ANALYTICSNETWORK x BBS MACHINE LEARNING

Agenda

1 Business Understanding

2 Data Understanding

3 Data Preparation

4 Model Training

5 Model Evaluation

6 Model Development

Data Understanding | Datasets

Dataset

BANK_CLIENT

Dataset containing information of client registry.

```
==================================================
 Dataset: Clients
 Shape: 14241 rows, 5 columns

 Structure:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14241 entries, 0 to 14240
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   CUSTOMER_ID        14241 non-null  object
 1   REGISTRATION_DATE  14241 non-null  object
 2   DATE_OF_BIRTH      14241 non-null  object
 3   REGION             14241 non-null  object
 4   GENDER             14241 non-null  object
dtypes: object(5)
```

Summary Statistics:

	CUSTOMER_ID	REGISTRATION_DATE	DATE_OF_BI
count	14241	14241	14
unique	14241	5724	8
top	BI01129496	2017-07-11	1985-0
freq	1	14	

Dataset

BANK_PRODUCTS

Dataset containing information on active products and services at client month-level.

```
==================================================
 Dataset: Products
 Shape: 180576 rows, 20 columns

 Structure:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180576 entries, 0 to 180575
Data columns (total 20 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   CUSTOMER_ID       180576 non-null  object
 1   MONTH             180576 non-null  float64
 2   N_ACCT            180576 non-null  int64
 3   N_MNG_ACCT        180576 non-null  int64
 4   N_SAVING_DEPOSIT  180576 non-null  int64
 5   N_BONDS           180576 non-null  int64
 6   N_SHARES          180576 non-null  int64
 7   N_FUNDS           180576 non-null  int64
 8   N_LIFE_INSURANCE  180576 non-null  int64
 9   N_MORTGAGE        180576 non-null  int64
 10  N_FINANCING       180576 non-null  int64
 11  N_WATER           180576 non-null  int64
...
```

Summary Statistics:

	MONTH	N_ACCT	N_MNG_ACCT	N_
count	180576.000000	180576.000000	180576.000000	
mean	201829.576793	1.127835	1.032341	
std	40.040738	0.528375	0.588830	
min	201803.000000	0.000000	0.000000	
25%	201806.000000	1.000000	1.000000	
50%	201809.000000	1.000000	1.000000	
75%	201812.000000	1.000000	1.000000	

Dataset

BANK_WEALTH

Dataset containing information wealth management at client-month level

```
==================================================
 Dataset: Wealth
 Shape: 180576 rows, 6 columns

 Structure:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180576 entries, 0 to 180575
Data columns (total 6 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   CUSTOMER_ID              180576 non-null  object
 1   MONTH                    180576 non-null  int64
 2   TOT_DIRECT_DEPOSIT       180576 non-null  float64
 3   TOT_ASSETS_UNDER_CUSTODY 180576 non-null  float64
 4   TOT_MNG_DEPOSIT          180576 non-null  float64
 5   TOT_INVESTMENTS          180576 non-null  float64
dtypes: float64(4), int64(1), object(1)
```

Summary Statistics:

	MONTH	TOT_DIRECT_DEPOSIT	TOT_ASSETS_U
count	180576.000000	1.805760e+05	
mean	201829.576793	3.318563e+04	
std	40.040738	9.120417e+04	
min	201803.000000	0.000000e+00	
25%	201806.000000	1.479000e+03	
50%	201809.000000	8.401000e+03	
75%	201812.000000	3.430225e+04	
max	201903.000000	5.515419e+06	

Dataset

BANK_TRANSFER

Dataset containing information on transactions at client-month level.

```
==================================================
 Dataset: Transfers
 Shape: 691816 rows, 7 columns

 Structure:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 691816 entries, 0 to 691815
Data columns (total 7 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   MONTH              691816 non-null  float64
 1   TRANSFER_TYPE_DES  352494 non-null  object
 2   CUSTOMER_ID        691816 non-null  object
 3   AM_CREDIT          691816 non-null  float64
 4   AM_DEBIT           691816 non-null  float64
 5   NUM_CREDIT         691816 non-null  float64
 6   NUM_DEBIT          691816 non-null  float64
dtypes: float64(5), object(2)
```

Summary Statistics:

	MONTH	AM_CREDIT	AM_DEBIT	NUM_CRED
count	691816.000000	6.918160e+05	6.918160e+05	691816.000
mean	201829.357200	5.710315e+02	5.541438e+02	0.2352
std	39.678366	9.914219e+03	8.149430e+03	0.7911
min	201803.000000	0.000000e+00	0.000000e+00	0.0000
25%	201806.000000	0.000000e+00	0.000000e+00	0.0000
50%	201809.000000	0.000000e+00	0.000000e+00	0.0000
75%	201812.000000	0.000000e+00	1.300000e+02	0.0000
max	201903.000000	4.289092e+06	1.451300e+06	80.0000

Data Understanding | Data Metrics

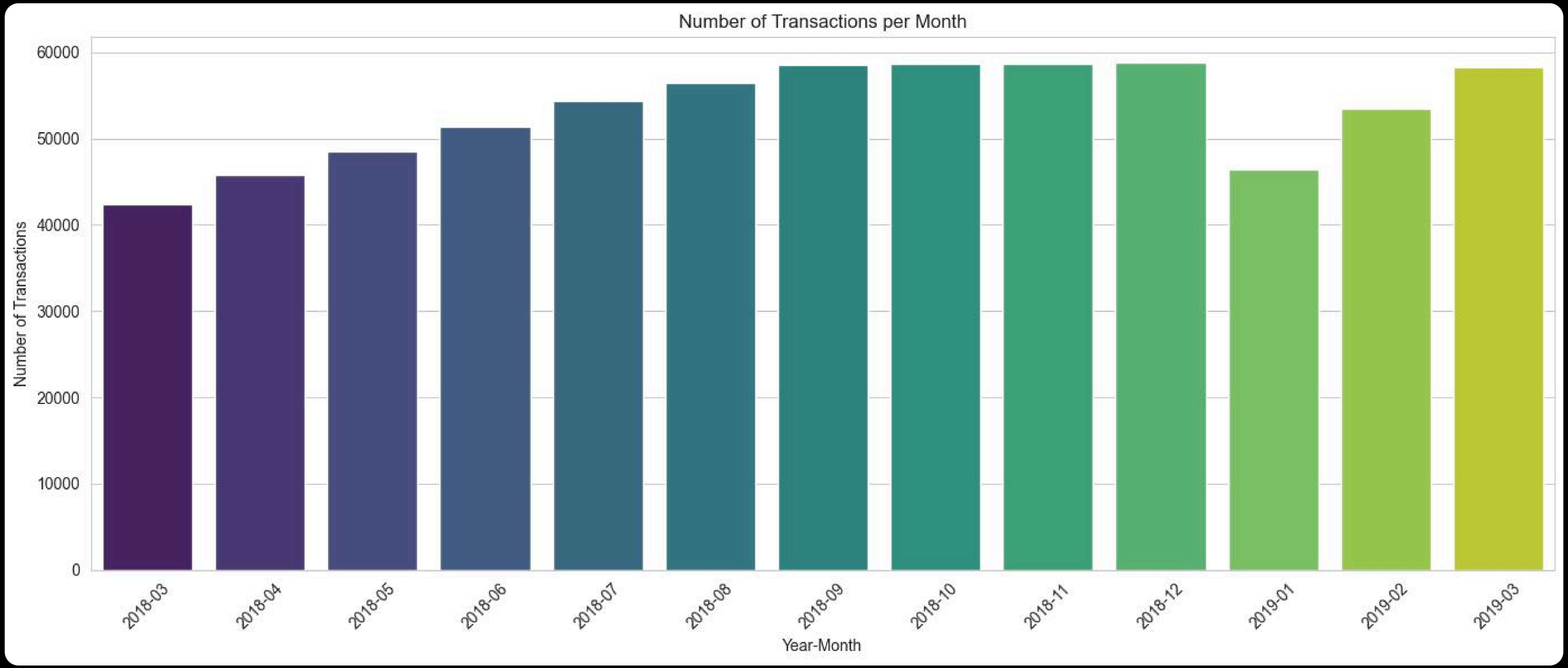
Average Transaction Value (ATV)

The **Average Transaction Value (ATV)** is the total amount of debit spending divided by the total number of debit transactions. It reflects how much, on average, a customer spends per transaction.

ATV = Total Debit Amount / Number of Debit Transactions

Steps to Calculate ATV:
1. Convert the MONTH column to a proper datetime format (YYYY-MM)
2. Filter out transactions with AM_DEBIT > 0 and NUM_DEBIT > 0
3. Group by CUSTOMER_ID and YEAR_MONTH
4. Compute: ATV = AM_DEBIT / NUM_DEBIT
5. Plot the distribution of transactions across time

Role in Propensity Scoring
- 🛍️ **High ATV** → Likely to adopt premium products, loans, or credit cards
- 💸 **Low but frequent ATV** → Good fit for daily banking features
- 🔄 **Fluctuations in ATV** → May signal income changes or lifestyle shifts



	CUSTOMER_ID	YEAR_MONTH	AM_DEBIT	NUM_DEBIT	ATV
0	AA00309049	2019-01	761.51	11.0	69.228182
1	AA00309049	2019-02	422.37	3.0	140.790000
2	AA00309049	2019-03	691.94	6.0	115.323333
3	AA00313547	2019-01	5441.45	35.0	155.470000
4	AA00313547	2019-02	7510.16	44.0	170.685455

Data Understanding | Data Metrics

ANALYTICSNETWORK x BBS MACHINE LEARNING

Income Stability Index (ISI)

The **Income Stability Index (ISI)** measures the **consistency of a customer's income** over the last 12 months, based on direct deposit data (e.g., salaries, pensions). It reflects not just how much someone earns — but **how regularly** they earn it.

ISI Calculation Steps:

1. Select the last 12 months of data (MONTH column)
2. Aggregate monthly direct deposits per customer
3. Filter out micro-transactions (< 10€)
4. Calculate:
   - AVG_MONTHLY_INCOME
   - INCOME_STD_DEV (Volatility)
   - ISI = AVG / STD
5. Retain only customers with at least 10 valid months of data

ISI Value	Interpretation
> 10	🔵 Very stable income
5 – 10	🟢 Moderately stable
1 – 5	🟠 Unstable
< 1	🔴 Highly volatile income
NaN or 0	⚪ Insufficient data / inactive

	CUSTOMER_ID	AVG_MONTHLY_INCOME	INCOME_STD_DEV	MONTHS_COUNT	ISI
0	AA00263211	197.916667	126.420265	12	1.565545
1	AA00309049	4159.166667	2305.444892	12	1.804062
2	AA00313547	40754.166667	2146.878192	12	18.982990
3	AA00349160	68815.166667	7142.619151	12	9.634444
4	AA00355641	7450.083333	966.177516	12	7.710885

Data Understanding | Data Metrics

Ratio Features

Credit Debit Ratio

Measures the proportion of credited (inflow) amounts relative to debited (outflow) amounts.

Investment Deposit Ratio

Captures the customer's appetite for investment by comparing total investments to managed deposits.

Spending Activity Ratio

Represents the average number of monthly transactions (debits + credits) normalized by customer seniority in months.

Behavioral Dynamics Features

Debit Volatility

Measures how much debit amounts vary, relative to their average.

Credit Volatility

Analogous to debit volatility, this feature assesses variability in credited amounts.

Investment Dispersion

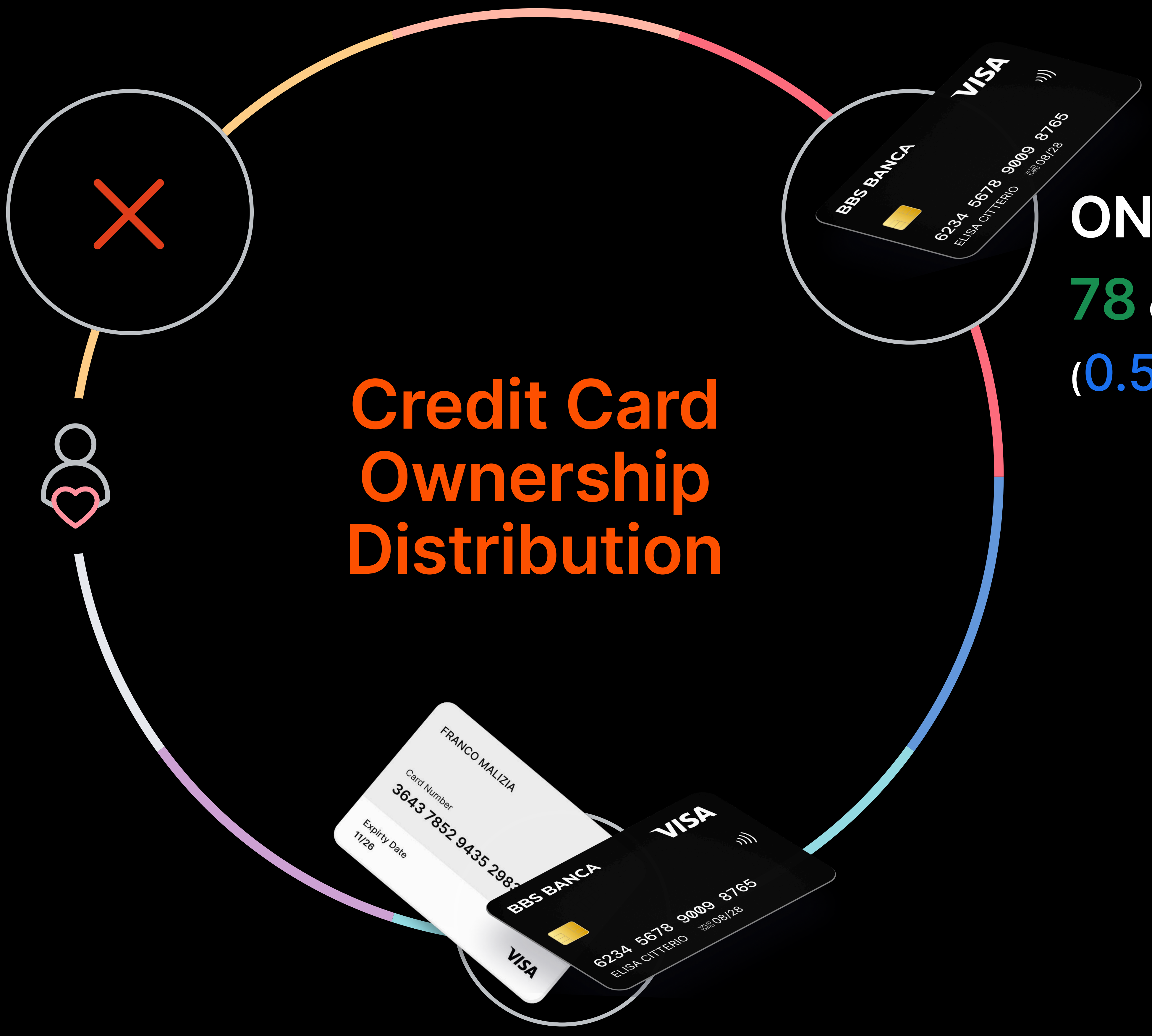
Quantifies how far the maximum investment amount deviates from the mean.

Deposit Dispersion

Captures the inconsistency in managed deposits across observed periods.

Data Understanding | Data Exploration

Target Variable Selection for Credit Card Propensity Model



ZERO CARD
9323 customers
(65.47%)

ONE CARD
78 customers
(0.55%)

MULTIPLE CARDs
4840 customers
(33.99%)

Financial Metrics by Credit Card Ownership Group

Group	Average ISI	Average Deposits	Average Assets
ZERO CARD	33.80	803.91	256,981.15
ONE CARD	2.76	395.29	37,122.64
MULTIPLE CARDS	5.13	962.65	292,516.24

Data Preparation | Missing Values and Outliers

Missing Values

```
==================================================
 No missing values in Clients dataset.

==================================================


==================================================
 No missing values in Products dataset.

==================================================


==================================================
 Missing values found in Transfers dataset:
```

	Missing Values
TRANSFER_TYPE_DES	339322

```
==================================================


==================================================
 No missing values in Wealth dataset.

==================================================
```

Outliers

Outliers are extreme values that fall outside a typical range in the dataset.
we detect them using the **Interquartile Range (IQR)** method across all numerical columns in each dataset.

How Outliers Are Identified?

- **IQR = Q3 - Q1** (Interquartile Range)
- **Lower Bound** = Q1 - 1.5 * IQR
- **Upper Bound** = Q3 + 1.5 * IQR
- Any value **outside these bounds** is considered an outlier.

What is Deviation?

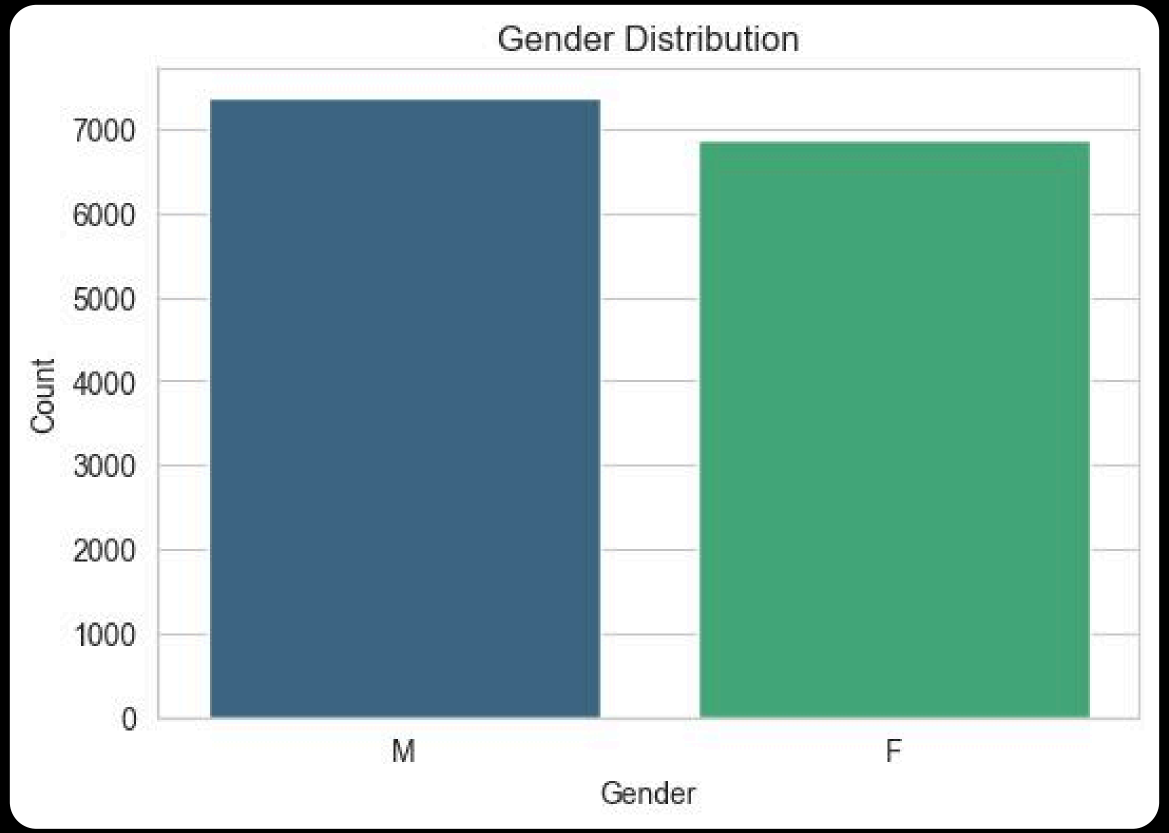
- Deviation measures **how far an outlier is from the nearest boundary**.
- If `x > upper_bound` → Deviation = `x - upper_bound`
- If `x < lower_bound` → Deviation = `lower_bound - x`
- **Example**: If the **upper bound** is 10,000 and an outlier is 15,000 → **Deviation = 5,000** (15,000 - 10,000).
- **How to interpret the average deviation?**
  - A **low average deviation** suggests that outliers are **only slightly outside the boundary**.
  - A **high average deviation** means that outliers are **far from the normal range**

```
========================================================================
 OUTLIER ANALYSIS FOR: BANK_CLIENT_clean
========================================================================
 No outliers detected in this dataset.

========================================================================
 OUTLIER ANALYSIS FOR: BANK_PRODUCTS_clean
========================================================================
                    MONTH    N_ACCT  N_MNG_ACCT  N_SAVING_DEPOSIT   N_BONDS
Total Outliers    42143.0  37878.00    49368.00          15628.00  18386.00
Avg Deviation        81.0      1.12        1.08              1.41      1.94
Max Deviation        82.0      5.00        5.00             31.00     30.50
Min Deviation        80.0      1.00        1.00              1.00      0.50

========================================================================
 OUTLIER ANALYSIS FOR: BANK_TRANSFER_clean
========================================================================
                     MONTH   AM_CREDIT    AM_DEBIT  NUM_CREDIT  NUM_DEBIT
Total Outliers   158234.00   118716.00   119436.00   118716.00  121090.00
Avg Deviation        81.07     3327.68     2745.39        1.37       3.69
Max Deviation        82.00  4289091.56  1450975.00       80.00     132.50
Min Deviation        80.00        0.01        0.03        1.00       0.50

========================================================================
...
Total Outliers    42143.0          19335.00                  36020.00
Avg Deviation        81.0          99793.28                  90504.47
Max Deviation        82.0        5431881.88                2887217.50
Min Deviation        80.0              0.88                      0.50
```

Data Preparation | Biases and Solutions

ANALYTICSNETWORK x BBS MACHINE LEARNING

Gender Bias

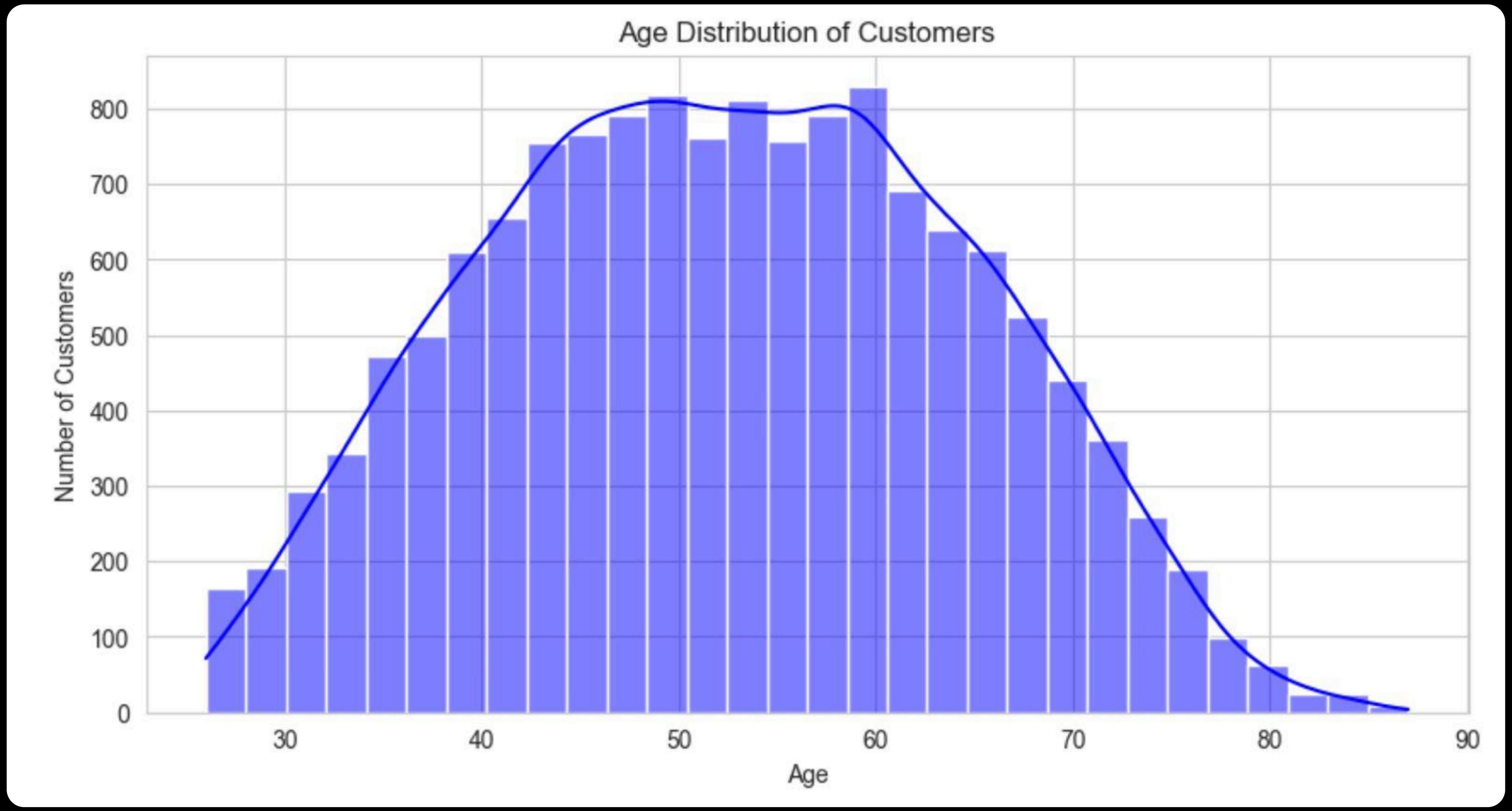


Encode as Binary

1

0

Age Distribution

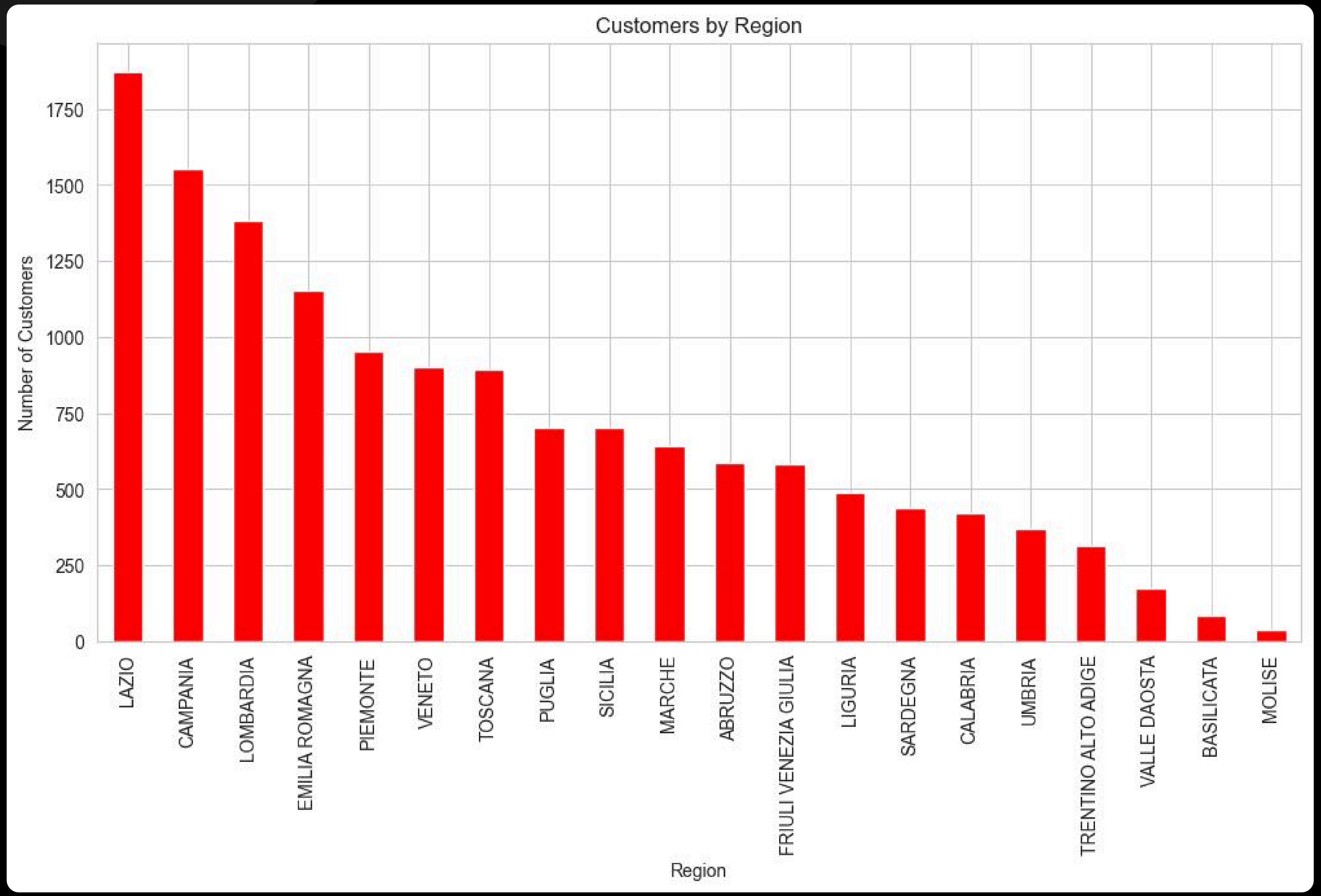


Normalization by Scaler

0.43214 | 0.23214 | 0.51216 | ... | 0.83278

Data Preparation | Bias and Solutions

Regional Distribution



Dummy Encoding

REGION_LAZIO

1

0

REGION_CAMPANIA

1

0

REGION_[...]

REGION_MOLISE

1

0

Data Preparation | Merging Data

ANALYTICSNETWORK x BBS MACHINE LEARNING

1 BANK_TRANSFER

MONTH	CUSTOMER_ID	AM_CREDIT	N_CREDIT_CARD	...
03/2018	Luca Longo	10000	6	...
04/2018	Luca Longo	20000	10	...
05/2018	Luca Longo	15000	7	...
06/2018	Luca Longo	15000	1	...
07/2018	Luca Longo	15000	1	...
08/2018	Luca Longo	45000	1	...
09/2018	Luca Longo	45000	1	...
10/2018	Luca Longo	45000	1	...

2 BANK_WEALTH

3 BANK_PRODUCTS

4 BANK_CLIENT

Merge

FINAL_DATA

CUSTOMER_ID	AGE	AM_CREDIT_SUM	AM_CREDIT_MEAN	AM_CREDIT_MAX	...
Luca Longo	24	200000	70000	40000	...
Kifah Qurmar	31	300000	60000	50000	...
Andrea Conanda	26	70000	10000	10000	...
Eleonora Porcu	24	1000000	500000	300000	...

Financial Features

_SUM

_MEAN

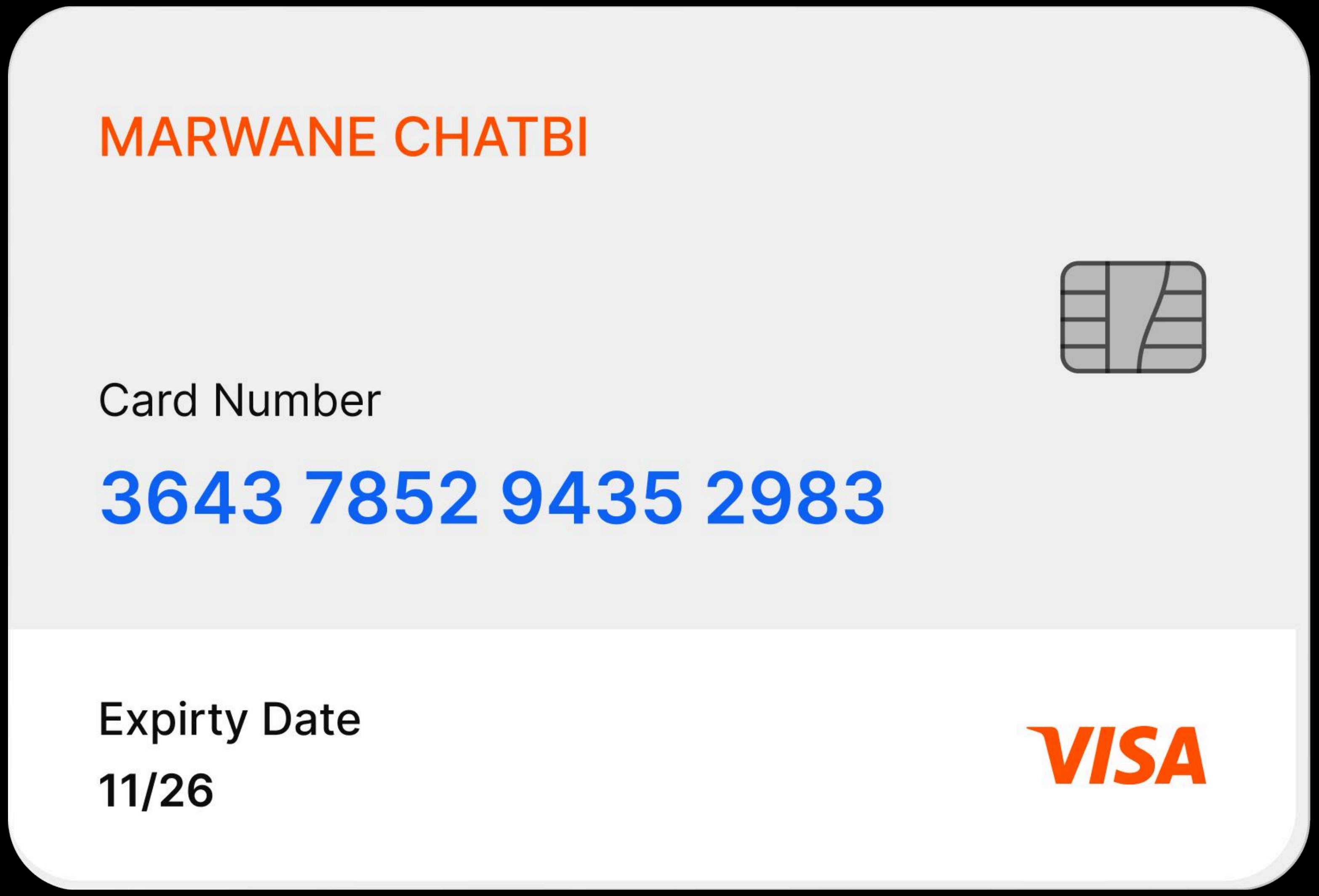
_MAX

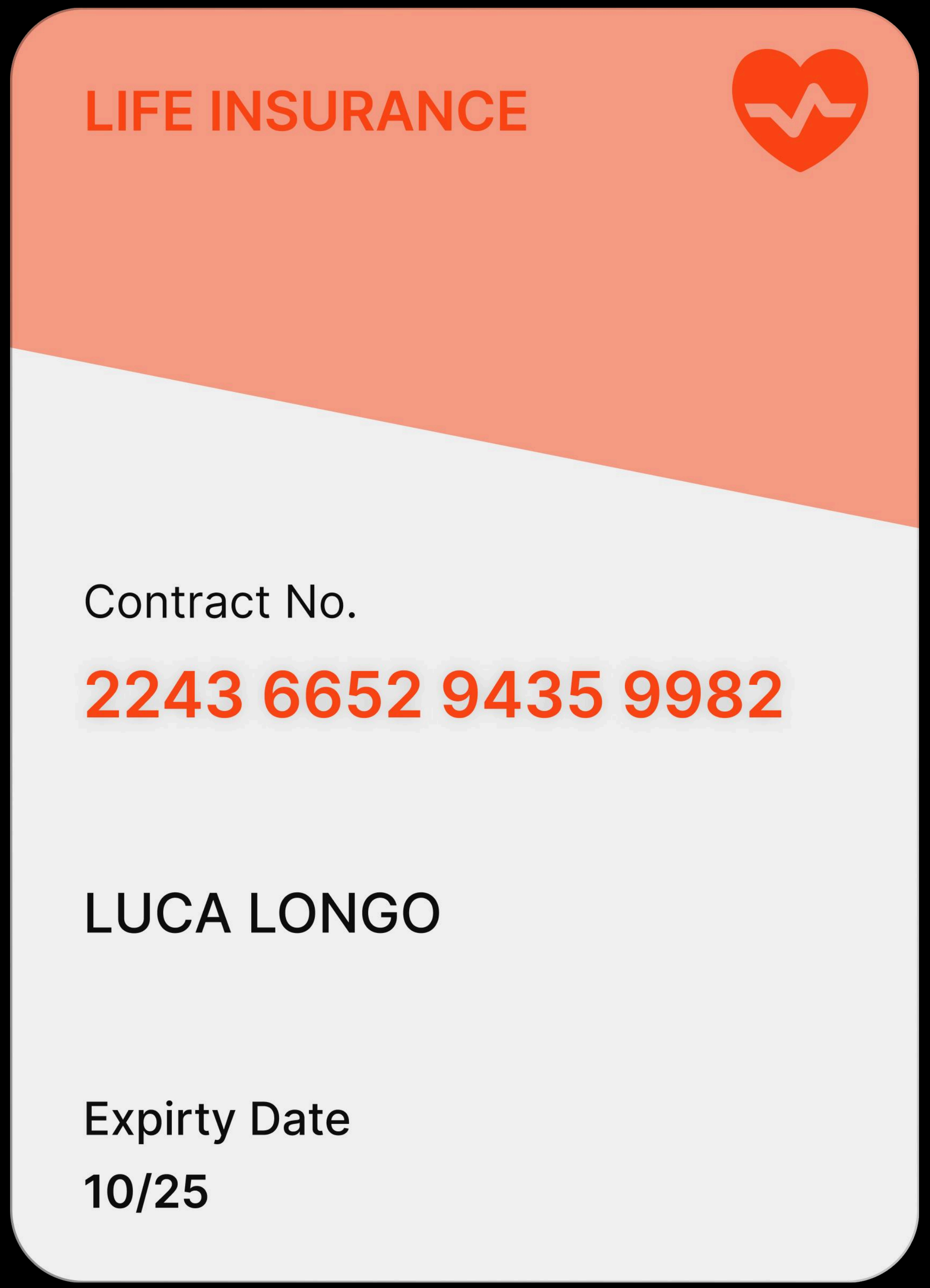
_MIN

Data Preparation | Target Binarization

Target Credit Card

N_CREDIT_CARD_SUM		TARGET_CREDIT_CARD
≥ 1	→	1
< 1	→	0





Target Life Insurance

N_LIFE_INSUARANCE_SUM		TARGET_LIFE_INSUARANCE
≥ 1	→	1
< 1	→	0

Data Preparation | Data Separation

ANALYTICSNETWORK x BBS MACHINE LEARNING

FINAL_DATA

Stratify split

CUSTOMER_ID	AGE	AM_CREDIT_SUM	AM_CREDIT_MEAN	AM_CREDIT_MAX	...
Luca Longo	24	200000	70000	40000	...
Kifah Qurmar	31	300000	60000	50000	...
Andrea Conanda	26	70000	20000	10000	...
...	...	...	...	...	...

Training Data (X_train)

CUSTOMER_ID	AGE	AM_CREDIT_SUM	...
Luca Longo	24	200000	...
Kifah Qurmar	31	300000	...
Andrea Conanda	26	70000	...
Eleonora Porcu	24	1000000	...

Training Targets (y_train)

TARGET_CREDIT_CARD	TARGET_LIFE_INSURANCE
1	1
1	1
0	1
0	0

Test Data (X_test)

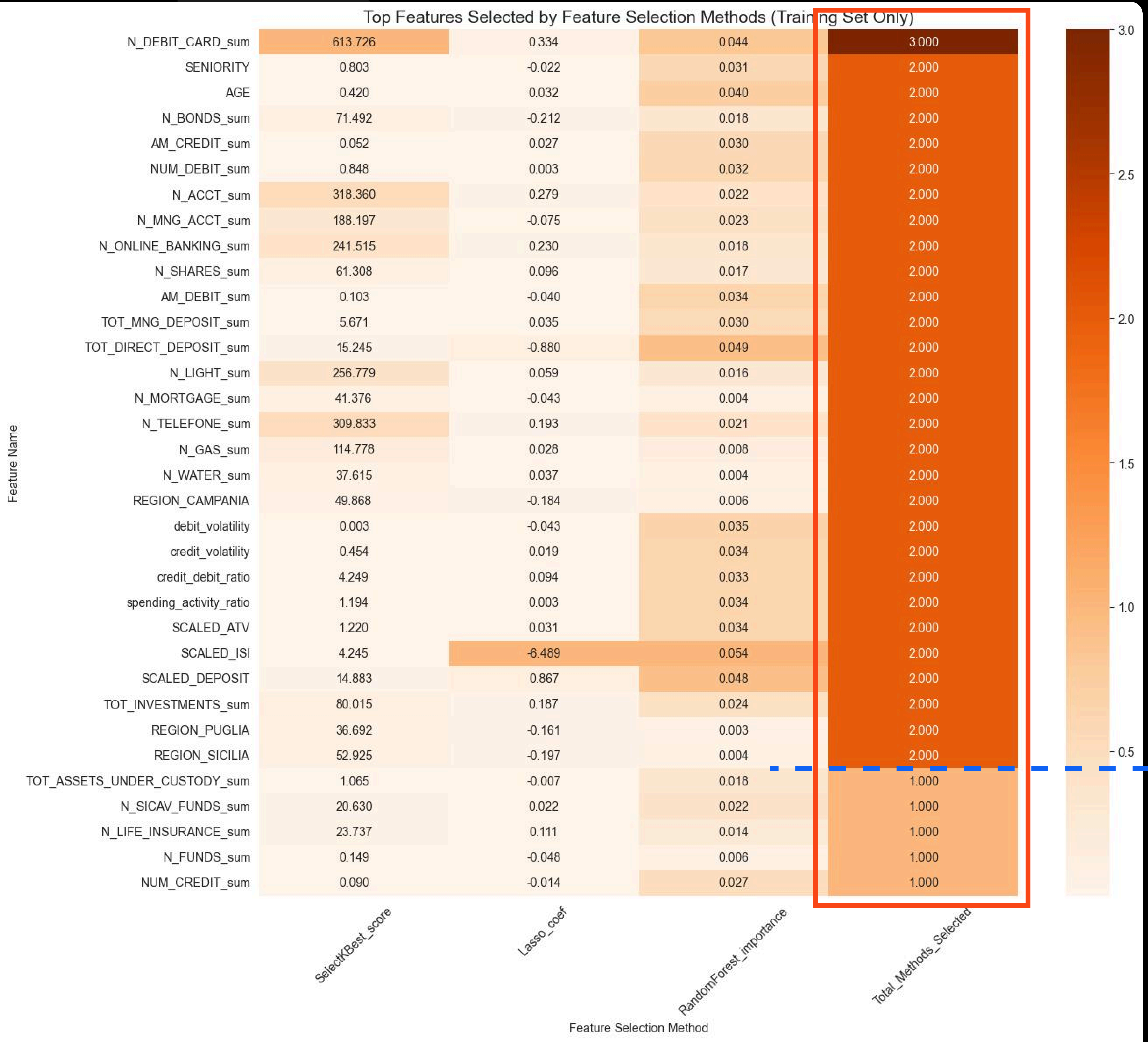
CUSTOMER_ID	AGE	AM_CREDIT_SUM	...
Elisa Citterio	22	250000	...
Franco Malizia	25	330000	...
El Alaoui Yousra	28	270000	...
Aguilar Lizbeth	26	1300000	...

Test Targets (y_test)

TARGET_CREDIT_CARD	TARGET_LIFE_INSURANCE
1	1
1	0
0	0
0	1

Data Preparation | Feature Selection

Three independent methods: SelectKBest, Lasso, Random Forest Importance



Keep

2.0

Threshold

Remove

Using all three allows us to cross-validate and analysis the most robust predictors, leading to a better understanding of feature relevance.

Each method applies its own logic and scale, so we define unified selection thresholds to standardize the comparison:

- **SelectKBest** → Features in the top 25% of F-scores
- **Lasso** → Features with non-zero model coefficients
- **Random Forest** → Features in the top 25% of importance scores

For every feature, we calculate how many of the three methods selected it. This results in a **Total_Methods_Selected** ranging from 0 (not selected by any method) to 3 (selected by all)

```
Original number of features in training set: 59
Selected features (by >= 2 methods): 29
Filtered training set shape: (9968, 29)
```

ANALYTICSNETWORK x BBS MACHINE LEARNING

Data Preparation | Data Preprocessing

Drop irrelevant columns

based on Total_Methods_Selected ≥ 2

Calculate percentage of zero for each columns

Remove sparse columns

0.90

Threshold

Remove

```
Most sparse features in the training set
REGION_PUGLIA                0.951244
REGION_SICILIA               0.950742
N_WATER_sum                  0.945927
N_ONLINE_BANKING_sum         0.910915
REGION_CAMPANIA              0.888343
N_MORTGAGE_sum               0.882123
N_GAS_sum                    0.857143
N_SHARES_sum                 0.764246
N_BONDS_sum                  0.656501
N_LIGHT_sum                  0.638844
N_TELEFONE_sum               0.637841
TOT_INVESTMENTS_sum          0.636437
TOT_MNG_DEPOSIT_sum          0.395165
N_DEBIT_CARD_sum             0.303070
credit_volatility            0.200843
AM_CREDIT_sum                0.197231
credit_debit_ratio           0.197231
debit_volatility             0.144663
AM_DEBIT_sum                 0.142556
NUM_DEBIT_sum                0.142556
SCALED_ATV                   0.142556
spending_activity_ratio      0.136738
N_MNG_ACCT_sum               0.063604
N_ACCT_sum                   0.048154
...
```

Keep

Apply SMOTE to balance the training data

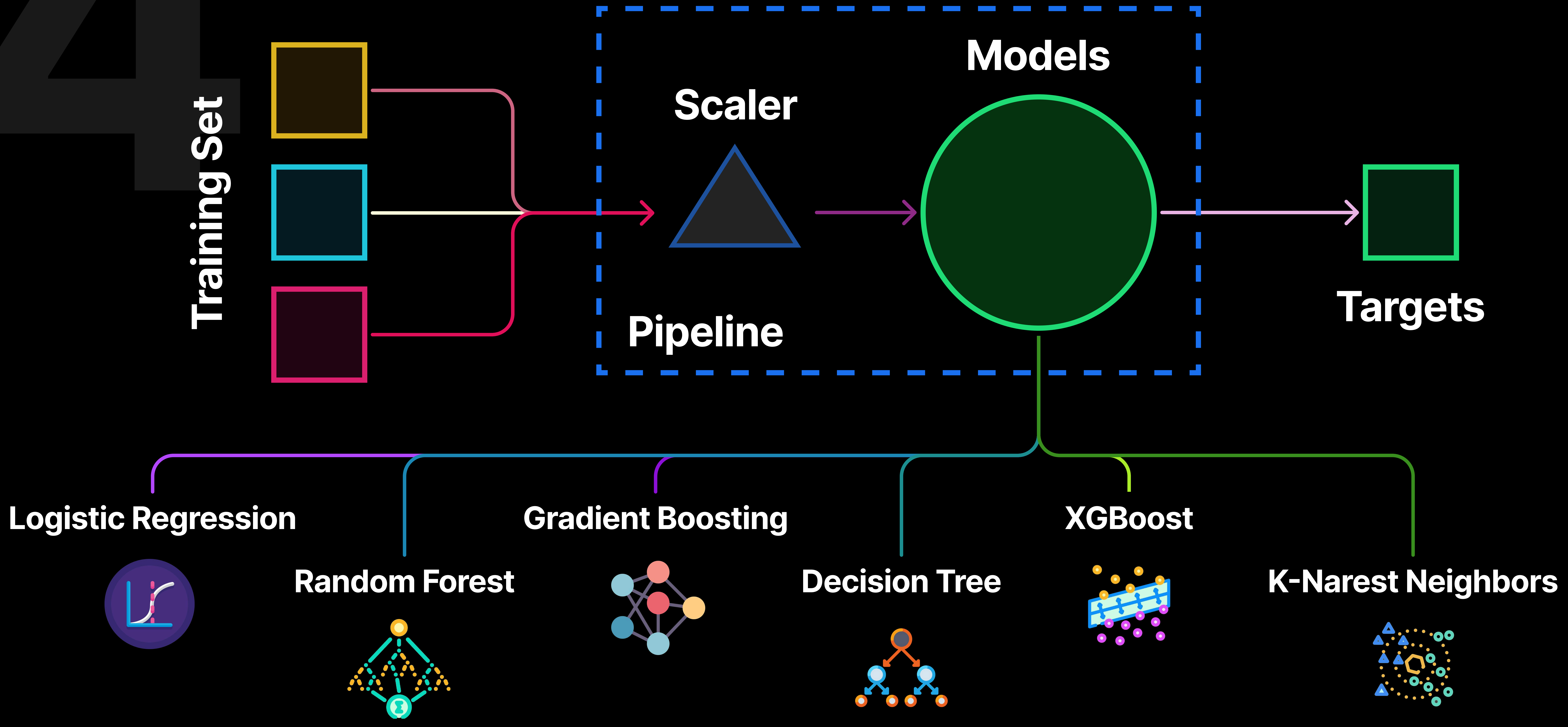
SMOTE (Synthetic Minority Over-sampling Technique) addresses class imbalance by generating synthetic samples for the minority class instead of simply duplicating existing ones.

SMOTE imbalanced learn

```
Balanced class distribution
TARGET_CREDIT_CARD
0    6526
1    6526
Name: count, dtype: int64
```

Make test set consistent with training set

Modeling | Model Selection



Modeling | Model Results

SMOTE

imbalanced learn

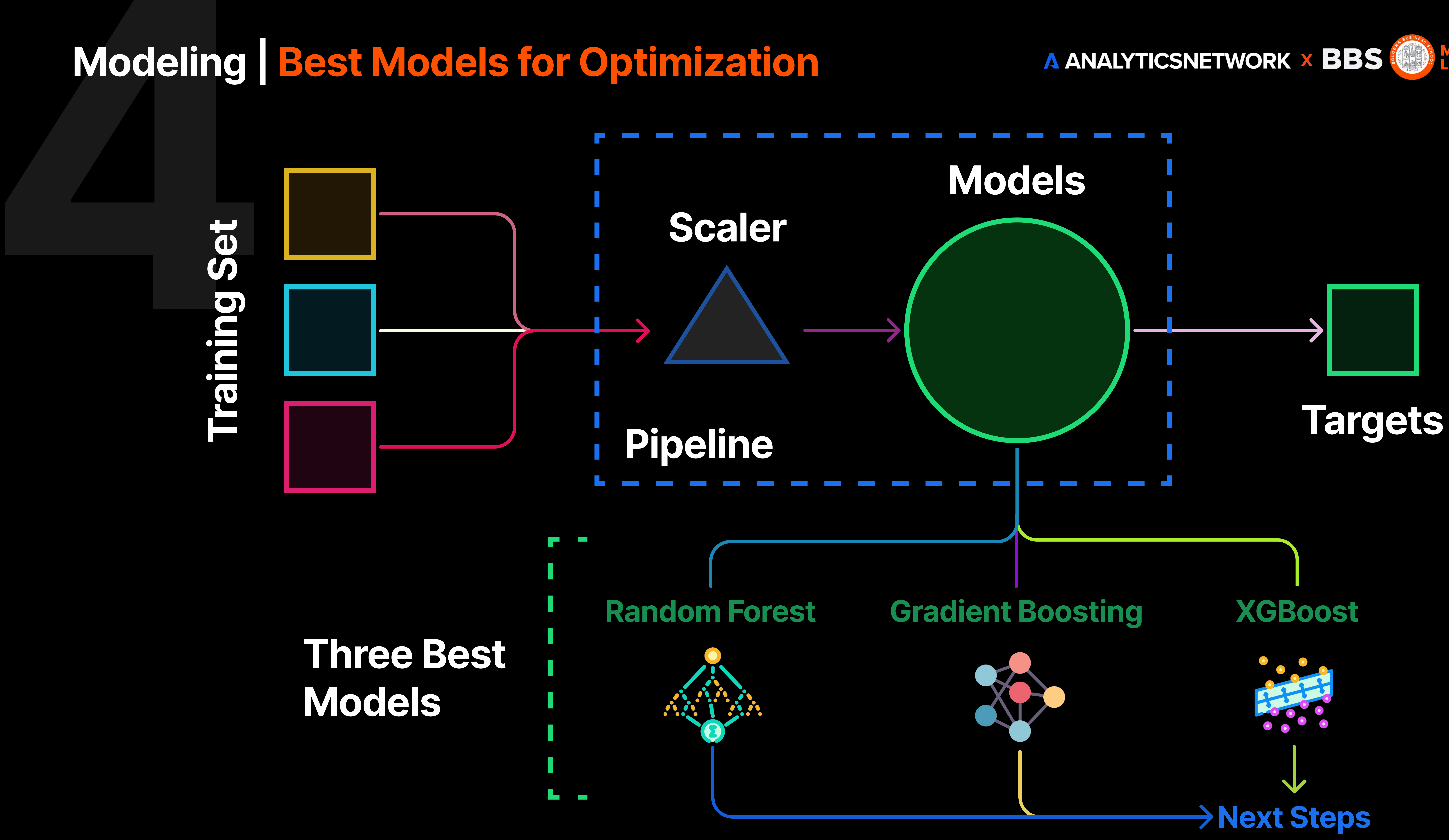
With SMOTE

No.	Model	ROC AUC	Precision	Recall	F1-Score
1	Random Forest	0.714518	0.544370	0.494580	0.518282
2	Gradient Boosting	0.704476	0.521771	0.487127	0.503854
3	Logistic Regression	0.693561	0.480131	0.597561	0.532448
4	XGBoost	0.692982	0.513848	0.477642	0.495084
5	Decision Tree	0.583102	0.440497	0.504065	0.470142

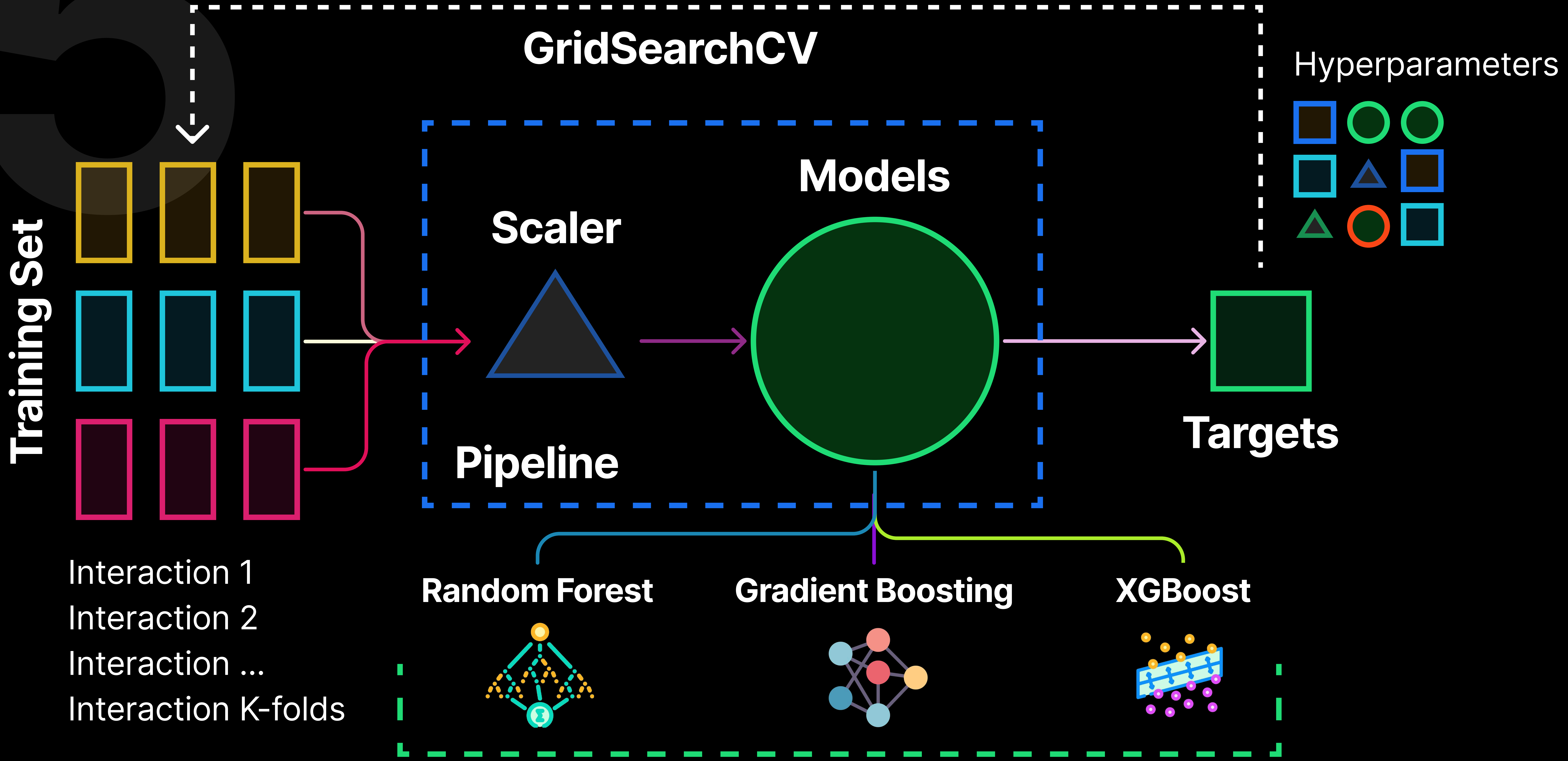
Without SMOTE

No.	Model	ROC AUC	Precision	Recall	F1-Score
1	Random Forest	0.721940	0.624818	0.289973	0.396113
2	Gradient Boosting	0.720582	0.583226	0.306233	0.401599
3	XGBoost	0.703455	0.539346	0.413279	0.467971
4	Logistic Regression	0.698826	0.485416	0.597561	0.535682
5	Decision Tree	0.579955	0.450169	0.449864	0.450017

Modeling | Best Models for Optimization



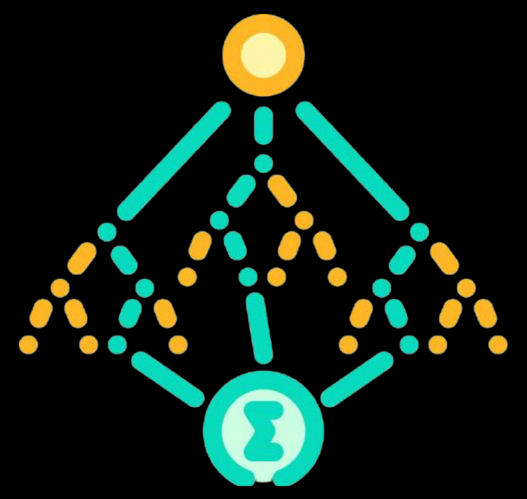
Model Evaluation | Hyperparameter Tuning



Best Hyperparameters • Best ROC AUC → Next Slides

Model Evaluation | Tuning Random Forest

Random Forest



1

Fitting 5 folds for each of 216 candidates, totalling 1080 fits

2

Best Hyperparameter

```
{'clf__max_depth': None,
 'clf__max_features': 'log2',
 'clf__min_samples_leaf': 1,
 'clf__min_samples_split': 2,
 'clf__n_estimators': 300}
```

3

Best ROC AUC (Training)

85.12%

Model Evaluation | Tuning XGBoost

XGBoost

1

Fitting 5 folds for each of 243 candidates, totalling 1215 fits

2

Best Hyperparameter

```
{'clf__colsample_bytree': 0.7,
 'clf__learning_rate': 0.1,
 'clf__max_depth': 7,
 'clf__n_estimators': 300,
 'clf__subsample': 0.8}
```

3

Best ROC AUC (Training)

83.87%

Model Evaluation | Tuning Gradient Boosting

ANALYTICSNETWORK x BBS MACHINE LEARNING

Gradient Boosting

1

Fitting 5 folds for each of 32 candidates, totalling 160 fits

2

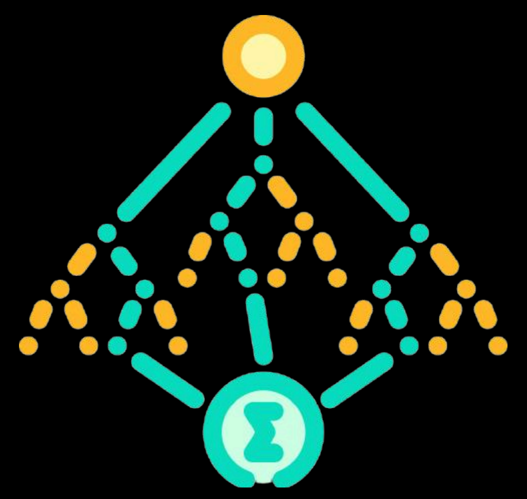
Best Hyperparameter

```
GridSearchCV

GridSearchCV(cv=5,
             estimator=Pipeline(steps=[('scaler', StandardScaler()),
                                       ('clf',
                                        GradientBoostingClassifier(random_state=42))]),
             n_jobs=-1,
             param_grid={'clf__learning_rate': [0.05, 0.1],
                         'clf__max_depth': [3, 5],
                         'clf__min_samples_split': [2, 5],
                         'clf__n_estimators': [100, 200],
                         'clf__subsample': [0.8, 1.0]},
             scoring='roc_auc', verbose=2)

best_estimator_: Pipeline

Pipeline(steps=[('scaler', StandardScaler()),
                ('clf',
                 GradientBoostingClassifier(max_depth=5, min_samples_split=5,
                                            n_estimators=200,
                                            random_state=42))])

StandardScaler

StandardScaler()

GradientBoostingClassifier

GradientBoostingClassifier(max_depth=5, min_samples_split=5, n_estimators=200,
                           random_state=42)
```

Model Evaluation | Final Evaluation

No.	Model	Best ROC AUC	Best Precision	Best Recall	Best F1-Score
1	Random Forest	0.718158	0.539910	0.485772	0.511412
2	Gradient Boosting	0.710334	0.525091	0.489160	0.506489
3	XGBoost	0.707771	0.520000	0.475610	0.496815

Random Forest



Model Evaluation | Visualization

Important Features

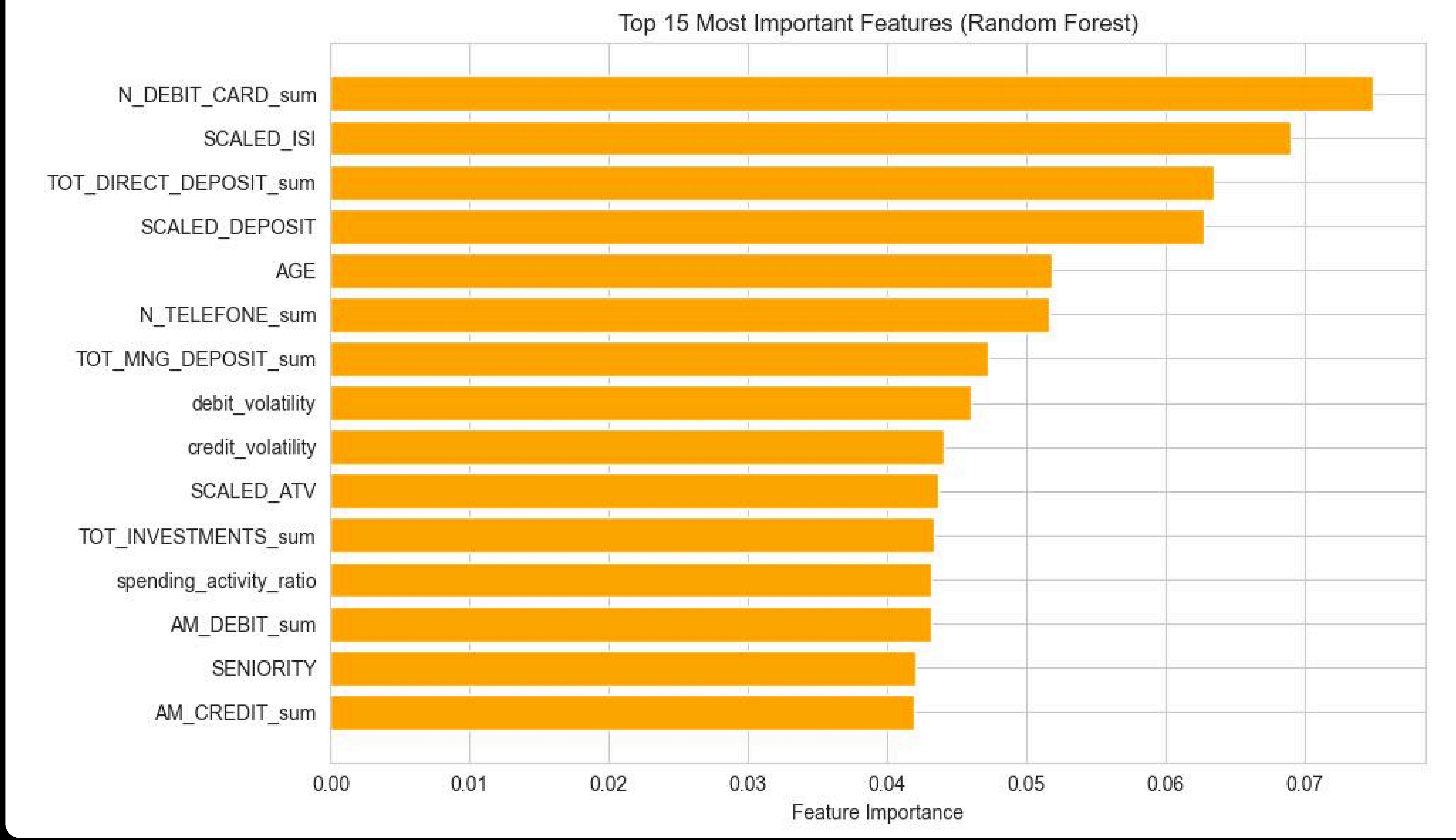
- The model is heavily influenced by behavioral and transactional patterns, such as card usage, deposits, and payment activity — more than by demographic variables alone.
- Income stability and transaction regularity (e.g., **SCALED_ISI**, **debit_volatility**) are strong signals.
- The presence of utility payment features (**N_TELEFONE_sum**) shows that non-obvious variables can carry predictive value, perhaps serving as proxies for engagement or financial habits.

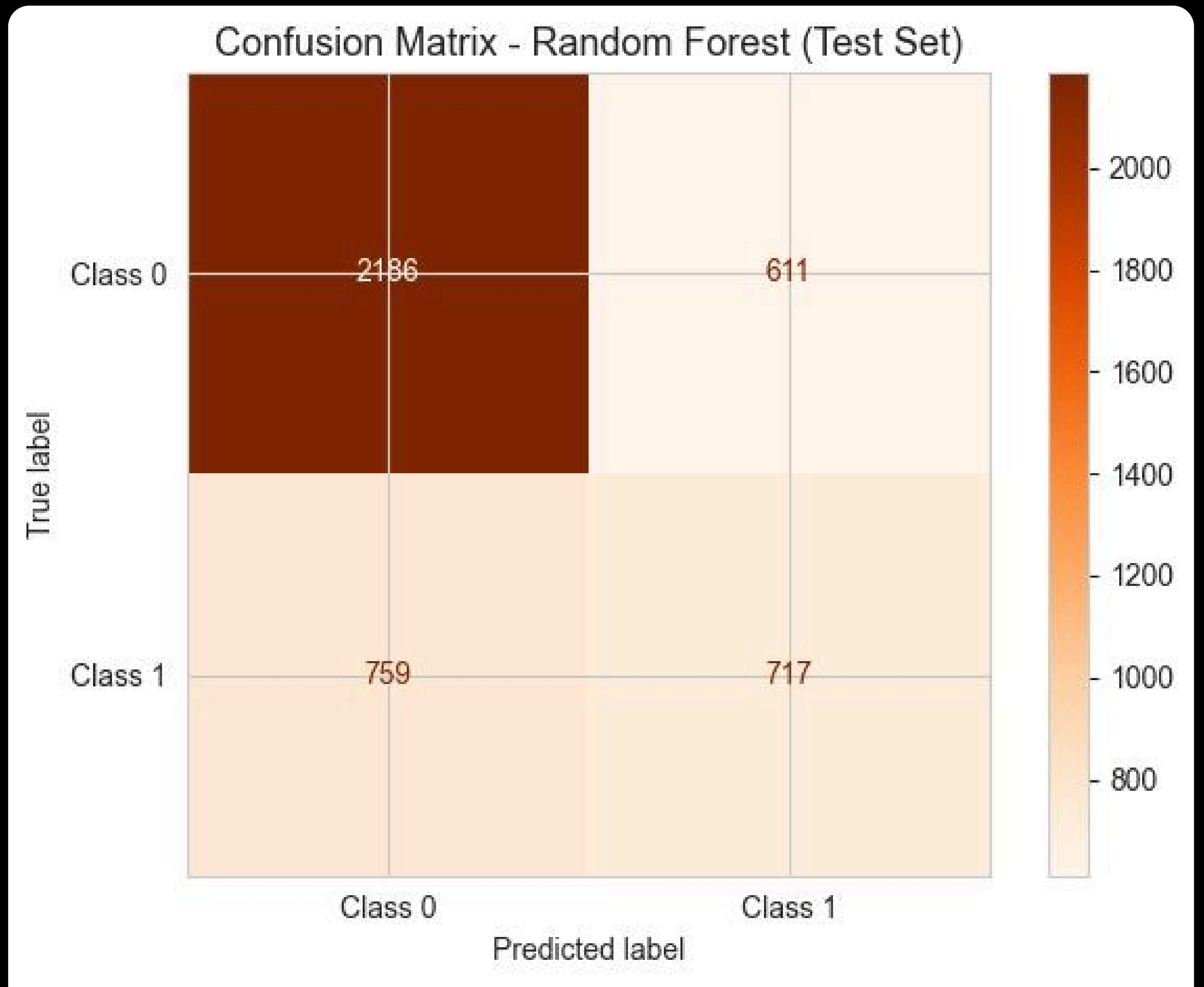
ROC Curve

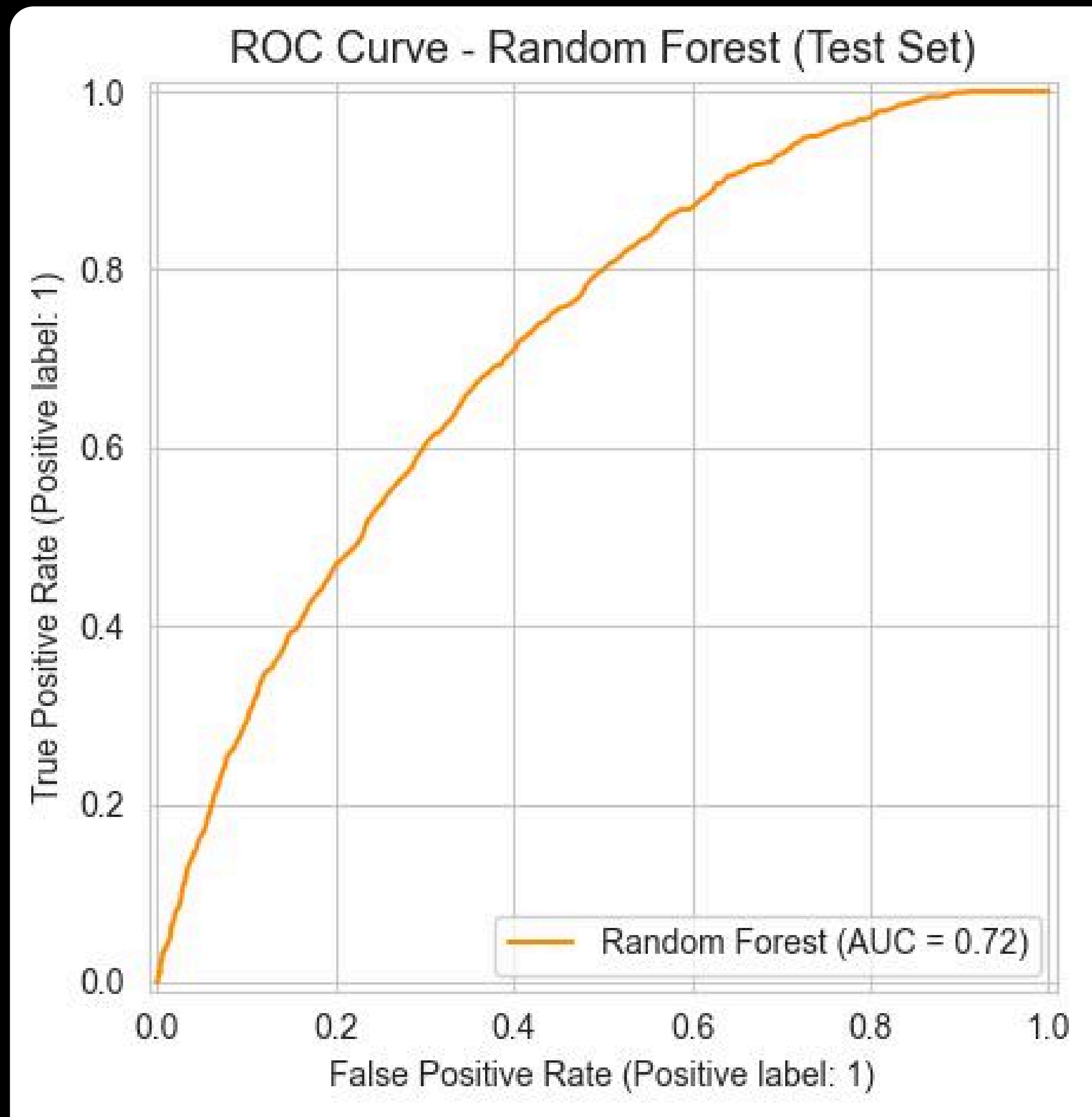
- AUC between **0.7** and **0.8** is generally considered acceptable/good in complex real-world scenarios.
- The curve shows that the model is able to retain a decent recall while not overwhelming the system with false positives — especially in the mid-threshold range.

Confusion Matrix

- The model is reasonably good at identifying non-holders (**Class 0**):
~**78**% of actual non-holders (**2186** out of **2797**) were classified correctly.
- However, it struggles more with **Class 1** (card holders):
- It misses **759** customers who actually own a credit card → this is a reca issue.
- It also makes **611** false alarms, predicting ownership where there is non impacts precision.







Model Development | Business Insights

1 Customers with high debit card usage and frequent direct deposits are strong candidates for credit cards

2 Age and financial stability indicators, like ISI score and transaction patterns, help predict credit card interest.