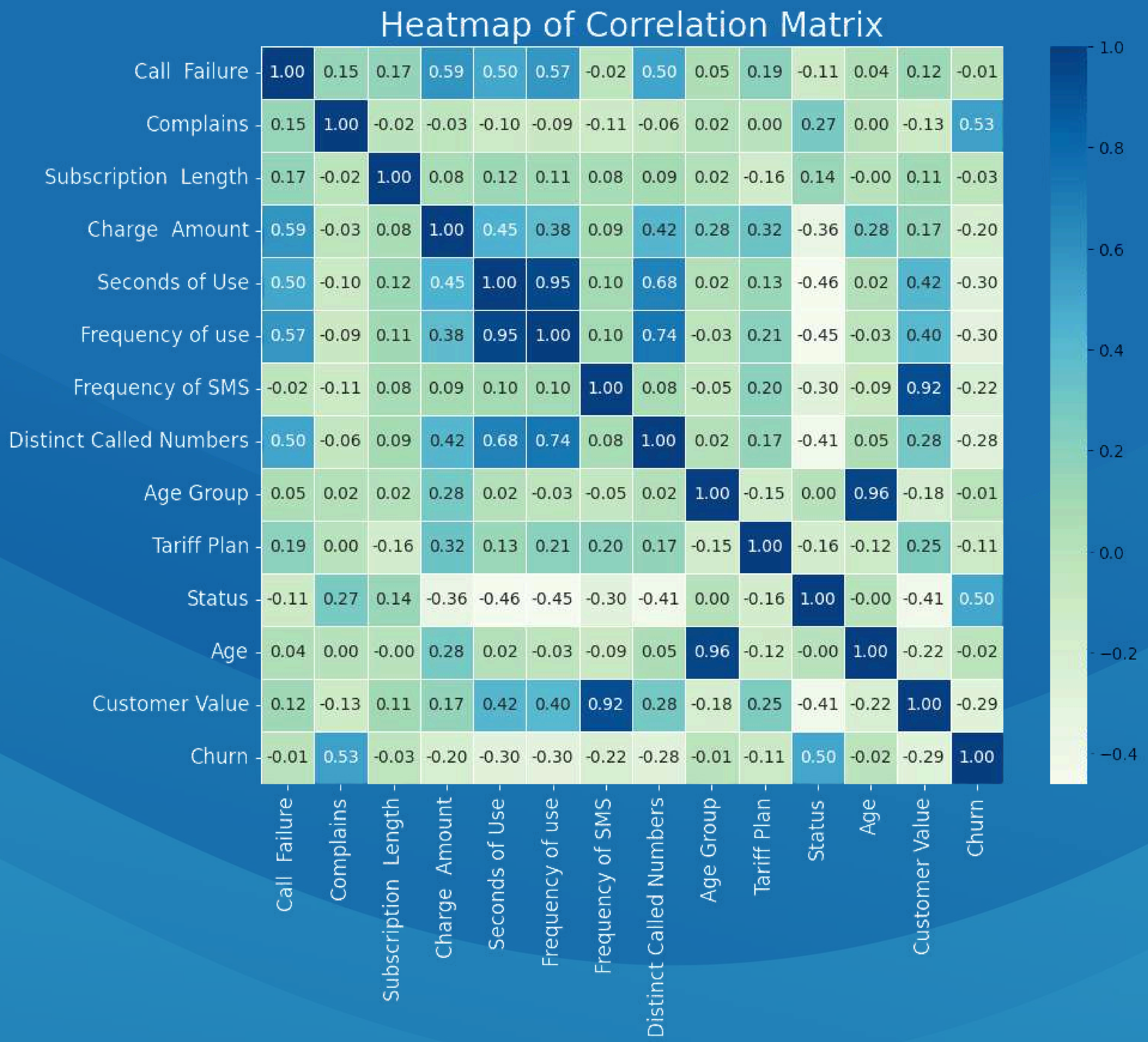
CHURN & COMPLAINS

There is a moderate correlation (0.53). This indicates that customers who complain frequently are more likely to churn.

0.50

CALL FAILURE & SECONDS OF USE

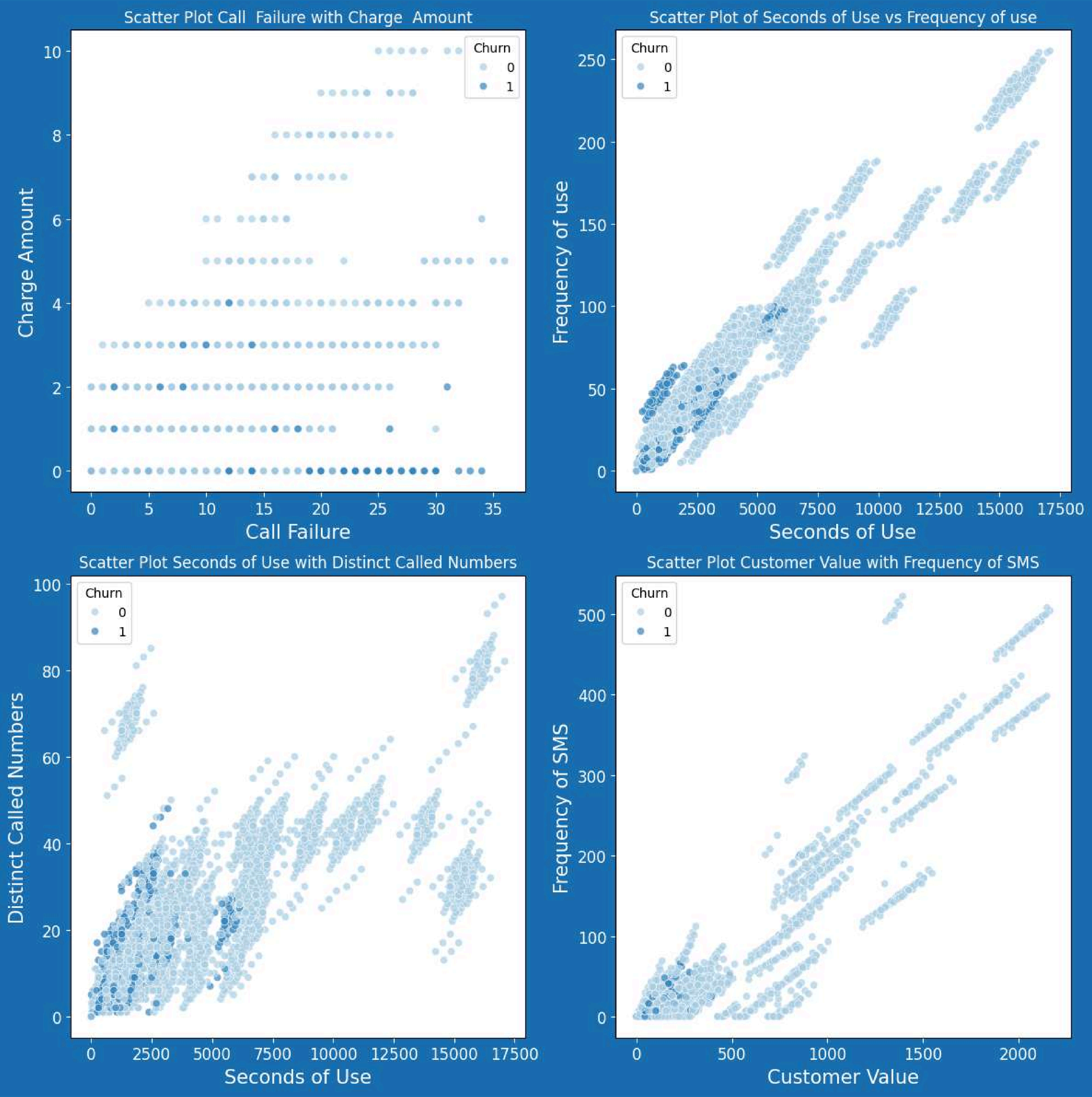
There is a moderate negative correlation (0.50). This may imply that customers who experience more call incidents tend to use the service less.

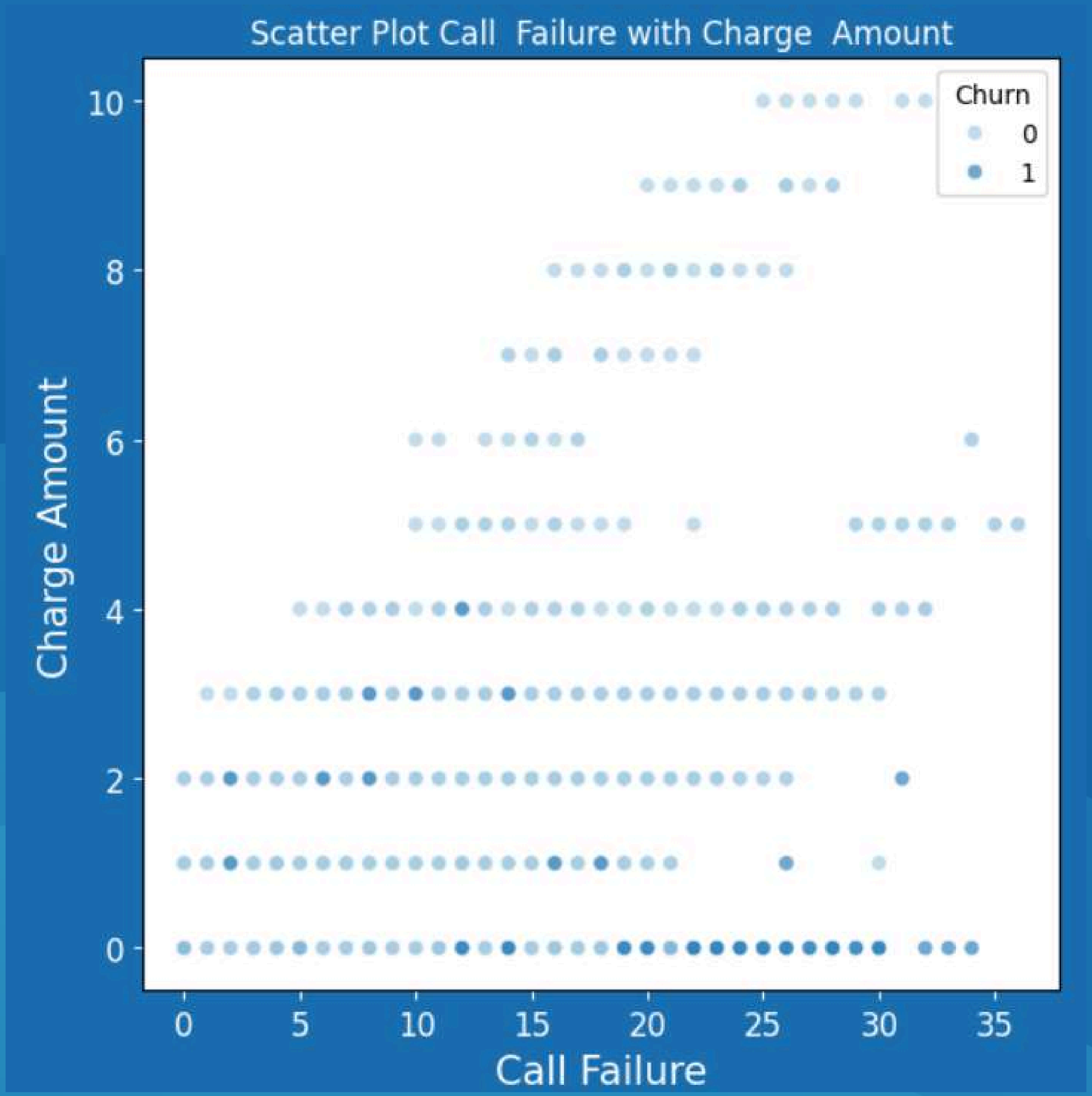


CHURN RATE REPORT

CUSTOMER VALUE & FREQUENCY OF SMS

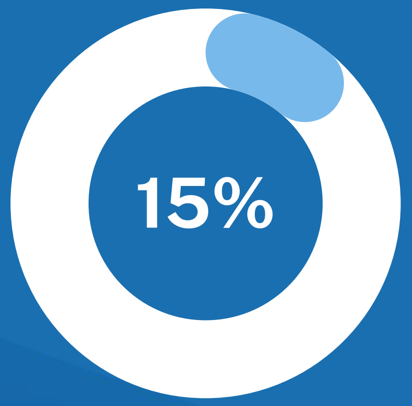
1. **Call Failure & Charge Amount:** No clear relationship between the number of failed calls and payment amount. Neither factor significantly affects customer churn.
2. **Seconds of Use & Frequency of Use:** Strong positive correlation between duration and frequency of use, but no significant difference between churned and retained customers.
3. **Seconds of Use & Distinct Called Numbers:** No clear relationship between duration of use and the number of different phone numbers called. Neither factor affects customer churn.



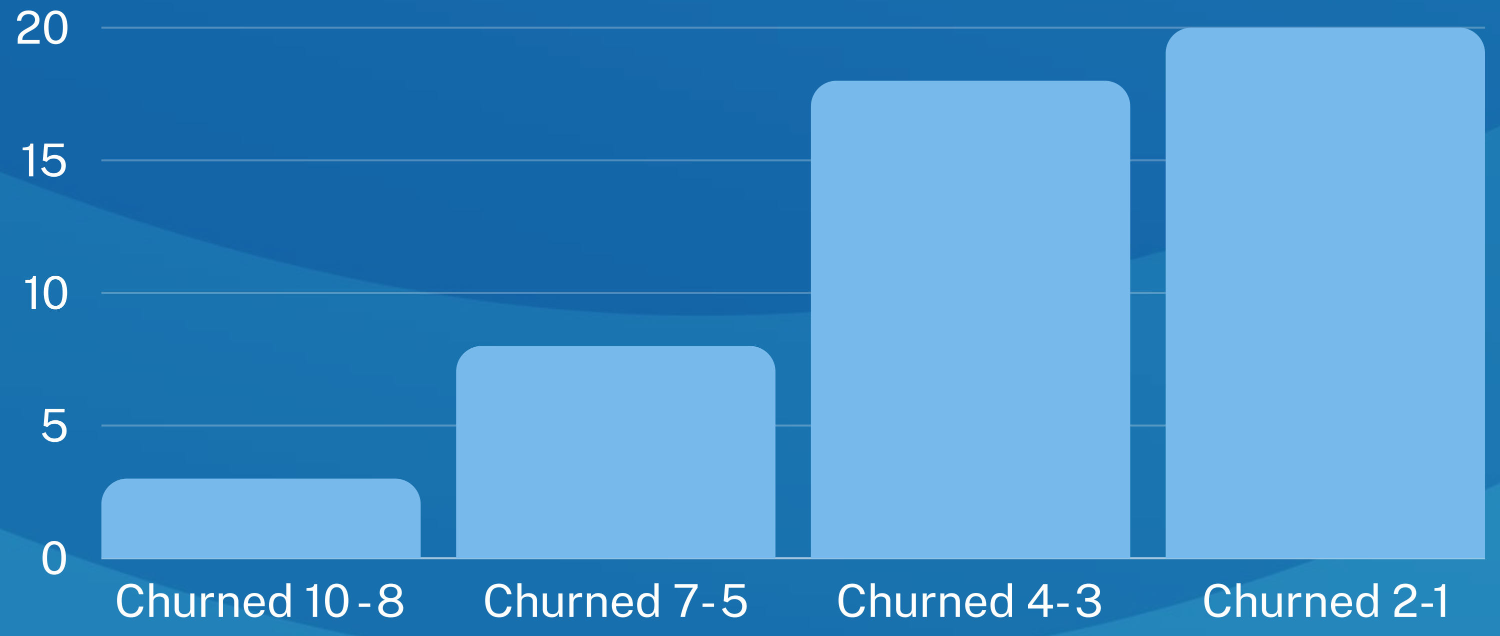


There is a strong positive correlation: Customers with higher value tend to send more SMS messages.

CHURN RATE REPORT



Churn is concentrated in the group with low value and low SMS frequency: This suggests that increasing promotions, focusing on high-value customer care, and encouraging SMS usage can help reduce churn.



Churn Iran

05-06-2024

TRAIN - SCALING - RESAMPLING

PRE-PROCESSING DATA EDA

TEST TRAIN SPLIT

```
[ ]  from sklearn.model_selection import train_test_split
```

```
[ ]  X=df.drop(['Churn'],axis=1)
     y=df['Churn']
```

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```

PRE-PROCESSING DATA EDA

SCALING

```
[ ]  from sklearn.preprocessing import StandardScaler

[ ]  # Initialize StandardScaler
     scaler = StandardScaler()

[ ]  # Apply StandardScaler to training and test data
     X_train_scaled = scaler.fit_transform(X_train)
     X_test_scaled = scaler.transform(X_test)
```

PRE-PROCESSING
DATA EDA

RESAMPLING

```
[ ] from collections import Counter
```

```
[ ] # Under-Sampling
    from imblearn.under_sampling import RandomUnderSampler
    rus = RandomUnderSampler(random_state=42, replacement=True)
```

```
[ ] # fit predictor and target varialbe
    X_rus, y_rus = rus.fit_resample(X_train_scaled, y_train)

    print('original dataset shape:', Counter(y))
    print('Resample dataset shape', Counter(y_rus))
```

```
original dataset shape: Counter({0: 2655, 1: 495})
Resample dataset shape Counter({0: 385, 1: 385})
```

Churn Iran

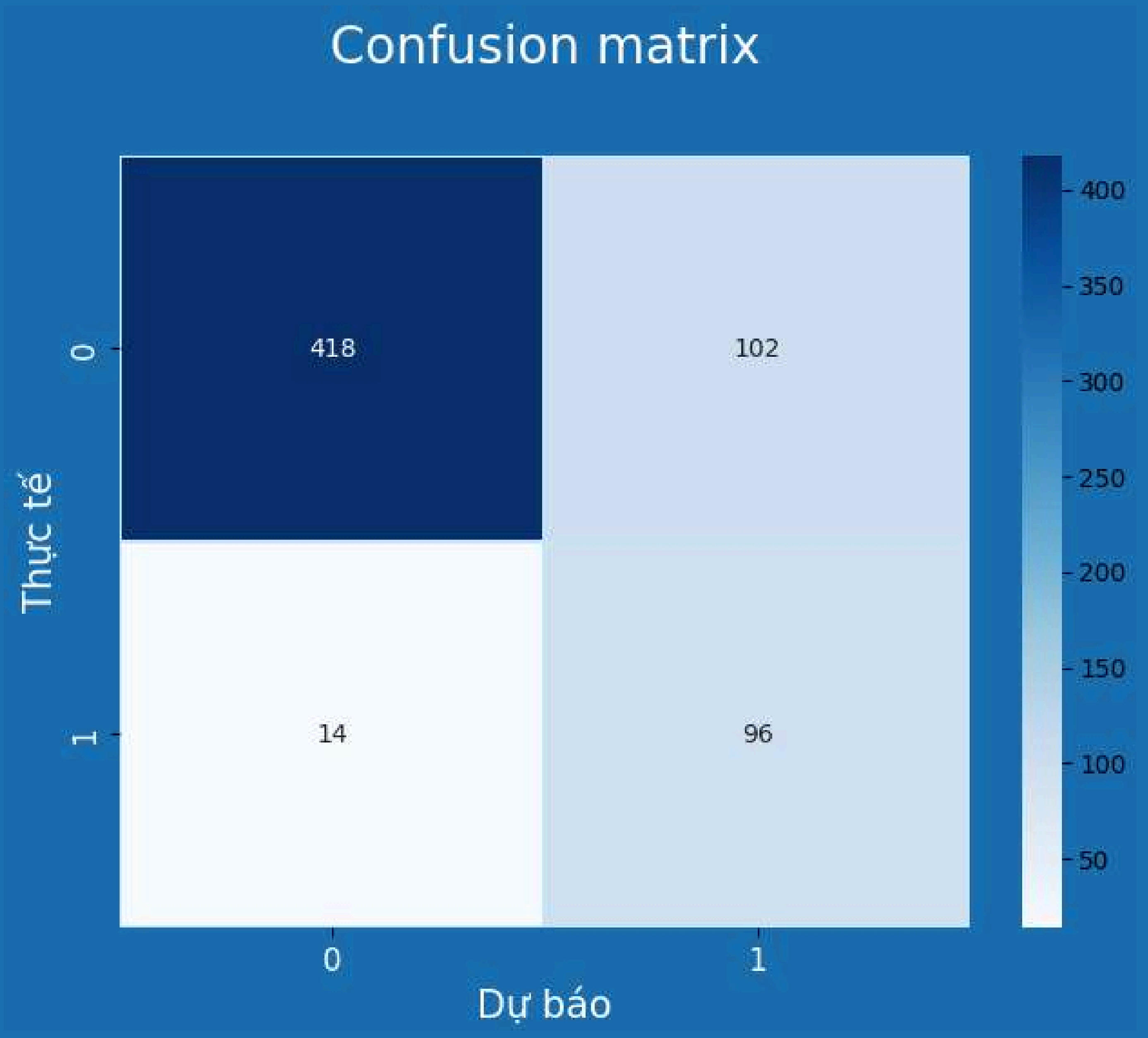
05-06-2024

MODELING

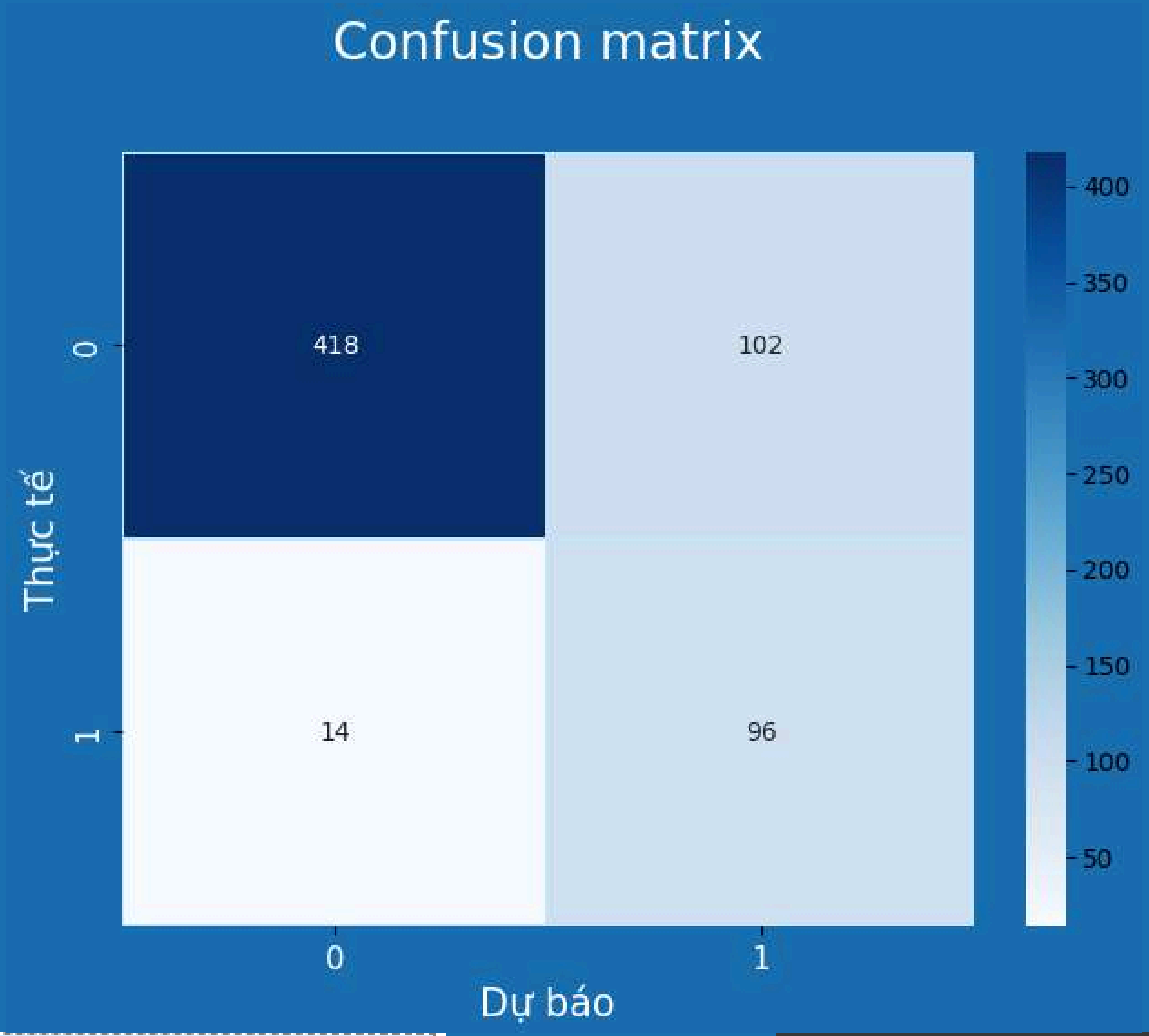
LOGISTIC REGRESSION

```
GridSearchCV
GridSearchCV(cv=5, estimator=LogisticRegression(),
             param_grid={'C': [0.01, 0.1, 1, 10, 100], 'penalty': ['l2'],
                         'solver': ['liblinear', 'saga', 'lbfgs']},
             scoring='accuracy')
estimator: LogisticRegression
LogisticRegression()
LogisticRegression
LogisticRegression()
```

	precision	recall	f1-score	support
0	0.97	0.80	0.88	520
1	0.48	0.87	0.62	110
accuracy			0.82	630
macro avg	0.73	0.84	0.75	630
weighted avg	0.88	0.82	0.83	630

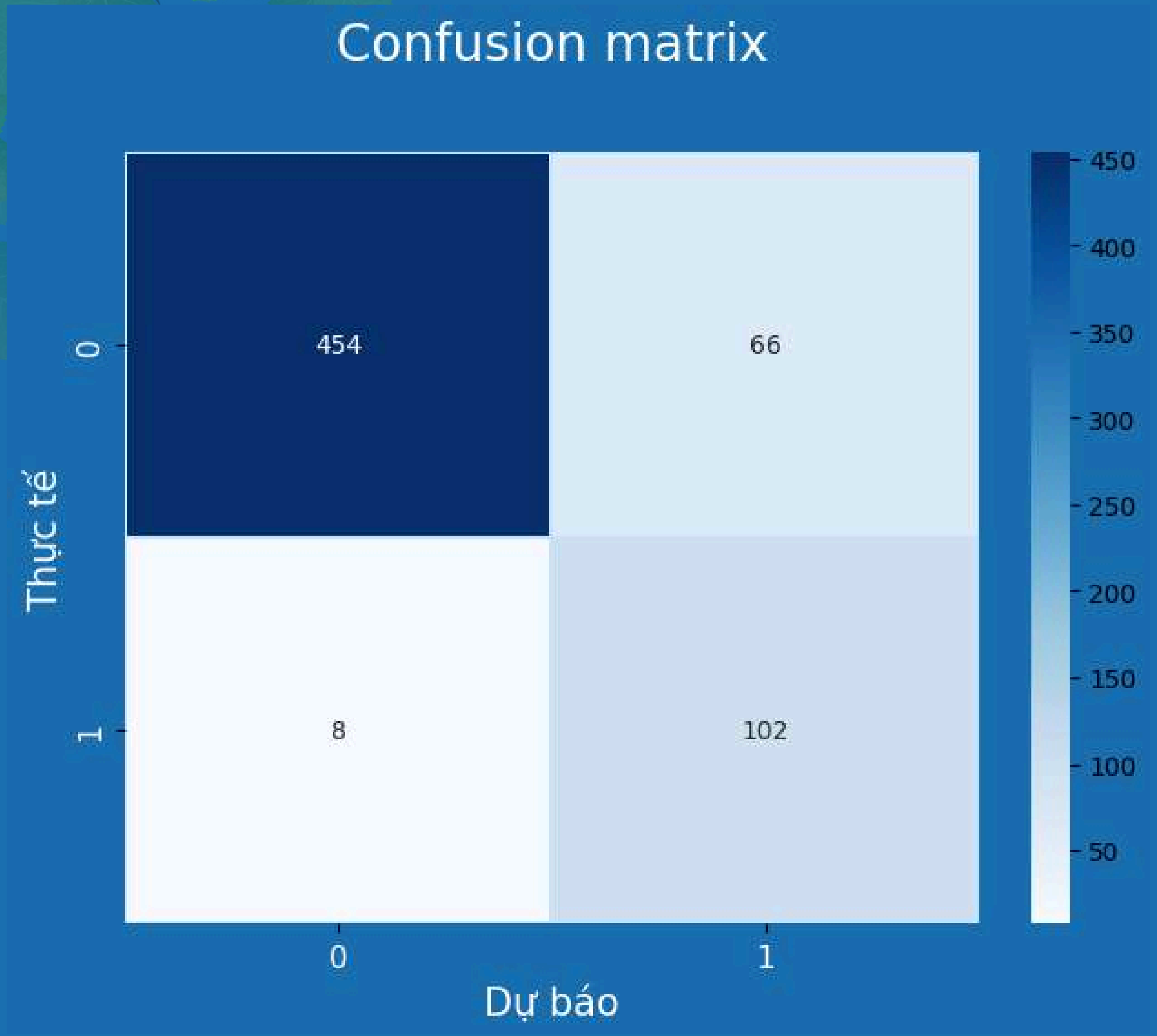


DECISION TREE




```
GridSearchCV
GridSearchCV(cv=5, estimator=DecisionTreeClassifier(),
             param_grid={'max_depth': [2, 4, 8], 'min_samples_leaf': [1, 4, 8],
                         'min_samples_split': [2, 5, 10]})
estimator: DecisionTreeClassifier
DecisionTreeClassifier()
DecisionTreeClassifier
DecisionTreeClassifier()
```

	precision	recall	f1-score	support
0	0.97	0.80	0.88	520
1	0.48	0.87	0.62	110
accuracy			0.82	630
macro avg	0.73	0.84	0.75	630
weighted avg	0.88	0.82	0.83	630



	precision	recall	f1-score	support
0	0.98	0.87	0.92	520
1	0.61	0.93	0.73	110
accuracy			0.88	630
macro avg	0.79	0.90	0.83	630
weighted avg	0.92	0.88	0.89	630

RANDOM FOREST

```
GridSearchCV
GridSearchCV(cv=5, estimator=RandomForestClassifier(),
             param_grid={'bootstrap': [True], 'max_depth': [5, 10],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2],
                         'min_samples_split': [2, 5],
                         'n_estimators': [50, 100]},
             scoring='accuracy')
estimator: RandomForestClassifier
RandomForestClassifier()
RandomForestClassifier
RandomForestClassifier()
```

KNEIGHBORS - KNN

GridSearchCV

```
GridSearchCV(cv=5, estimator=KNeighborsClassifier(),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'n_neighbors': [3, 5, 7, 9, 11], 'p': [1, 2],
                         'weights': ['uniform', 'distance']},
             scoring='accuracy')
```

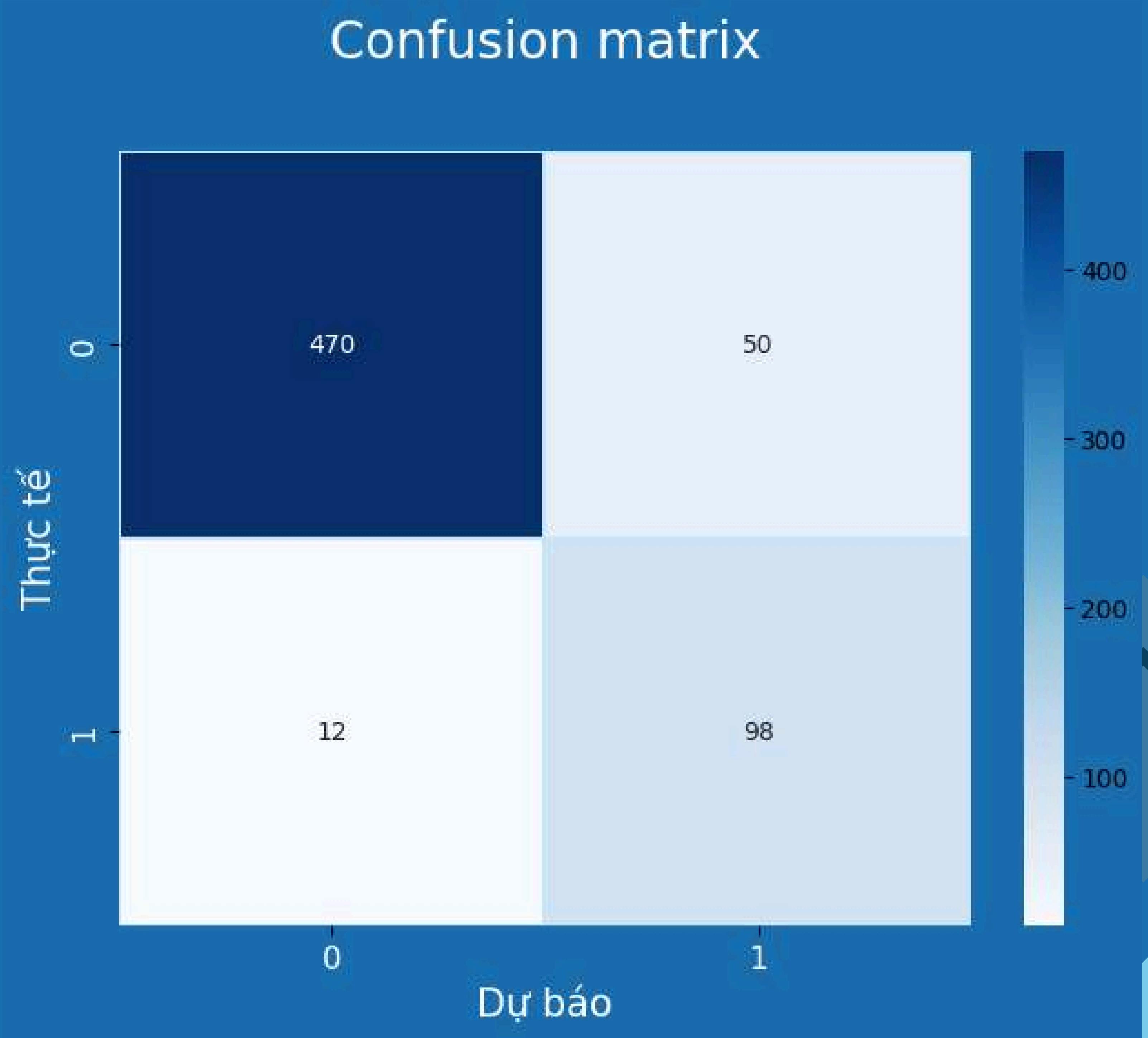
estimator: KNeighborsClassifier

KNeighborsClassifier()

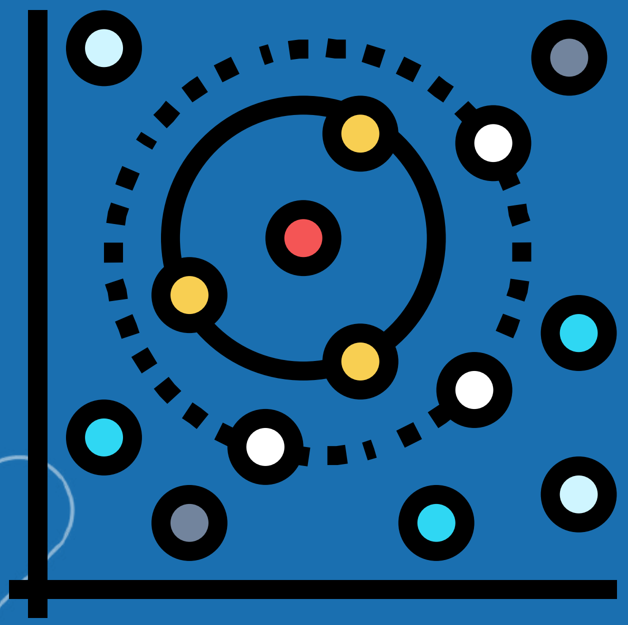
KNeighborsClassifier

KNeighborsClassifier()

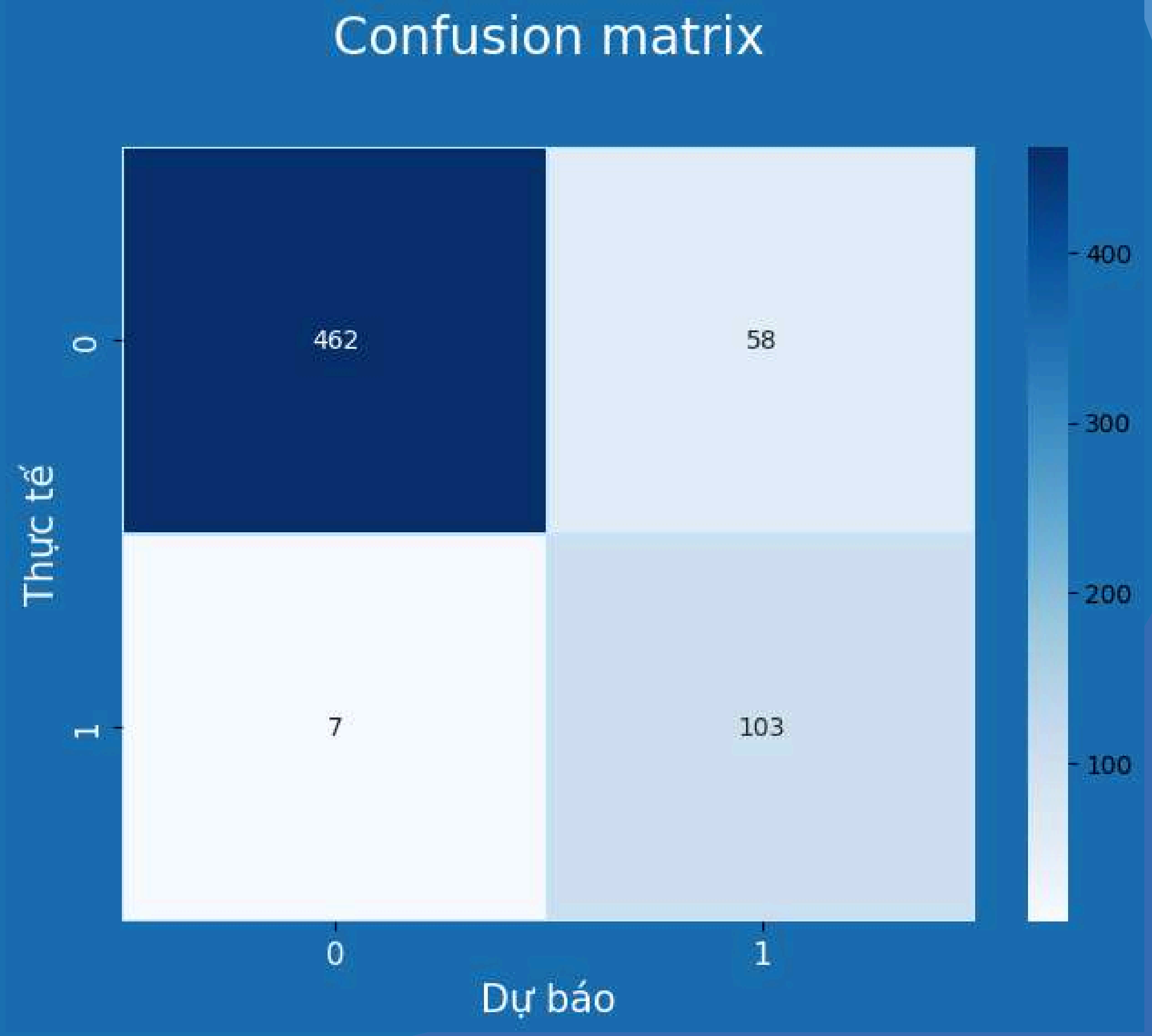
	precision	recall	f1-score	support
0	0.98	0.90	0.94	520
1	0.66	0.89	0.76	110
accuracy			0.90	630
macro avg	0.82	0.90	0.85	630
weighted avg	0.92	0.90	0.91	630



SVC MODEL



	precision	recall	f1-score	support
0	0.99	0.89	0.93	520
1	0.64	0.94	0.76	110
accuracy			0.90	630
macro avg	0.81	0.91	0.85	630
weighted avg	0.92	0.90	0.90	630



GridSearchCV

```
GridSearchCV(cv=5, estimator=SVC(),
             param_grid={'C': [1, 10], 'gamma': ['scale', 'auto'],
                         'kernel': ['linear', 'rbf']},
             scoring='accuracy')
```

estimator: SVC

SVC()

SVC

SVC()

THE BEST CHOICE

Based on the provided classification reports, the Support Vector Classifier (SVC) model is the most suitable choice for minimizing customer churn. It has the highest recall (0.94) for class 1 (churn), meaning it is the best model at correctly identifying customers who are likely to churn.

While Random Forest also has a high recall (0.93), SVC has a slightly higher precision (0.64 vs. 0.61), suggesting that it makes fewer false positives (predicting customers will churn when they do not).

Although K-Nearest Neighbors has a similar recall to Random Forest, its precision is lower, making SVC the preferred choice.

Logistic Regression and Decision Tree have the lowest recall values, making them less suitable for this specific goal of minimizing churn.

	Recall (Class 1)	Precision (Class 1)	F1-Score (Class 1)
Support Vector Classifier	0.94	0.64	0.76
Random Forest	0.93	0.61	0.73
K-Neighbors	0.89	0.66	0.76
Logistic Regression	0.87	0.48	0.62
Decision Tree	0.87	0.48	0.62

IN SUMMARY, CONSIDERING THE IMPORTANCE OF RECALL IN IDENTIFYING POTENTIAL CHURNERS, THE SVC IS THE RECOMMENDED MODEL AMONG THE FIVE OPTIONS.

Churn Iran

05-06-2024

THANKS FOR WATCHING