



PROJECT CREDIT CARD

PRESENT TO : PASD PUTTHAPIPAT

INTRODUCTION

ปัจจุบันประชาชนส่วนใหญ่มีการใช้งานบัตรเครดิตกี่เพิ่มมากขึ้น และมีการนำบัตรเครดิตมาใช้เป็นเครื่องมือหลักๆ ใน การใช้จ่ายในชีวิตประจำวัน จึงทำให้ผู้คนส่วนใหญ่มีความสะดวกรวดเร็วในการใช้จ่ายมากยิ่งขึ้น และยังมีความปลอดภัยในการใช้จ่ายมากกว่าการพกเงินสดติดตัวเป็นจำนวนมาก ซึ่งจะทำให้เสี่ยงต่อการสูญหายหรือโจรกรรมมากยิ่งขึ้น โดยสมัยนี้ร้านค้าห้างสรรพสินค้าและศูนย์การค้าต่างๆ ที่ให้บริการในหลายๆ แห่งก็ว่าประเทศ ได้มีการรับชำระเงินผ่านบัตรเครดิตกี่เพิ่มมากขึ้นผู้คนส่วนใหญ่สามารถใช้บัตรเครดิตในการชำระค่าสินค้าและค่าบริการ แทนการชำระเงินด้วยเงินสด

งานนี้เน้นการศึกษาการนำทางลูกหนี้ที่มีโอกาสในการผิดนัดชำระกับทางธนาคาร โดยการทดลองในครั้งนี้ เราทดลองกับข้อมูลการชำระบัญชีบัตรเครดิตซึ่งประกอบด้วย ข้อมูลจำนวนทั้งหมด 307,511 แถว และคอลัมน์ทั้งหมด 122 คอลัมน์ จากแหล่งข้อมูลสารสนเทศเว็บไซต์ <https://www.kaggle.com/datasets/mishra5001/credit-card?resource=download> โดยการแบ่งข้อมูลออกเป็น 2 กลุ่มใหญ่ๆ คือ กลุ่มลูกหนี้ปกติ คือกลุ่มลูกหนี้ที่ไม่ได้มีการผิดนัดชำระกับทางธนาคาร และกลุ่มลูกหนี้ที่ไปปกติ คือกลุ่มลูกหนี้ที่มีการผิดนัดชำระกับทางธนาคาร

OBJECTIVE

01

การศึกษาการทำนายลูกหนี้ที่มีโอกาสในการพิดนัดชำระกับทางธนาคาร

02

เพื่อศึกษาว่าบุคคลที่มีความสามารถในการชำระสินเชื่อได้ จะไม่ถูกปฏิเสธ

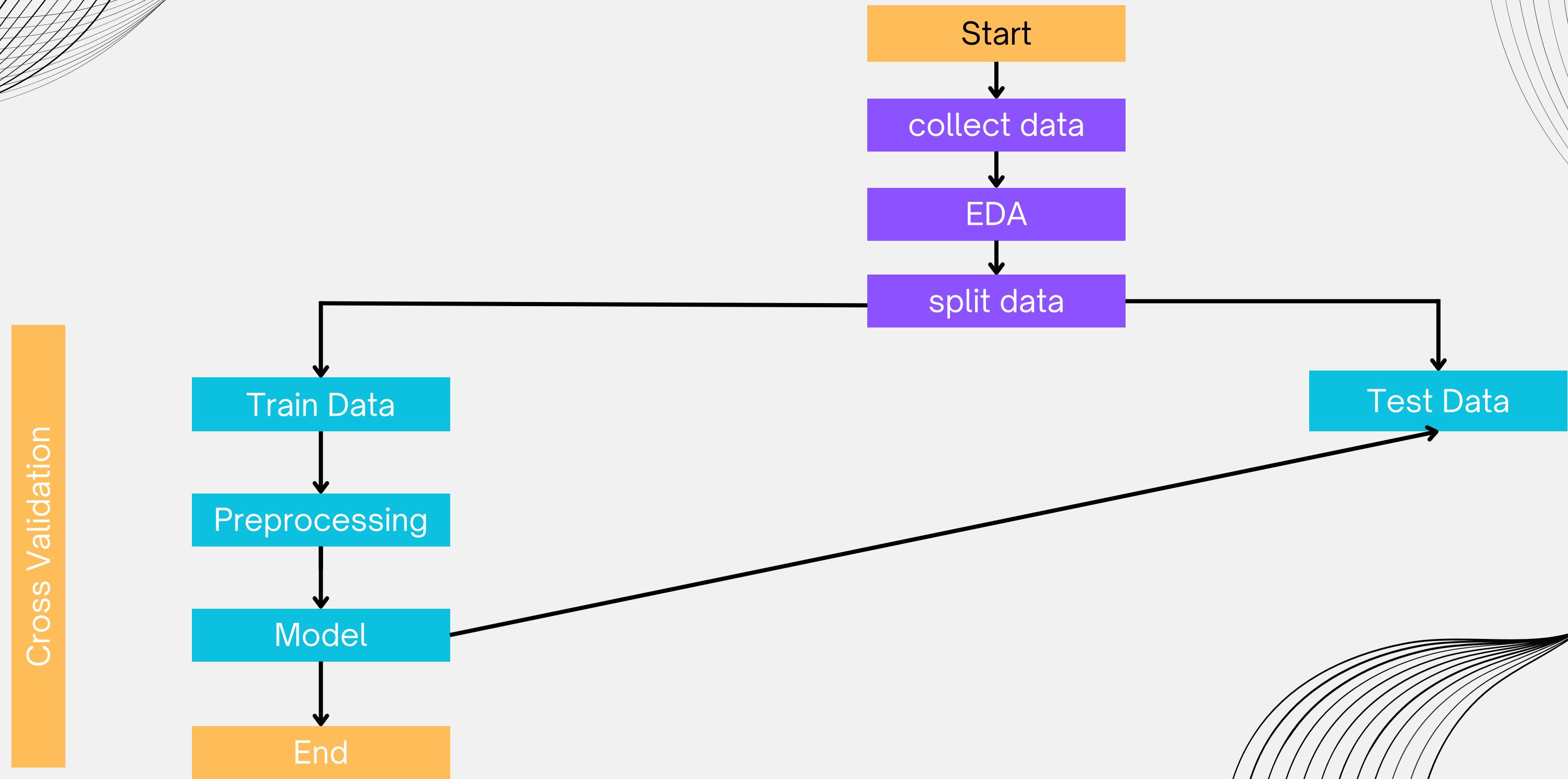
03

เพื่อวิเคราะห์ผู้สมัครที่สามารถชำระคืนเงินกู้ได้จะไม่ถูกปฏิเสธการได้สินเชื่อ
(ความสามารถในการขอสินเชื่อบัตรเครดิต)

*ตัดสินใจอนุมัติสินเชื่อตามประวัติของผู้สมัคร มีคนอยู่ 2 ประเภท

- 1) กลุ่มลูกหนี้ปกติ คือกลุ่มลูกหนี้ที่ไม่ได้มีการพิดนัดชำระกับทางธนาคาร
- 2) กลุ่มลูกหนี้ที่ไม่ปกติ ตือกลุ่มลูกหนี้ที่มีการพิดนัดชำระกับทางธนาคาร

วิธีดำเนินการ



วิธีดำเนินการ

01

การเก็บรวบรวมข้อมูล
ข้อมูลการทำธุกรรมสินเชื่อ
บัตรเครดิตซึ่งประกอบด้วย
ข้อมูลจำนวนทั้งหมด
307,511 แถว และคอลัมน์
ทั้งหมด 122 คอลัมน์
จากแหล่งข้อมูลสารสนเทศ

02

การสำรวจข้อมูล
การทำ exploratory
Data Analysis (EDA)
สำรวจข้อมูล เช่น ดูค่าทาง
สถิติข้อมูล ดูจำนวนข้อมูล
ในแต่ละคอลัมน์และทำการ
ดูความสัมพันธ์ในแต่ละ
คอลัมน์

03

การเตรียมข้อมูล
การทำความสะอาดข้อมูล
 เช่น การลบคอลัมน์ที่ไม่มี
ผลต่อการทำนาย และ
การแปลงชนิดของข้อมูล
ที่ถูกเก็บอยู่ในแต่ละ
คอลัมน์

04

การทำแบบจำลองข้อมูล
การแบ่งข้อมูลออกเป็น Train
และ Test และทำ Machine
Learning Algorithms เช่น
Logistic Regression,
XGBoostClassifier,
K-nearest Neighbors,
Random Forest , Support
Vector Classifier (SVC),
Gradient Boosting

DATA

- นำเข้าข้อมูล Application Data และ previous application จากนั้นทำการสำรวจข้อมูลโดยข้อมูลที่นำเข้ามามีขนาด 166.13 MB และ 404.97 MB ตามลำดับ ซึ่งทำการวิเคราะห์ด้วย Spark และ Pandas

Application Data

SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL
100002	1	Cash loans	M	N	Y	0	202500.0
100003	0	Cash loans	F	N	N	0	270000.0
100004	0	Revolving loans	M	Y	Y	0	67500.0
100006	0	Cash loans	F	N	Y	0	135000.0
100007	0	Cash loans	M	N	Y	0	121500.0
100008	0	Cash loans	M	N	Y	0	99000.0
100009	0	Cash loans	F	Y	Y	1	171000.0
100010	0	Cash loans	M	Y	Y	0	360000.0
100011	0	Cash loans	F	N	Y	0	112500.0
100012	0	Revolving loans	M	N	Y	0	135000.0
100014	0	Cash loans	F	N	Y	1	112500.0
100015	0	Cash loans	F	N	Y	0	38419.155
100016	0	Cash loans	F	N	Y	0	67500.0
100017	0	Cash loans	M	Y	N	1	225000.0
100018	0	Cash loans	F	N	Y	0	189000.0
100019	0	Cash loans	M	Y	Y	0	157500.0
100020	0	Cash loans	M	N	N	0	108000.0
100021	0	Revolving loans	F	N	Y	1	81000.0
100022	0	Revolving loans	F	N	Y	0	112500.0
100023	0	Cash loans	F	N	Y	1	90000.0

previous application

SK_ID_PREV	SK_ID_CURR	NAME_CONTRACT_TYPE	AMT_ANNUITY	AMT_APPLICATION	AMT_CREDIT	AMT_DOWN_PAYMENT	AMT_GOODS
2030495	271877	Consumer loans	1730.43	17145.0	17145.0	0.0	1
2802425	108129	Cash loans	25188.615	607500.0	679671.0	NULL	60
2523466	122040	Cash loans	15060.735	112500.0	136444.5	NULL	11
2819243	176158	Cash loans	47041.335	450000.0	470790.0	NULL	45
1784265	202054	Cash loans	31924.395	337500.0	404055.0	NULL	33
1383531	199383	Cash loans	23703.93	315000.0	340573.5	NULL	31
2315218	175704	Cash loans	NULL	0.0	0.0	NULL	
1656711	296299	Cash loans	NULL	0.0	0.0	NULL	
2367563	342292	Cash loans	NULL	0.0	0.0	NULL	
2579447	334349	Cash loans	NULL	0.0	0.0	NULL	
1715995	447712	Cash loans	11368.62	270000.0	335754.0	NULL	27
2257824	161140	Cash loans	13832.775	211500.0	246397.5	NULL	21
2330894	258628	Cash loans	12165.21	148500.0	174361.5	NULL	14
1397919	321676	Consumer loans	7654.86	53779.5	57564.0	0.0	5
2273188	270658	Consumer loans	9644.22	26550.0	27252.0	0.0	2
1232483	151612	Consumer loans	21307.455	126490.5	119853.0	12649.5	12
2163253	154602	Consumer loans	4187.34	26955.0	27297.0	1350.0	2
1285768	142748	Revolving loans	9000.0	180000.0	180000.0	NULL	18
2393109	396305	Cash loans	10181.7	180000.0	180000.0	NULL	18
1173070	199178	Cash loans	4666.5	45000.0	49455.0	NULL	4

DATA PREPARATION

ตรวจสอบประเภทข้อมูล และค่าว่างของข้อมูล
ด้วย `printSchema()`
ซึ่งข้อมูลมีทั้ง ประเภท double , string และ integer
จากตาราง Application Data

```
df_app.printSchema()

|-- FLOORSMIN_AVG: double (nullable = true)
|-- LANDAREA_AVG: double (nullable = true)
|-- LIVINGAPARTMENTS_AVG: double (nullable = true)
|-- LIVINGAREA_AVG: double (nullable = true)
|-- NONLIVINGAPARTMENTS_AVG: double (nullable = true)
|-- NONLIVINGAREA_AVG: double (nullable = true)
|-- APARTMENTS_MODE: double (nullable = true)
|-- BASEMENTAREA_MODE: double (nullable = true)
|-- YEARS_BEGINEXPLUATATION_MODE: double (nullable = true)
|-- YEARS_BUILD_MODE: double (nullable = true)
|-- COMMONAREA_MODE: double (nullable = true)
|-- ELEVATORS_MODE: double (nullable = true)
|-- ENTRANCES_MODE: double (nullable = true)
|-- FLOORSMAX_MODE: double (nullable = true)
|-- FLOORSMIN_MODE: double (nullable = true)
|-- LANDAREA_MODE: double (nullable = true)
|-- LIVINGAPARTMENTS_MODE: double (nullable = true)
|-- LIVINGAREA_MODE: double (nullable = true)
|-- NONLIVINGAPARTMENTS_MODE: double (nullable = true)
|-- NONLIVINGAREA_MODE: double (nullable = true)
|-- APARTMFTS_MFDT: double (nullable = true)
```

DATA PREPARATION

ตรวจสอบประเกกข้อมูล และค่าว่างของข้อมูล

ด้วย `.printSchema()`

ซึ่งข้อมูลมีทั้ง ประเกก `double` , `string` และ `integer`

จากตาราง Previous Application

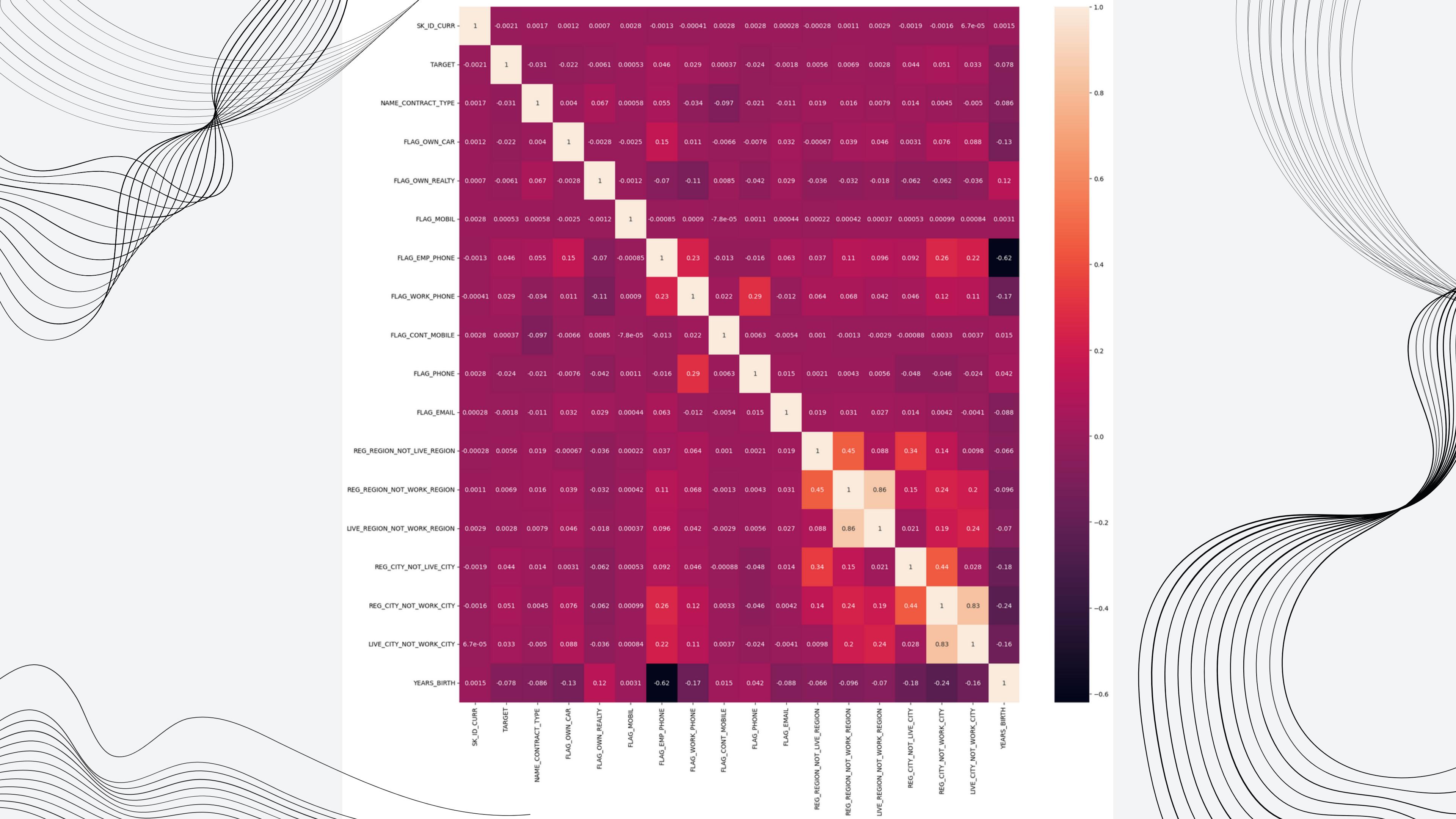
```
df_pv_app.printSchema()
```

```
root
| -- SK_ID_PREV: integer (nullable = true)
| -- SK_ID_CURR: integer (nullable = true)
| -- NAME_CONTRACT_TYPE: string (nullable = true)
| -- AMT_ANNUITY: double (nullable = true)
| -- AMT_APPLICATION: double (nullable = true)
| -- AMT_CREDIT: double (nullable = true)
| -- AMT_DOWN_PAYMENT: double (nullable = true)
| -- AMT_GOODS_PRICE: double (nullable = true)
| -- WEEKDAY_APPR_PROCESS_START: string (nullable = true)
| -- HOUR_APPR_PROCESS_START: integer (nullable = true)
| -- FLAG_LAST_APPL_PER_CONTRACT: string (nullable = true)
| -- NFLAG_LAST_APPL_IN_DAY: integer (nullable = true)
| -- RATE_DOWN_PAYMENT: double (nullable = true)
| -- RATE_INTEREST_PRIMARY: double (nullable = true)
| -- RATE_INTEREST_PRIVILEGED: double (nullable = true)
| -- NAME_CASH_LOAN_PURPOSE: string (nullable = true)
| -- NAME_CONTRACT_STATUS: string (nullable = true)
| -- DAYS_DECISION: integer (nullable = true)
| -- NAME_PAYMENT_TYPE: string (nullable = true)
| -- CODE_REJECT_REASON: string (nullable = true)
| -- NAME_TYPE_SUITE: string (nullable = true)
```

การสำรวจข้อมูล

EXPLORATORY DATA ANALYSIS (EDA)

โดยการวิเคราะห์ระดับความสัมพันธ์ของตัวแปร ค้นหาว่าตัวแปรแต่ละตัวมีความสัมพันธ์กันมากน้อยเพียงใด วัตถุประสงค์เพื่อลดจำนวนตัวแปรที่ใช้ในการพัฒนาแบบจำลอง เพื่อความรวดเร็วในการทำแบบจำลอง และเพื่อเพิ่มประสิทธิภาพในการเรียนรู้ข้อมูลของแบบจำลอง และยังทำให้แบบจำลองที่ได้มีประสิทธิภาพในการทำนายที่มีความแม่นยำมากยิ่งขึ้น โดยการทำการสำรวจความสัมพันธ์ของแต่ละคอลัมน์ โดยการใช้เทคนิคที่เรียกว่า correlation ในการแสดงค่าความสัมพันธ์ของข้อมูลในแต่ละคอลัมน์ โดยค่า Correlation จะมีค่าอยู่ระหว่าง -1 ถึง 1



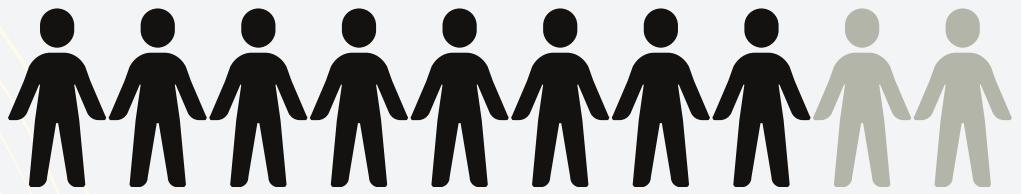


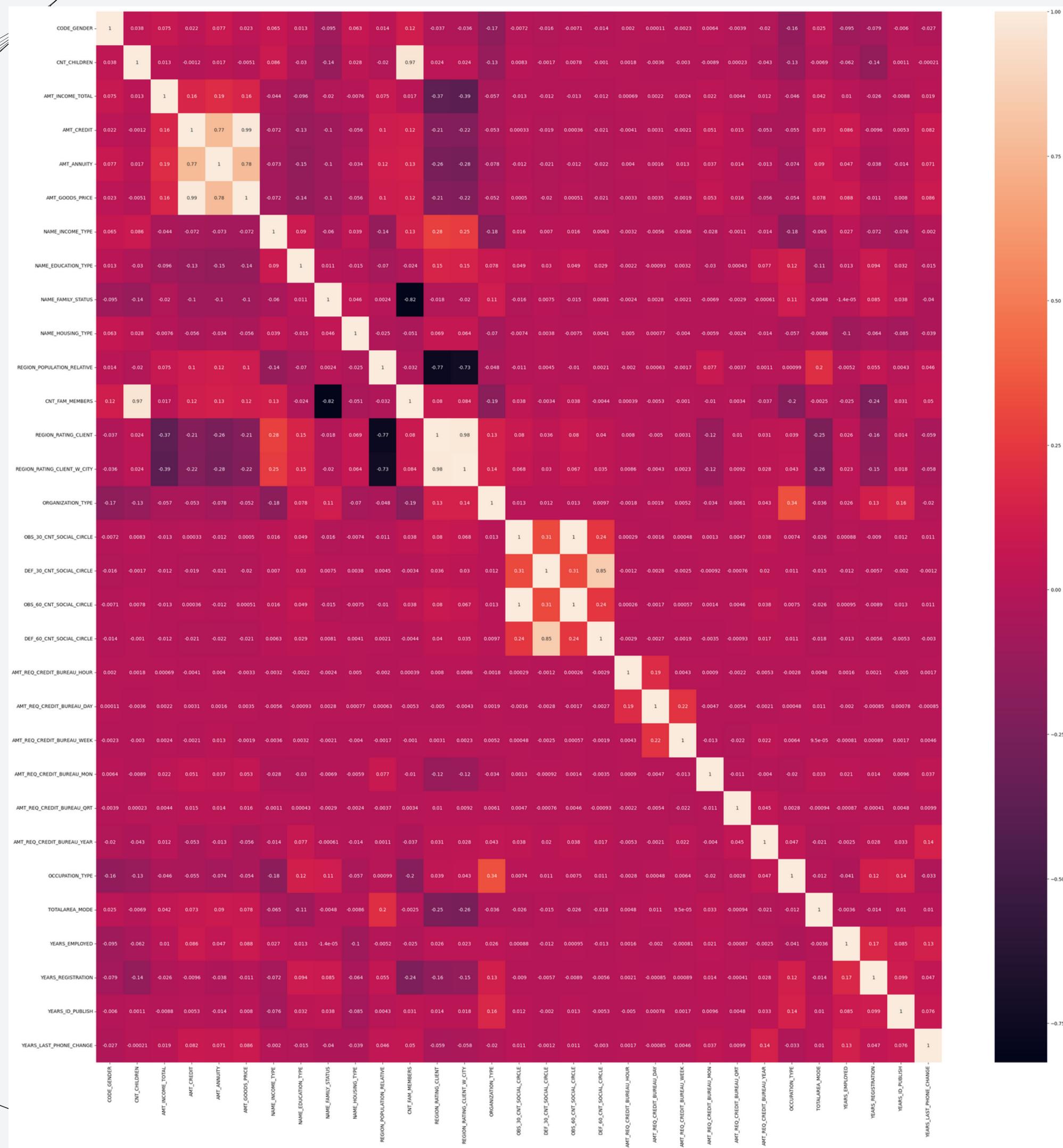
int

ซึ่งจากการหาความสัมพันธ์ของข้อมูลในแต่ละคอลัมน์ ที่จัดเก็บข้อมูลบน int จะเห็นได้ว่ามีบางค่าคอลัมน์ที่มีค่าความสูงโดย REG_REGION_NOT_WORK_REGION และ LIVE_REGION_NOT_WORK_REGION มีความสัมพันธ์ เท่ากับ 0.86 และ REG_CITY_NOT_WORK_CITY และ LIVE_CITY_NOT_WORK_CITY มีความสัมพันธ์เท่ากับ 0.83



0.86

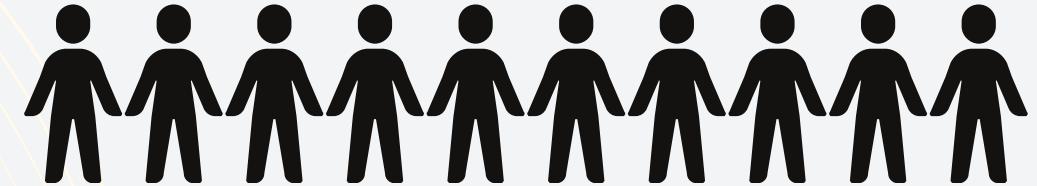




float

ซึ่งจากการหาความสัมพันธ์ของข้อมูลในแต่ละคอลัมน์ ที่จัดเก็บข้อมูลนิด float จะเห็นได้ว่ามีบางค่าคอลัมน์ที่มีความสูงมากๆ โดย ATM_CREDIT และ ATM_GOODS_PRICE มีความสัมพันธ์ เก่ากับ 0.99 และ REGION_RATING_CLIENT และ REGION_RATING_CIENT_W_CITY มีความสัมพันธ์เก่ากับ 0.98

0.99

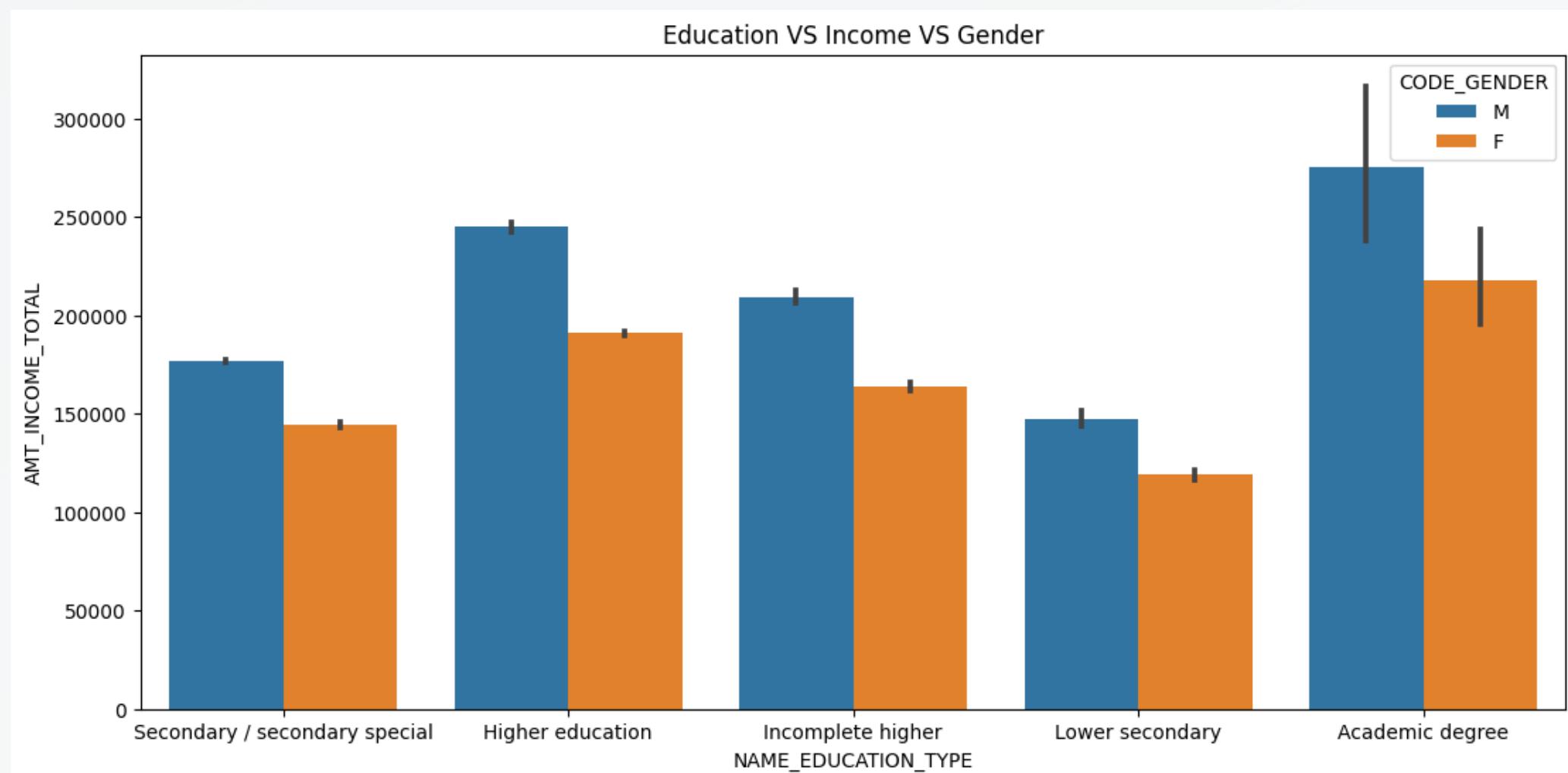


การสำรวจข้อมูล

EXPLORATORY DATA ANALYSIS (EDA)

Observation

สังเกตได้ว่าผู้ชายมีการศึกษามากกว่าและได้รับค่าตอบแทนดีกว่าผู้หญิง

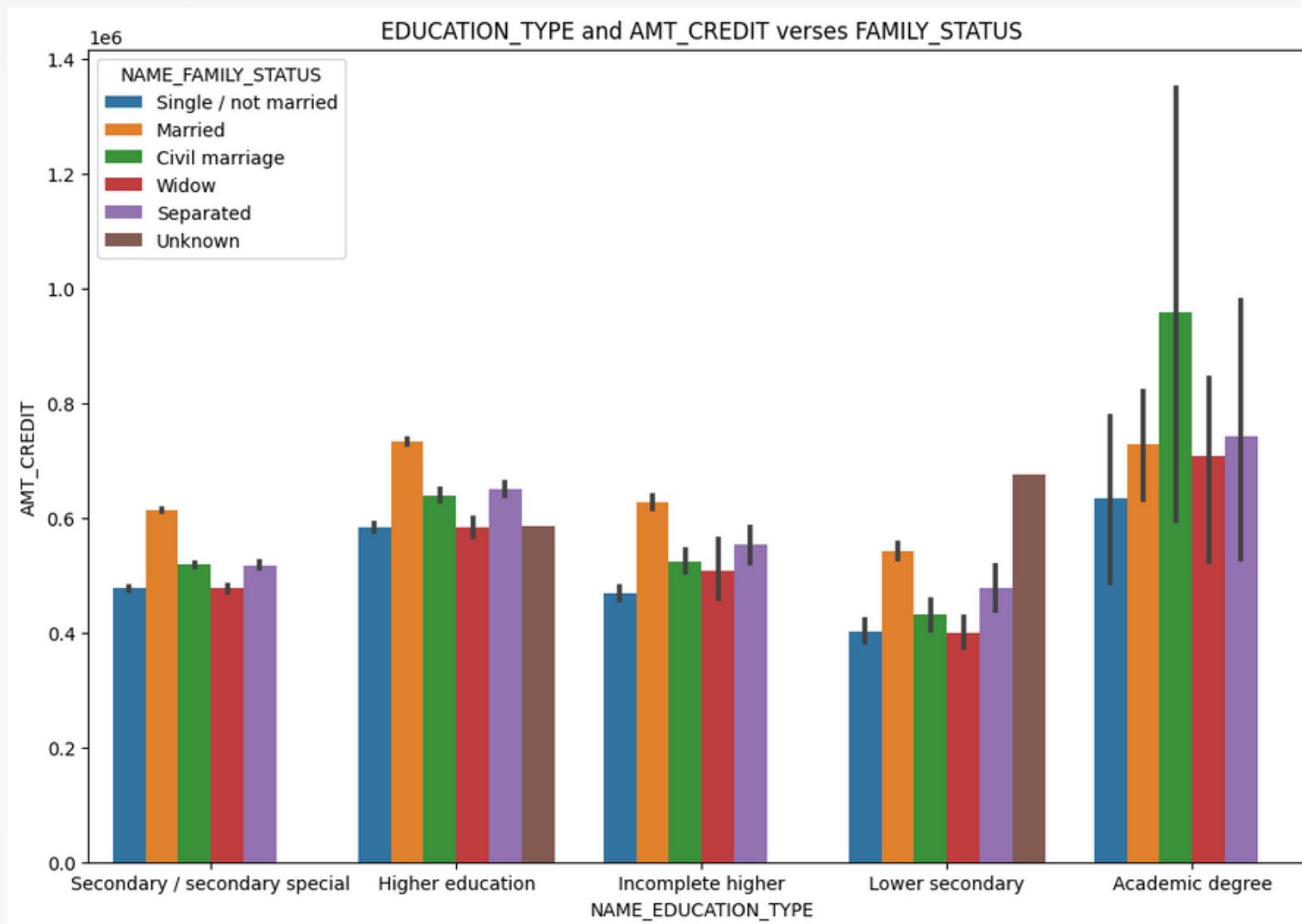


การสำรวจข้อมูล

EXPLORATORY DATA ANALYSIS (EDA)

Observations

เราจะเห็นว่าผู้สมัครได้รับเครดิตจำนวนเงินกู้ส่วนใหญ่ที่มีสถานะทางครอบครัวเป็นการสมรสและมีวุฒิการศึกษา^{รายังสังเกตเห็นว่าโซด/ไม่ได้แต่งงานมี Amt_Credit ต่ำที่สุด โดยไม่คำนึงถึงประเภทการศึกษา}

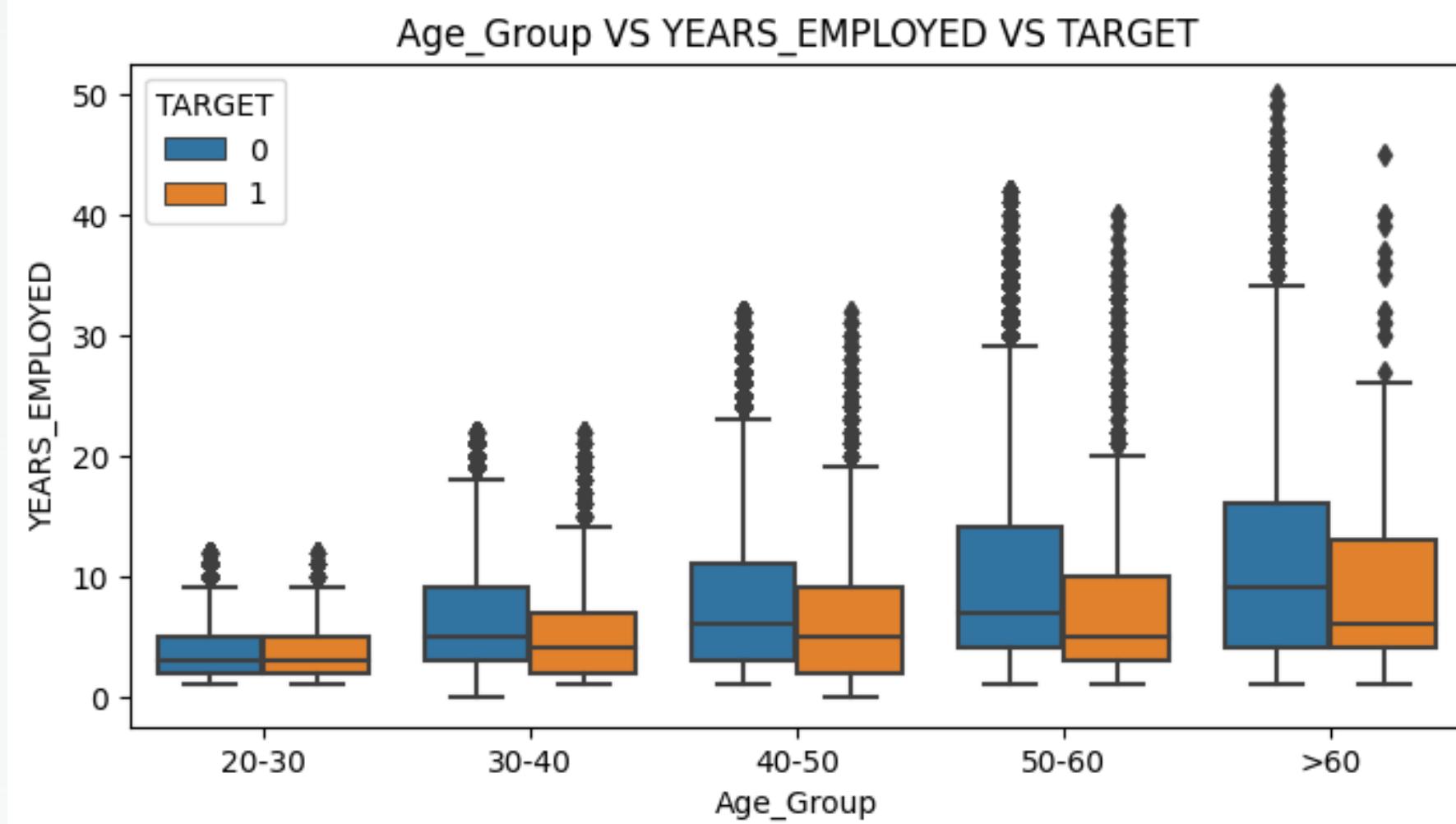


การสำรวจข้อมูล

EXPLORATORY DATA ANALYSIS (EDA)

Observations

- เมื่ออายุเพิ่มขึ้น ความน่าจะเป็นในการชำรุดเงินกู้จะสูงขึ้นโดยไม่คำนึงถึงปัจจัยทำงาน



PREPARING DATA



จากจากชุดข้อมูลที่เรานำมาใช้ในการวิเคราะห์เพื่อพัฒนาแบบจำลอง มีข้อมูลที่ขาดหายไป (Missing Value) เป็นจำนวนมาก ซึ่งจะจัดการกับข้อมูลที่ขาดหายไปก่อนที่จะนำข้อมูลเหล่านี้ไปทำแบบจำลอง เพื่อให้ได้แบบจำลองที่มีประสิทธิภาพที่ดียิ่งขึ้น โดยจะจัดการกับข้อมูลที่ขาดหายไป ทั้งในแนวระดับแควและในแนวระดับคอลัมน์ โดยจะจำจัดคอลัมน์ที่มีข้อมูลที่ขาดหายไปมากกว่า 45% ซึ่งเราจะไม่นำคอลัมน์เหล่านี้ มาใช้ในการวิเคราะห์ข้อมูลเพื่อกำหนด 122 คอลัมน์ และทำการลบคอลัมน์ที่มีจำนวนข้อมูลที่ขาดหายไปมากกว่า 45% จะทำให้เหลือคอลัมน์ที่นำมาใช้ในการวิเคราะห์ข้อมูลเพื่อกำหนด 74 คอลัมน์

COMMONAREA_MEDI	69.872297
COMMONAREA_AVG	69.872297
COMMONAREA_MODE	69.872297
NONLIVINGAPARTMENTS_MODE	69.432963
NONLIVINGAPARTMENTS_AVG	69.432963
...	...
NAME_HOUSING_TYPE	0.000000
NAME_FAMILY_STATUS	0.000000
NAME_EDUCATION_TYPE	0.000000
NAME_INCOME_TYPE	0.000000
SK_ID_CURR	0.000000

Length: 122, dtype: float64

PREPARING DATA

2

ลบคอลัมน์ที่เป็น “FLAG_DOCUMENT” ออกรหัสหนด เนื่องจาก เป็นคอลัมน์ที่ใช้ในการเก็บข้อมูลที่เกี่ยวข้องกับการเก็บเอกสาร ซึ่งไปป่าจะมีประโยชน์ที่จะนำคอลัมน์เหล่านี้ไปใช้ในการวิเคราะห์ ข้อมูลเพื่อทำแบบจำลอง โดยในขั้นตอนนี้ได้ มีการลบคอลัมน์ที่ เกี่ยวข้องกับการเก็บเอกสาร จะทำให้เหลือคอลัมน์ที่จะนำไปใช้ในการวิเคราะห์ข้อมูลเพื่อทำแบบจำลองทั้งหมด 49 คอลัมน์

3

ลบข้อมูลบางแควในชุดข้อมูลออก เพราะข้อมูลในแควนี้ส่วนใหญ่ จะเก็บข้อมูลที่เป็นค่าว่าง (NaN) เนื่องจากมีข้อมูลจำนวน 307,511 แคว จึงสามารถลบข้อมูลส่วนนี้ออก กีบข้อมูลที่เป็นค่าว่างออกได้ แล้ว เมื่อทำการทำความสะอาดข้อมูล (Data cleansing) เรียบร้อย แล้ว จะพบว่าข้อมูลที่ถูกจัดเก็บอยู่ในคอลัมน์ทั้งหมด ไม่มีคอลัมน์ ไหนที่มีการจัดเก็บข้อมูลที่ เป็นค่าว่างอีกแล้ว

```
1 # Create a Spark session
2 spark = SparkSession.builder.appName("DropColumns").getOrCreate()
3
4 # Assuming you have your data as a DataFrame named "app_data_clean" in PySpark
5 # Define the list of columns to drop
6 columns_to_drop = ['FLAG_DOCUMENT_2', 'FLAG_DOCUMENT_3', 'FLAG_DOCUMENT_4', 'FLAG_DOCUMENT_5',
7                     'FLAG_DOCUMENT_6', 'FLAG_DOCUMENT_7', 'FLAG_DOCUMENT_8', 'FLAG_DOCUMENT_9',
8                     'FLAG_DOCUMENT_10', 'FLAG_DOCUMENT_11', 'FLAG_DOCUMENT_12', 'FLAG_DOCUMENT_13',
9                     'FLAG_DOCUMENT_14', 'FLAG_DOCUMENT_15', 'FLAG_DOCUMENT_16', 'FLAG_DOCUMENT_17',
10                    'FLAG_DOCUMENT_18', 'FLAG_DOCUMENT_19', 'FLAG_DOCUMENT_20', 'FLAG_DOCUMENT_21']
11
12 # Drop the specified columns
13 df_data_clean = df_data_clean.drop(*columns_to_drop)
```

PREPARING DATA

4

ลบข้อมูลบางแควที่เก็บข้อมูลเพศเป็นค่า “XNA” ออก
เนื่องจากเมื่อทำการนับจำนวนข้อมูลในคอลัมน์เพศ
แล้ว พบร่วมกับข้อมูลเพศก็คงเหลือ 3 ค่า คือ M, F, XNA
ซึ่งข้อมูลจริงๆ ก็ถูกต้องที่ควรจัดเก็บ ควรนีแค่ 2 ค่า
คือ เพศชายและเพศหญิง (M, F)

5

ลบบางคอลัมน์ที่เก็บข้อมูลส่วนใหญ่ที่เป็นค่า “1” ออก
 เพราะคอลัมน์เหล่านี้ ไม่ค่อยมีความแตกต่างเมื่อนำมา^{ใช้ในการวิเคราะห์เพื่อกำหนดแบบจำลอง คอลัมน์เหล่านี้จะ}
ไม่มีผลต่อการทำนาย ที่จะสามารถนำไปใช้ในการแบ่ง
แยกข้อมูลได้

REGION_POPULATION_RELATIVE	FLAG_MOBIL	FLAG_EMP_PHONE	FLAG_WORK_PHONE	FLAG_PHONE
0.018801	1	1	0	0
0.003540999999999999	1	1	0	0
0.010032	1	1	1	1
0.008019	1	1	1	0
0.028663	1	1	1	0
0.03579200000000004	1	1	1	1
0.03579200000000004	1	1	1	0
0.003121999999999...	1	1	1	1
0.018634	1	0	0	0
0.01968899999999998	1	1	0	0
0.0228	1	1	1	0
0.015221	1	0	0	0
0.031329	1	1	1	1
0.01661200000000002	1	1	0	0
0.01000600000000001	1	1	1	0
0.020713	1	1	0	0
0.018634	1	1	0	0
0.010966	1	1	0	0
0.04622	1	1	1	0
0.015221	1	1	1	1

PREPARING DATA

6

เพิ่มคอลัมน์ที่เก็บอายุของลูกหนี้ โดยการแปลงข้อมูลจากข้อมูลในคอลัมน์ DAYS_BIRTH ที่เก็บข้อมูลวันเกิดของลูกหนี้ แต่เปลี่ยนจากการเก็บในลักษณะของวันให้กลายเป็นปี เพื่อนำไปเป็นปัจจัยที่ช่วยในการทำนายลูกหนี้ที่มีโอกาสในการผิดนัดชำระกับทางธนาคาร

7

ทำการแปลงชนิด (Type) ของข้อมูล เนื่องจากเมื่อเราทำการดูชนิดของข้อมูลแล้ว ยังมีบางคอลัมน์ที่เก็บชนิดของข้อมูลไม่ตรงกับลักษณะของข้อมูลที่ถูกจัดเก็บอยู่จริงๆ จึงได้มีการแปลงชนิดของข้อมูล ให้มีชนิดของข้อมูลที่ถูกต้อง ก่อนนำไปทำแบบจำลอง

RangeIndex: 307511 entries, 0 to 307510			
Data columns (total 49 columns):			
#	Column	Non-Null Count	Dtype
0	SK_ID_CURR	307511	non-null float64
1	TARGET	307511	non-null float64
2	NAME_CONTRACT_TYPE	307511	non-null float64
3	CODE_GENDER	307507	non-null float64
4	FLAG_OWN_CAR	307511	non-null float64
5	FLAG_OWN_REALTY	307511	non-null float64
6	CNT_CHILDREN	280762	non-null float64
7	AMT_INCOME_TOTAL	307511	non-null float64
8	AMT_CREDIT	307511	non-null float64
9	AMT_ANNUITY	307499	non-null float64
10	AMT_GOODS_PRICE	307233	non-null float64
11	NAME_INCOME_TYPE	307506	non-null float64
12	NAME_EDUCATION_TYPE	297234	non-null float64
13	NAME_FAMILY_STATUS	287741	non-null float64
14	NAME_HOUSING_TYPE	296328	non-null float64
15	REGION_POPULATION_RELATIVE	307511	non-null float64
16	FLAG_MOBIL	307511	non-null float64
17	FLAG_EMP_PHONE	307511	non-null float64
18	FLAG_WORK_PHONE	307511	non-null float64
19	FLAG_CONT_MOBILE	307511	non-null float64
20	FLAG_PHONE	307511	non-null float64
21	FLAG_EMAIL	307511	non-null float64
22	CNT_FAM_MEMBERS	149152	non-null float64
23	REGION_RATING_CLIENT	80527	non-null float64
24	REGION_RATING_CLIENT_W_CITY	78027	non-null float64
25	REG_REGION_NOT_LIVE_REGION	307511	non-null float64
26	REG_REGION_NOT_WORK_REGION	307511	non-null float64
27	LIVE_REGION_NOT_WORK_REGION	307511	non-null float64
28	REG_CITY_NOT_LIVE_CITY	307511	non-null float64
29	REG_CITY_NOT_WORK_CITY	307511	non-null float64
30	LIVE_CITY_NOT_WORK_CITY	307511	non-null float64
31	ORGANIZATION_TYPE	305004	non-null float64
32	OBS_30_CNT_SOCIAL_CIRCLE	276682	non-null float64
33	DEF_30_CNT_SOCIAL_CIRCLE	301167	non-null float64
34	OBS_60_CNT_SOCIAL_CIRCLE	276724	non-null float64
35	DEF_60_CNT_SOCIAL_CIRCLE	303320	non-null float64
36	AMT_REQ_CREDIT_BUREAU_HOUR	265936	non-null float64
37	AMT_REQ_CREDIT_BUREAU_DAY	265886	non-null float64
38	AMT_REQ_CREDIT_BUREAU_WEEK	265793	non-null float64
39	AMT_REQ_CREDIT_BUREAU_MON	260606	non-null float64
40	AMT_REQ_CREDIT_BUREAU_QRT	251580	non-null float64
41	AMT_REQ_CREDIT_BUREAU_YEAR	215800	non-null float64
42	OCCUPATION_TYPE	301565	non-null float64
43	TOTALAREA_MODE	159080	non-null float64
44	YEARS_BIRTH	307511	non-null float64
45	YEARS_EMPLOYED	220292	non-null float64
46	YEARS_REGISTRATION	291937	non-null float64
47	YEARS_ID_PUBLISH	291669	non-null float64
48	YEARS_LAST_PHONE_CHANGE	249889	non-null float64

dtypes: float64(49)

FEATURE TRANSFORMATIONS WITH ENSEMBLES OF TREES

8

Feature transformations with ensembles of trees

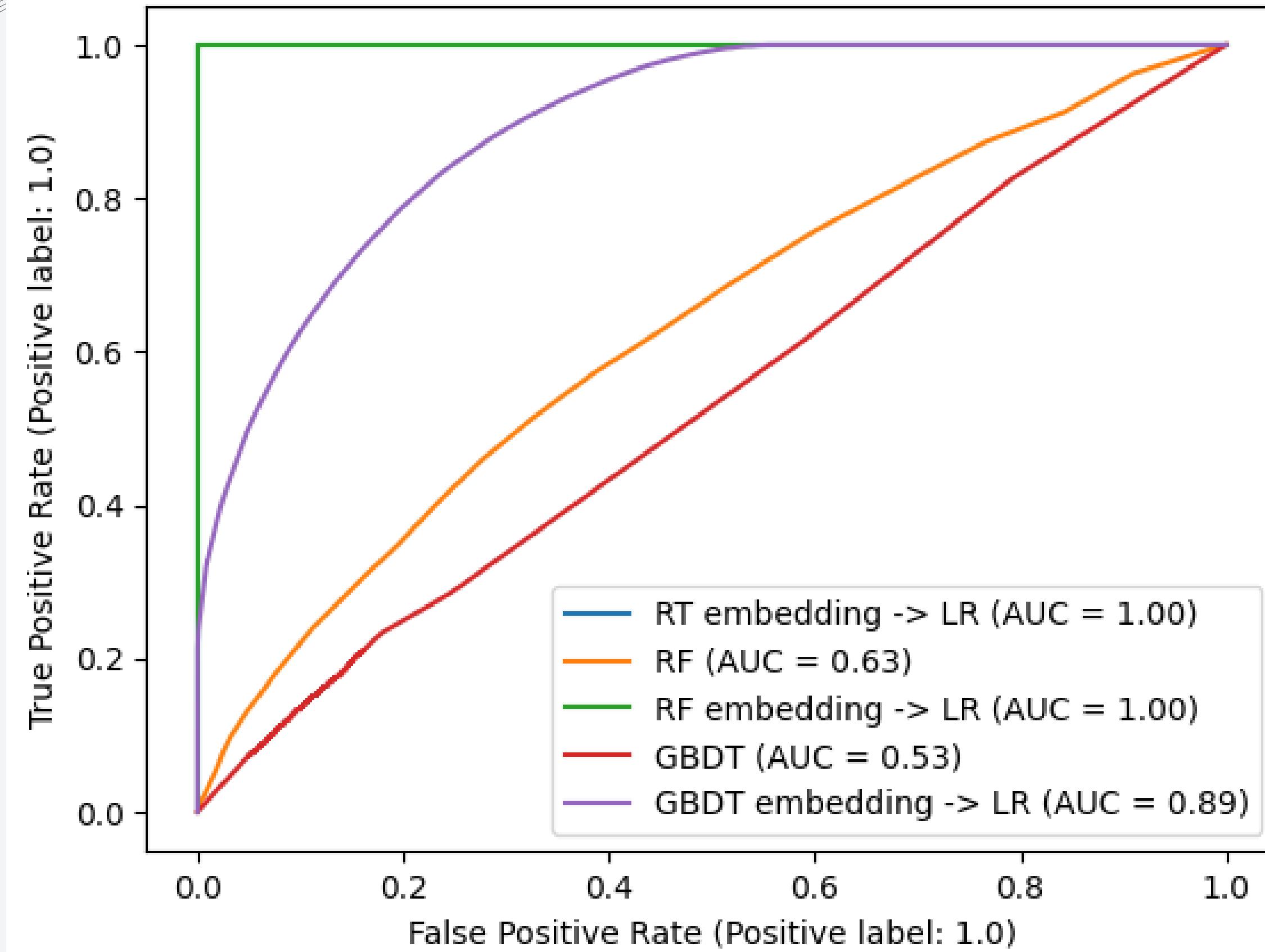
หมายถึงการใช้อัลกอริทึมที่ใช้ต้นไม้ (trees) ในกระบวนการการเรียนรู้และการแปลงคุณลักษณะ (feature transformation) ของข้อมูล. อัลกอริทึมที่ใช้ต้นไม้ในกระบวนการการเรียนรู้ เช่น Random Forest, Gradient Boosting, หรือ Extreme Gradient Boosting (XGBoost) มักจะมีความสามารถในการจัดการกับคุณลักษณะ (features) ที่มีความซับซ้อนหรือที่ไม่เหมาะสมสำหรับการสร้างโมเดลเชิงเส้นหรือโมเดลอื่น ๆ ที่ใช้ข้อมูลที่มีคุณลักษณะตัวเลข (numerical features).

ขั้นตอนหลักใน Feature transformations with ensembles of trees ประกอบด้วย:

- การเรียนรู้โดยใช้ต้นไม้: ใช้อัลกอริทึมที่ใช้ต้นไม้ (tree-based algorithms) เพื่อสร้างโมเดลที่ใช้ข้อมูลคุณลักษณะ โดยปกติอัลกอริทึมเหล่านี้จะสร้างต้นไม้หลายต้นและใช้การผสานผ่านหลายต้นเพื่อกำหนายผลลัพธ์.
- การแปลงคุณลักษณะ: ในกระบวนการการเรียนรู้, อัลกอริทึมต้นไม้อาจสร้างคุณลักษณะใหม่ ๆ หรือแปลงคุณลักษณะที่มีอยู่เพื่อให้สามารถนำมาใช้ในการคำนวณได้ดีขึ้น. ตัวอย่างเช่น, ในการแปลงคุณลักษณะอาจจะมีการรวมคุณลักษณะที่มีความสัมพันธ์กันหรือการสร้างคุณลักษณะใหม่จากคุณลักษณะที่มีอยู่.
- การนำคุณลักษณะที่แปลงมาไปใช้ในการคำนวณ: คุณลักษณะที่ถูกแปลงหรือสร้างขึ้นจากกระบวนการการเรียนรู้โดยใช้ต้นไม้จะถูกนำมาใช้ในการคำนวณผลลัพธ์หรือคลาสของข้อมูล.

การใช้ Feature transformations with ensembles of trees ช่วยให้การจัดการกับข้อมูลที่ซับซ้อนและการคำนวณที่แม่นยำมากขึ้น

ROC curve



วิธีการวัดเส้นโค้ง ROC

AUC = 1: โมเดลมีประสิทธิภาพสูงมากที่สุดในการจำแนกคลาส

AUC > 0.5: โมเดลมีประสิทธิภาพในการจำแนกคลาสดีกว่าการทายสุ่ม

AUC = 0.5: โมเดลไม่มีประสิทธิภาพในการจำแนกคลาสและเกี้ยบเท่ากับการทายสุ่ม

AUC < 0.5: โมเดลมีประสิทธิภาพในการจำแนกคลาสที่แย่กว่าการทายสุ่ม

- มี 2 Model ที่ AUC = 1 มีประสิทธิภาพสูงในการจำแนกคลาส

RT embedding -> LR [ใช้ RandomTreesEmbedding (RT) เพื่อแปลงคุณลักษณะ (features) ของข้อมูลเป็นรูปแบบใหม่ และใช้ Logistic Regression เพื่อจำแนกคลาส]

RF embedding -> LR [ใช้ Random Forest (RF) การแปลงคุณลักษณะข้อมูล และ Logistic Regression ในกระบวนการการทำนายด้วยหลายขั้นตอน]

- Model ที่มีประสิทธิภาพรองลงมา ก็ คือ

GBDT embedding -> LR [โมเดลแบบประสานงานที่สามารถใช้ในการทำนายผลลัพธ์ของข้อมูล ในรูปแบบที่แปลงไปของตัวไม้จ้า GBDT ให้กลายเป็นตัวแปรดั้มมีและนำมาใช้กับ Logistic Regression ในการทำนาย]

- Model ที่มีประสิทธิภาพดีกว่าการทายสุ่มมากน้อย คือ

random_forest

gradient_boosting

ROC (Receiver Operating Characteristic)

Curve เป็นเส้นที่ใช้วัดถึงประสิทธิภาพของโมเดลแบบ Classification ว่าสามารถทำนายประเด็นที่สนใจได้อย่างแม่นยำขนาดไหน (โดยกrückไปนิยมวัดประสิทธิภาพของโมเดลแบบ Binary)

AUC

Area Under ROC Curve หรือพื้นที่ใต้กราฟ ROC

****AUC มีค่าสูงก็หมายความว่าโมเดลมีประสิทธิภาพมาก**

อัลกอริทึม ที่จะถูกนำมาใช้ใน การทำแบบจำลอง

1. Logistic Regression

เป็นอัลกอริทึมที่ใช้ในการสร้างโมเดล การจำแนกคลาส (Classification Model) และจำแนกคลาสเป็นสองกลุ่มหรือมากกว่าสองกลุ่มโดยใช้การสร้างฟังก์ชันสร้างของค่าคลาส (Class Probability) ที่เป็นไปได้จากคุณลักษณะ (Features) หรือตัวแปรอิสระ (Independent Variables) และประสิทธิภาพในการจำแนกคลาสจากข้อมูลอบรมอุ่น (Training Data)

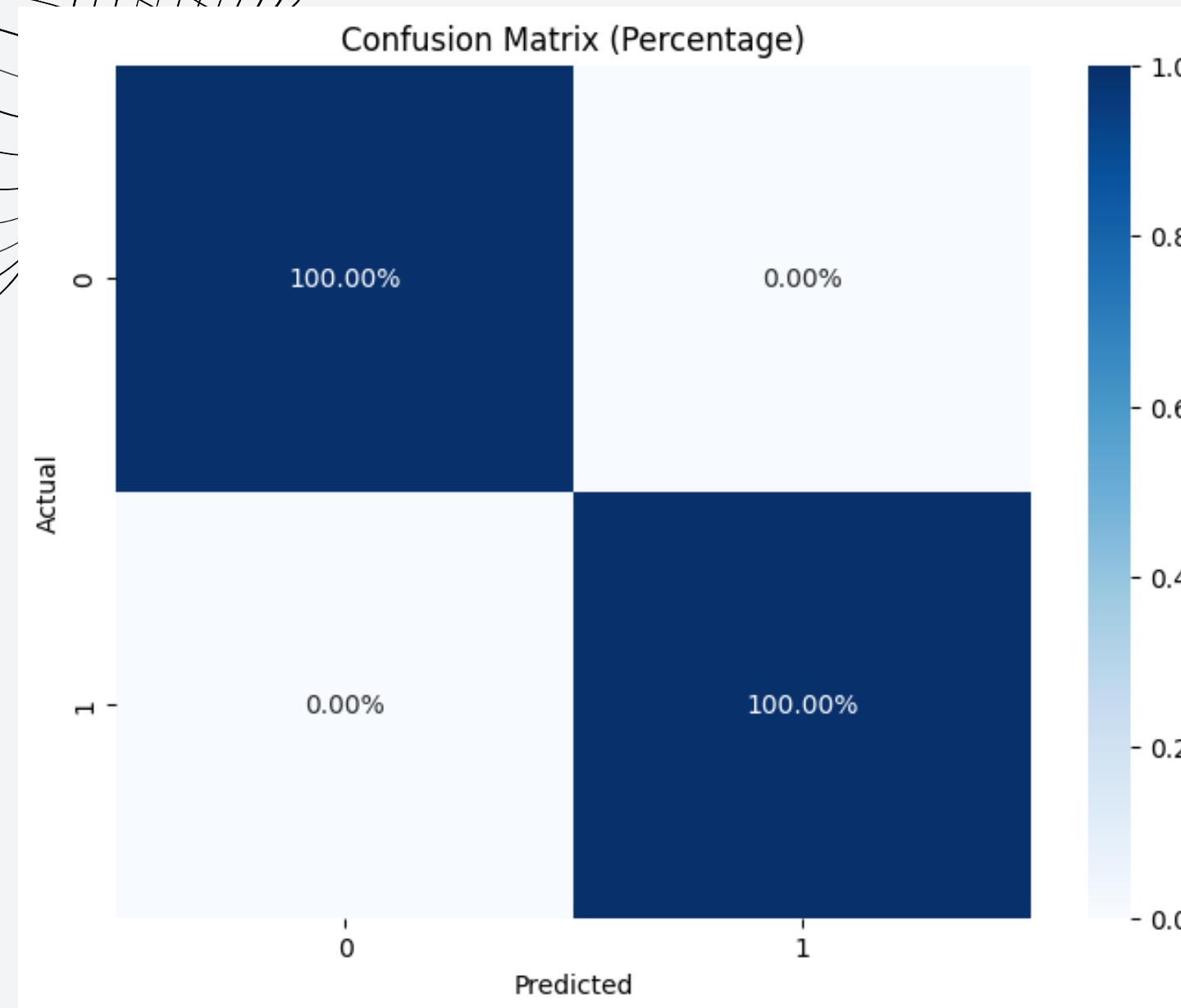
2.XGBoostClassifier

เทคโนโลยีต้นไม้ตัดสินใจหลายต้นมาช่วยกันในการตัดสินใจ มากช่วยกันในการทำงานเพื่อทำให้ผลลัพธ์ที่ดียิ่งขึ้น ซึ่งการทำงานของแบบจำลอง XGBoost คือการสร้างต้นไม้ตัดสินใจหลายต้น โดยที่ต้นไม้ตัดสินใจแต่ละต้นจะถูกสร้างขึ้นมาจากการปรับปรุงประสิทธิภาพของแบบจำลองที่ถูกสร้างขึ้นก่อนหน้า แล้วจะพยายามแก้ไขความผิดพลาด(error) ของแบบจำลองที่ถูกสร้างขึ้นก่อนหน้า ให้แบบจำลองที่ถูกสร้างขึ้นในครั้งถัดๆไป มีความถูกต้องแม่นยำในการทำงานมากยิ่งขึ้นเรื่อยๆ เมื่อมีการเรียนรู้ของต้นไม้ตัดสินใจต่อเนื่องกันจนมีความลึกมากพอก แบบจำลองจะหยุดการเรียนรู้ก็ต่อเมื่อไม่เหลือค่าความผิดพลาดจากต้นไม้ตัดสินใจก่อนหน้าให้เรียนรู้แล้ว

3.KNN Classifier

เป็นอัลกอริทึมการจำแนกคลาสใน Machine Learning ซึ่งใช้กลุ่มข้อมูล (data points) ที่ใกล้เคียงกันในคุณลักษณะ (features) ในการคาดการณ์คลาสของข้อมูลที่ไม่รู้คลาส (unlabeled data) โดยอัลกอริทึม KNN มีลักษณะง่ายและเป็นอัลกอริทึมที่ไม่พึ่งพา參數 (parameter-free) คือไม่ต้องใช้การเรียนรู้ (training) ก่อน และทำงานโดยการค้นหาข้อมูลที่ใกล้กับสุดจากข้อมูลทดสอบ (test data) ในชุดข้อมูล (training data) โดยใช้ค่า K (จำนวนข้อมูลใกล้เคียง) ที่กำหนดจากผู้ใช้ เมื่อพบข้อมูลที่ใกล้กับสุด K ตัวแรก อัลกอริทึมจะใช้โหวตจากคลาสของข้อมูลเหล่านี้เพื่อกำหนดคลาสของข้อมูลทดสอบ (test data).

ຕົວຢ່າງ Logistic Regression



F2 Score: 1.0

Accuracy: 1.0

ROC AUC: 1.0

Classification Report:

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	30303
1.0	1.00	1.00	1.00	2717
accuracy			1.00	33020
macro avg	1.00	1.00	1.00	33020
weighted avg	1.00	1.00	1.00	33020

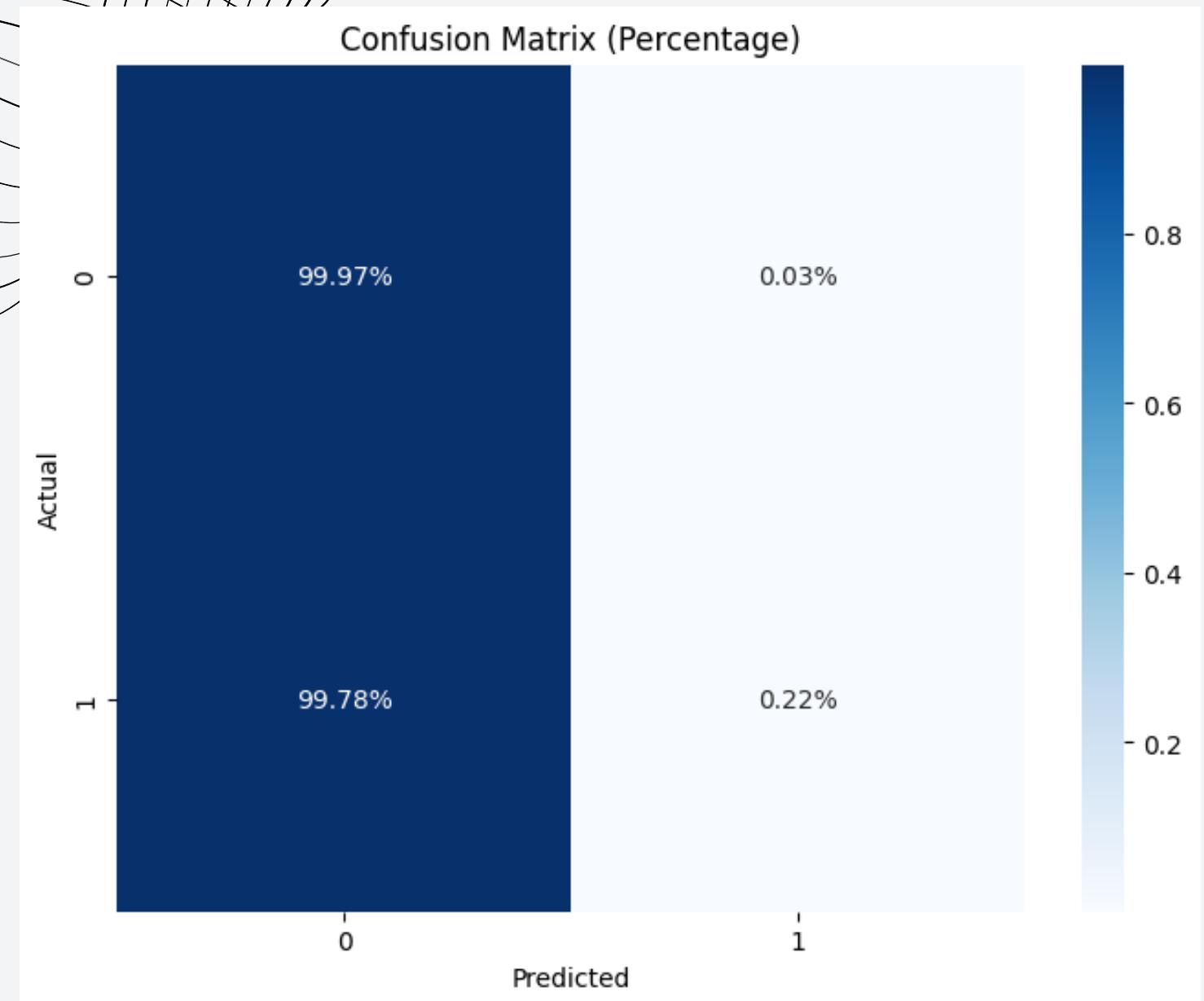
ສ້າງ RandomTreesEmbedding

```
random_tree_embedding = RandomTreesEmbedding(n_estimators=100,  
max_depth=50, random_state=42)
```

ສ້າງໂມເດລໄອຍຮັນ RandomTreesEmbedding ແລະ Logistic Regression ໃນ pipeline

```
rt_model = make_pipeline(random_tree_embedding,  
LogisticRegression(max_iter=100))
```

ตัวอย่าง XGBoostClassifier



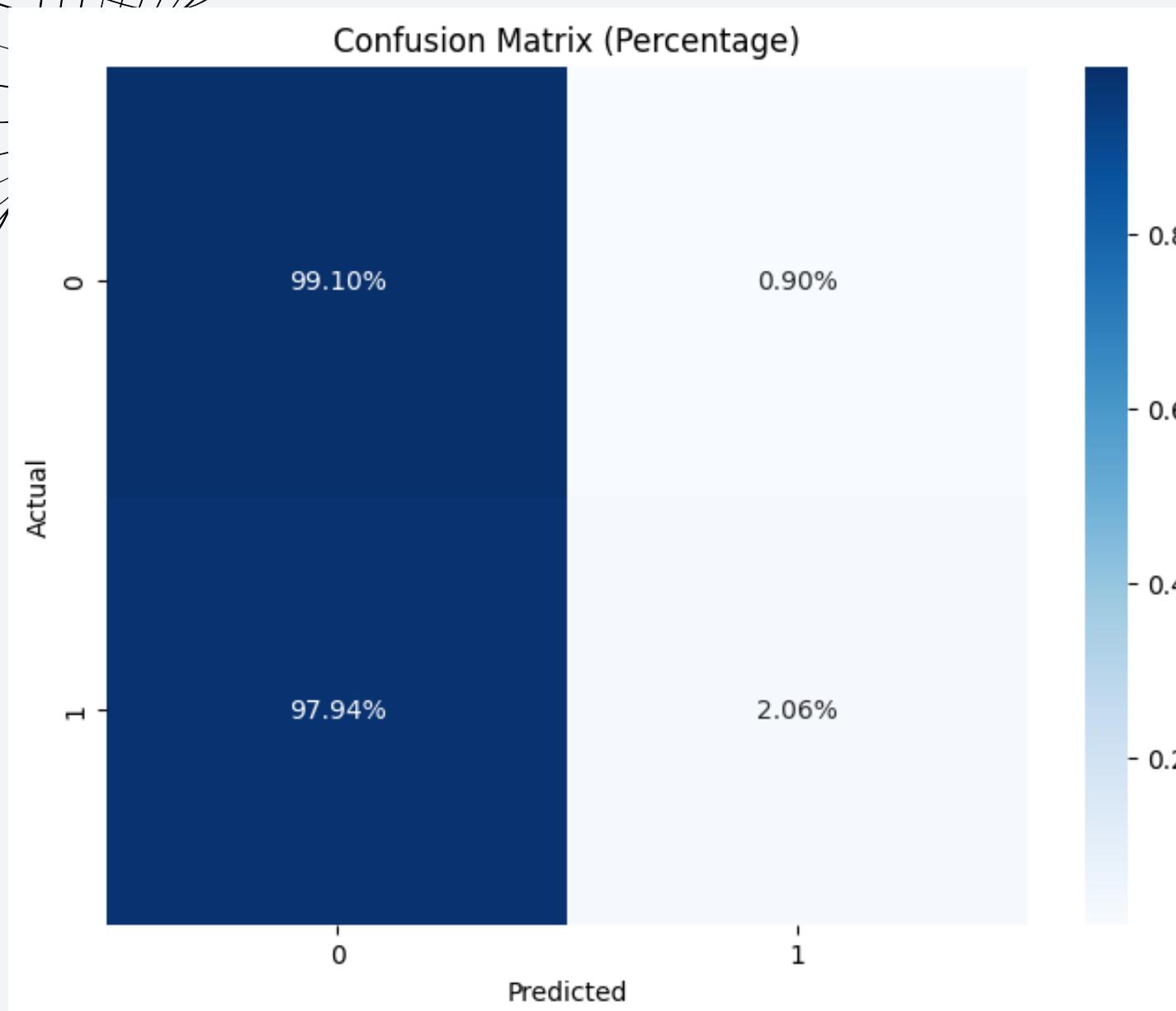
➡ Accuracy: 0.917655966081163
Precision: 0.42857142857142855
Recall: 0.002208317997791682
F1 Score: 0.004393994873672648
F2 Score: 0.0027568461679838272
ROC AUC Score: 0.6448225638509039

```
# สร้างและฝึกโมเดล Random Trees Embedding (RTE)
rte = RandomTreesEmbedding(n_estimators=100, random_state=42)
X_train_transformed = rte.fit_transform(X_train)
X_test_transformed = rte.transform(X_test)

# สร้างและฝึกโมเดล XGBoost โดยใช้คุณลักษณะที่แปลงแล้ว
xgb_model = XGBClassifier()
xgb_model.fit(X_train_transformed, y_train)

# ทำนายคลาสโดยใช้โมเดล XGBoost
y_pred = xgb_model.predict(X_test_transformed)
```

ຕົວອຍ່າງ KNN Classifier

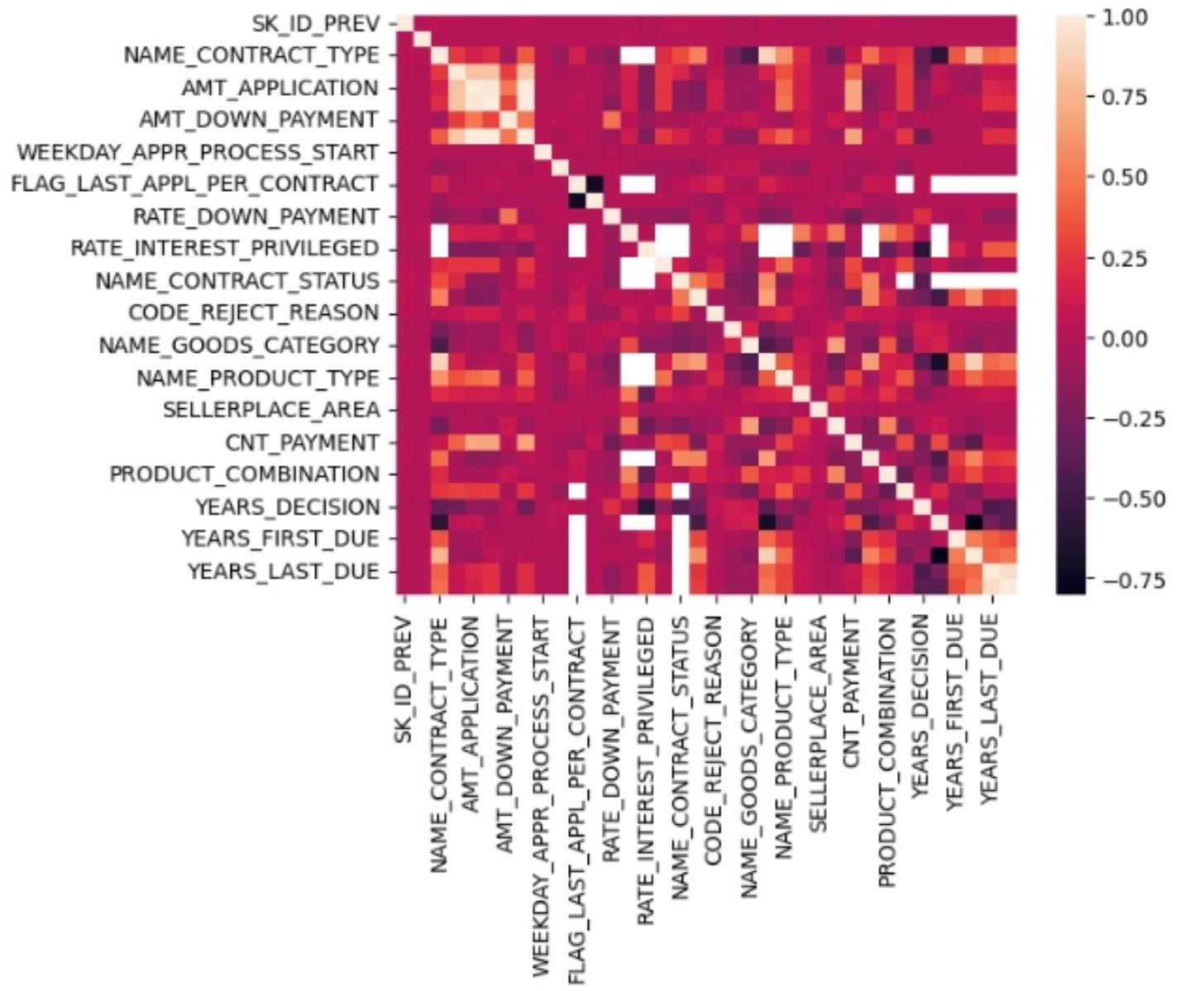


Accuracy: 0.9111750454270139
Precision: 0.17073170731707318
Recall: 0.020610967979389033
F1 Score: 0.036781609195402305
F2 Score: 0.02500893176134334
ROC AUC Score: 0.6448225638509039

```
# ສ້າງແລະຝຶກໂນໂດເ Random Trees Embedding (RTE)
# Transform the features using RF
# Create and train the K-Nearest Neighbors (KNN) classifier using the
# transformed features
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train_transformed, y_train)

# Make predictions using the KNN model
y_pred = knn_model.predict(X_test_transformed)
```

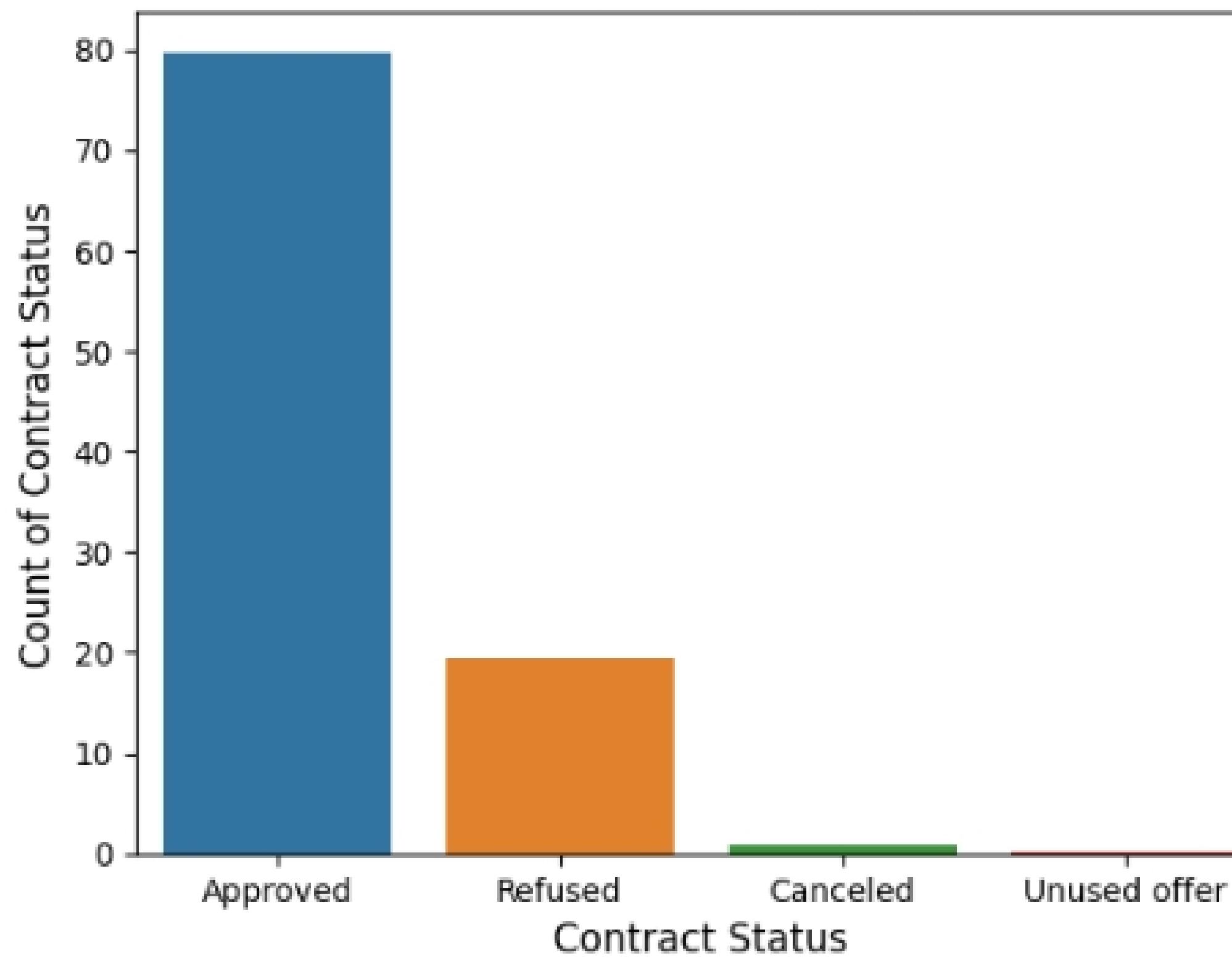
PREV APPLICATION

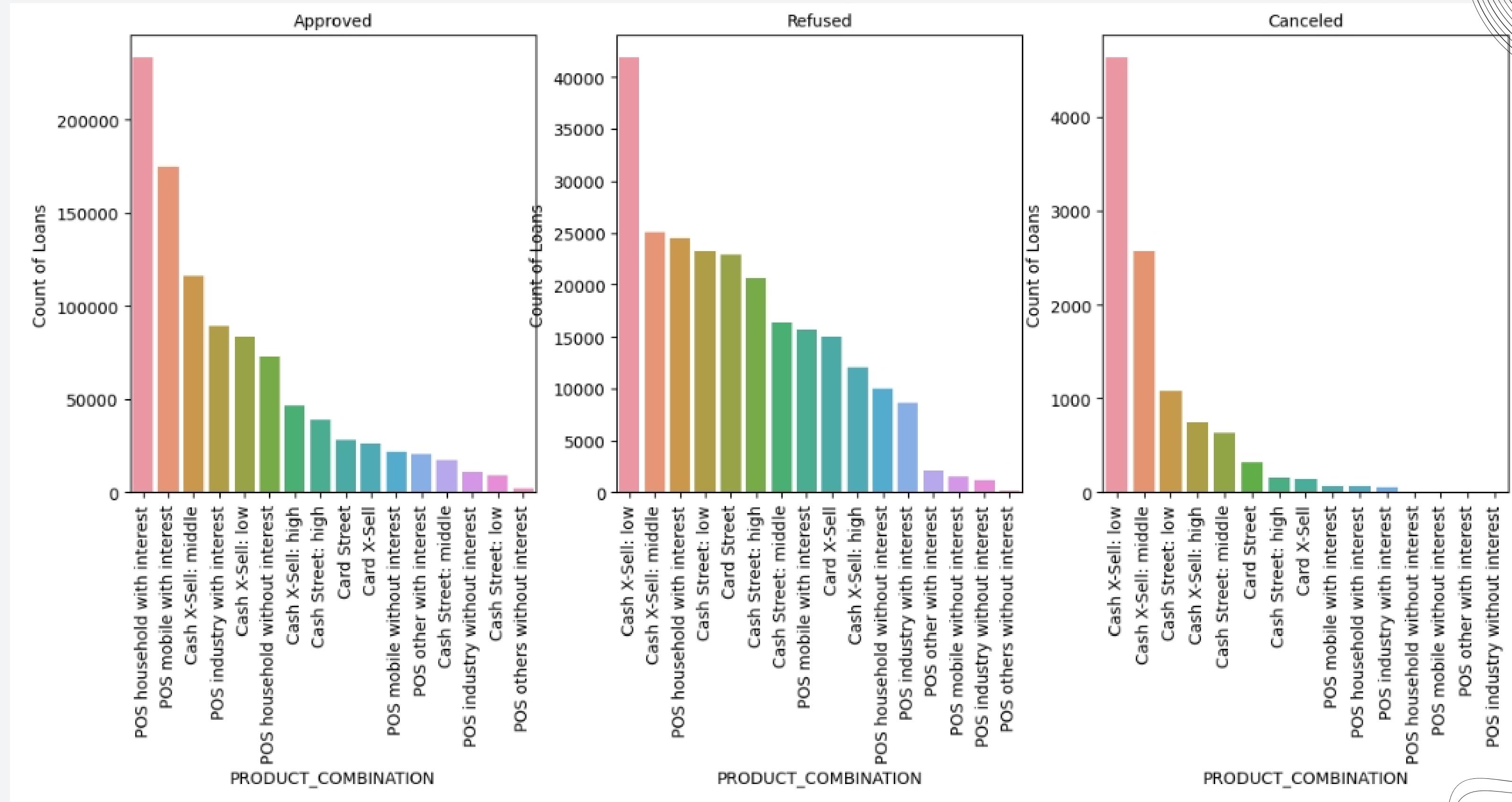


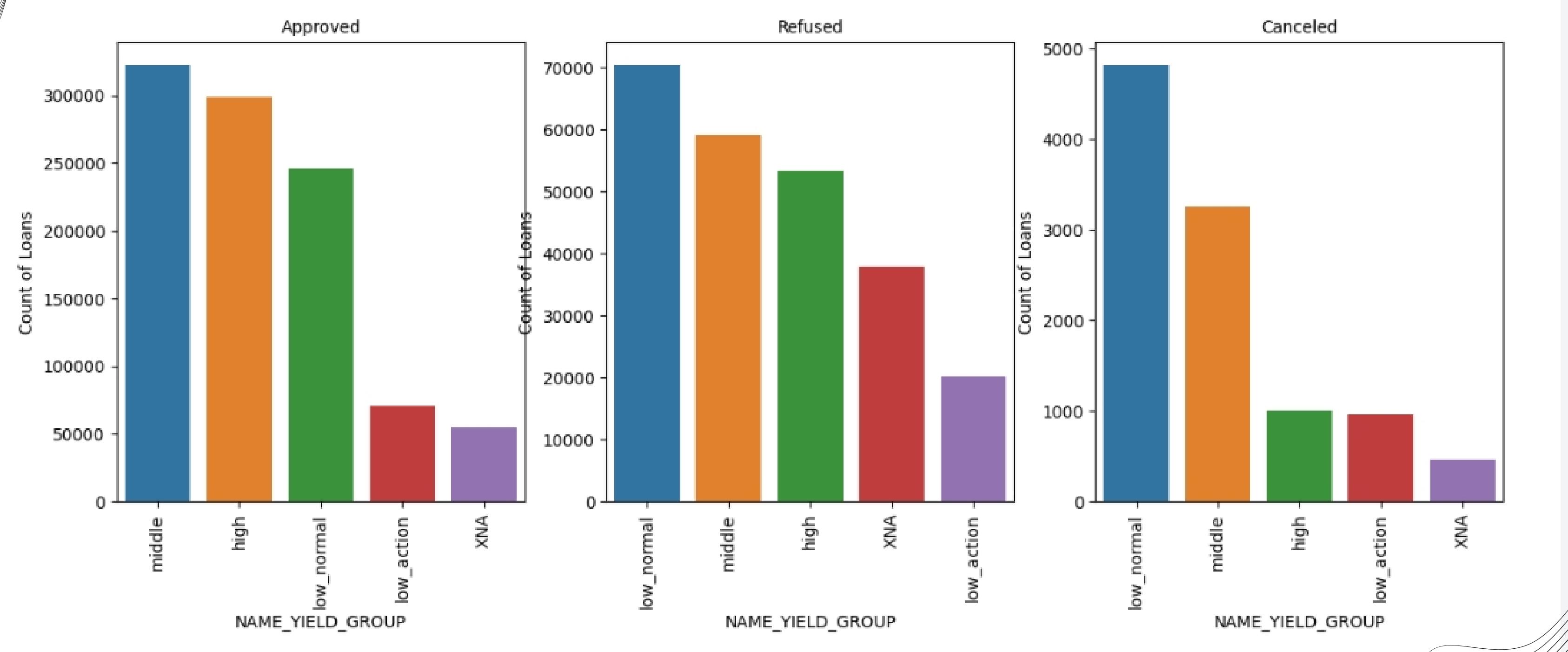
บอกความสัมพันธ์ของข้อมูล ในการทำ
credit card ครึ่งก่อนหน้า

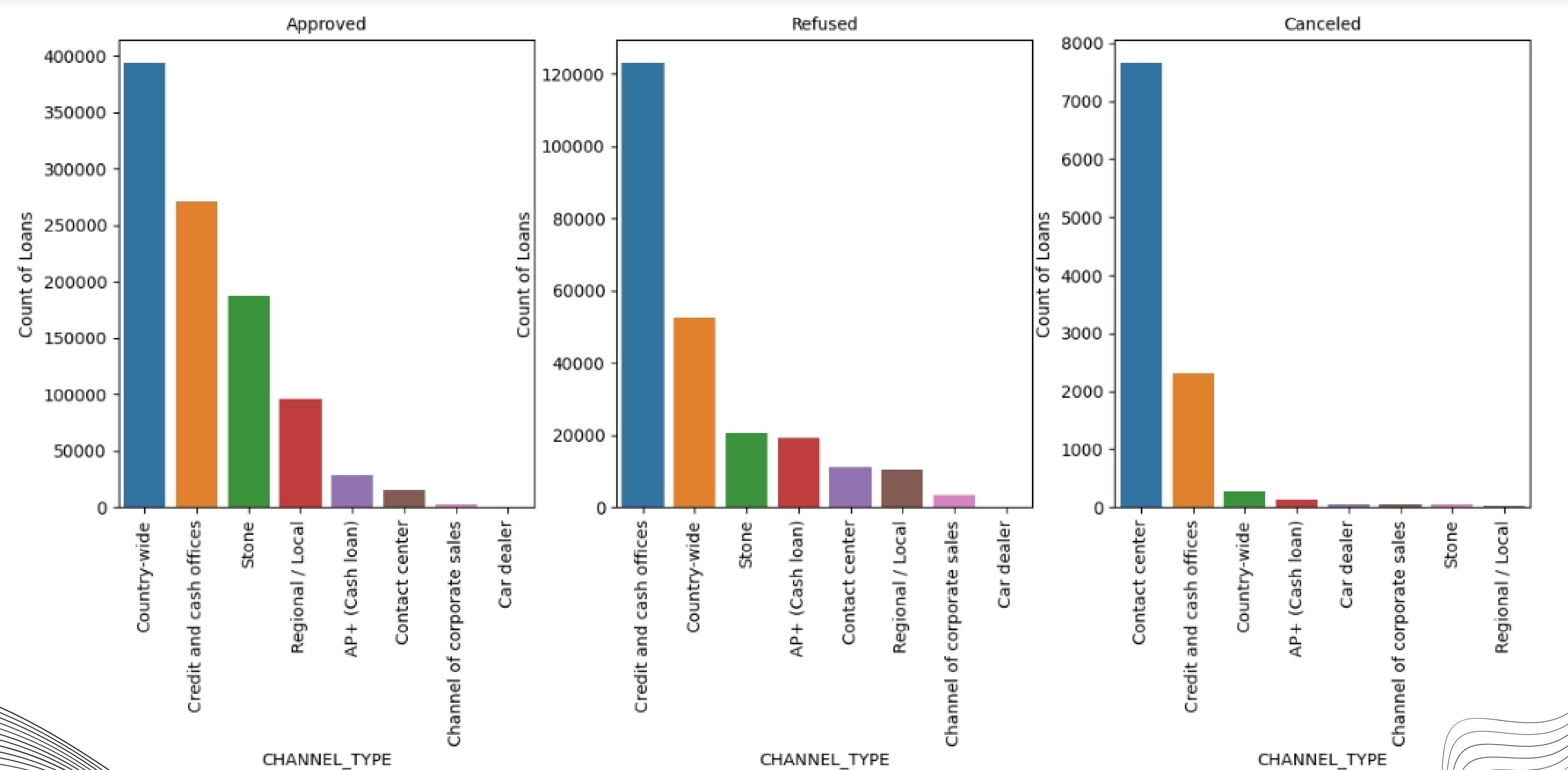


Distribution of Contract Status



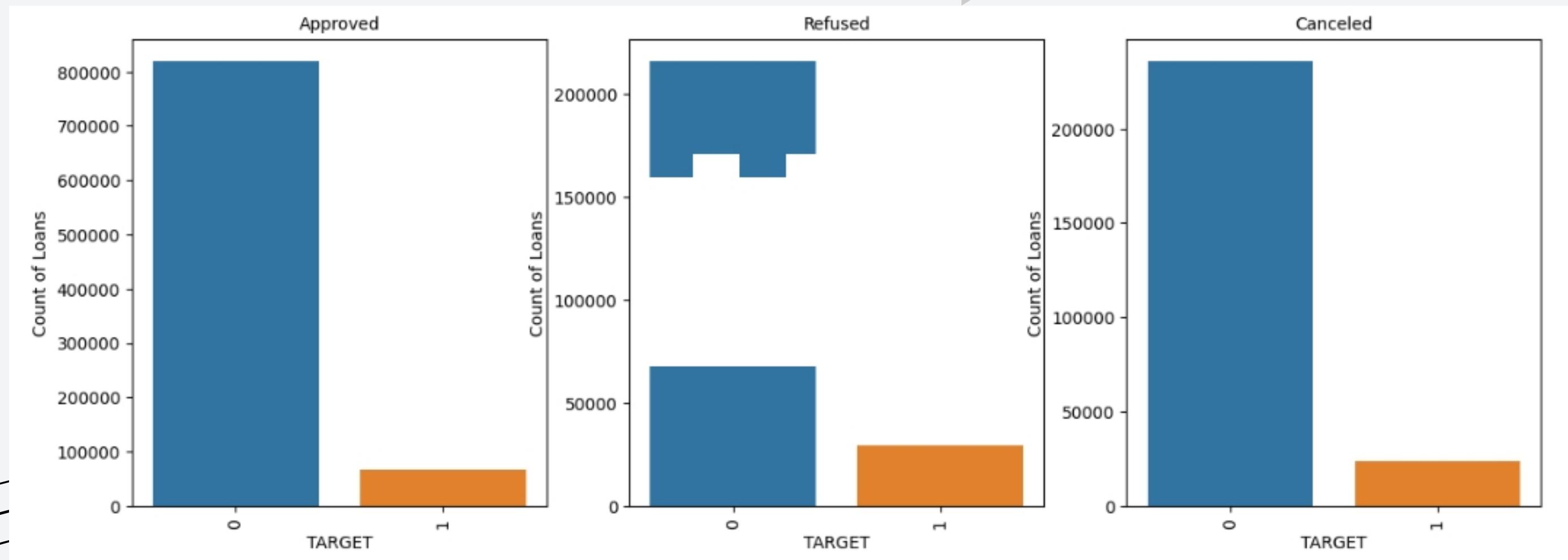


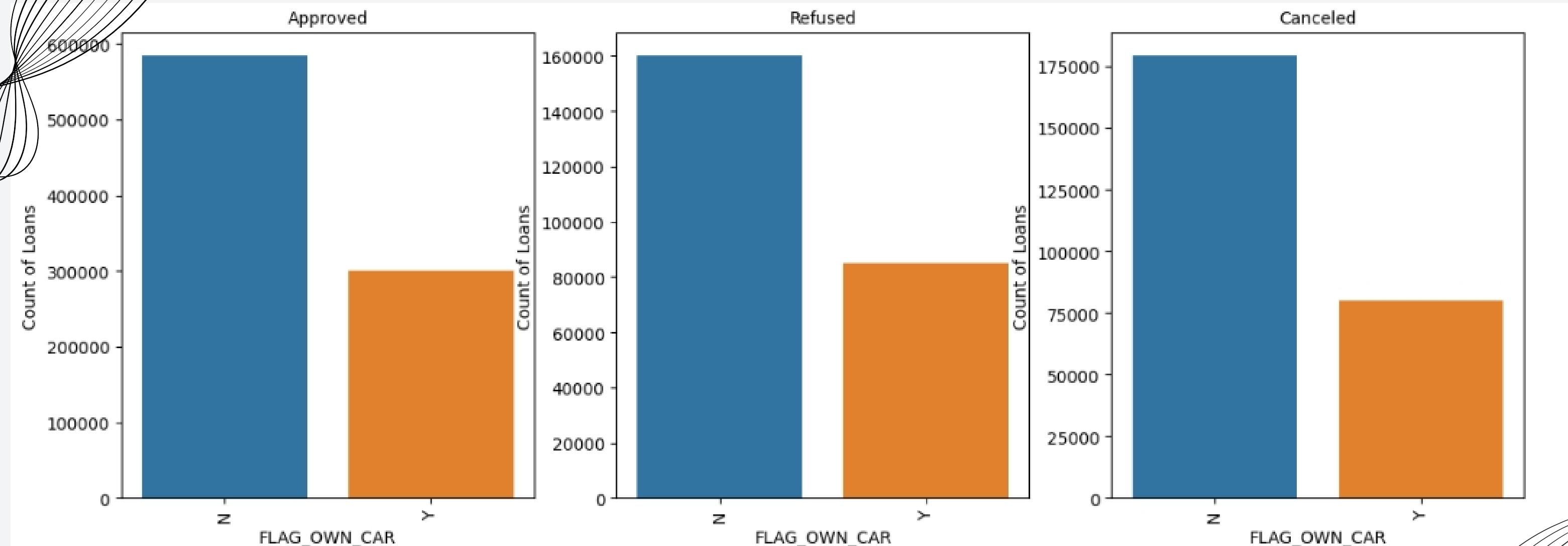


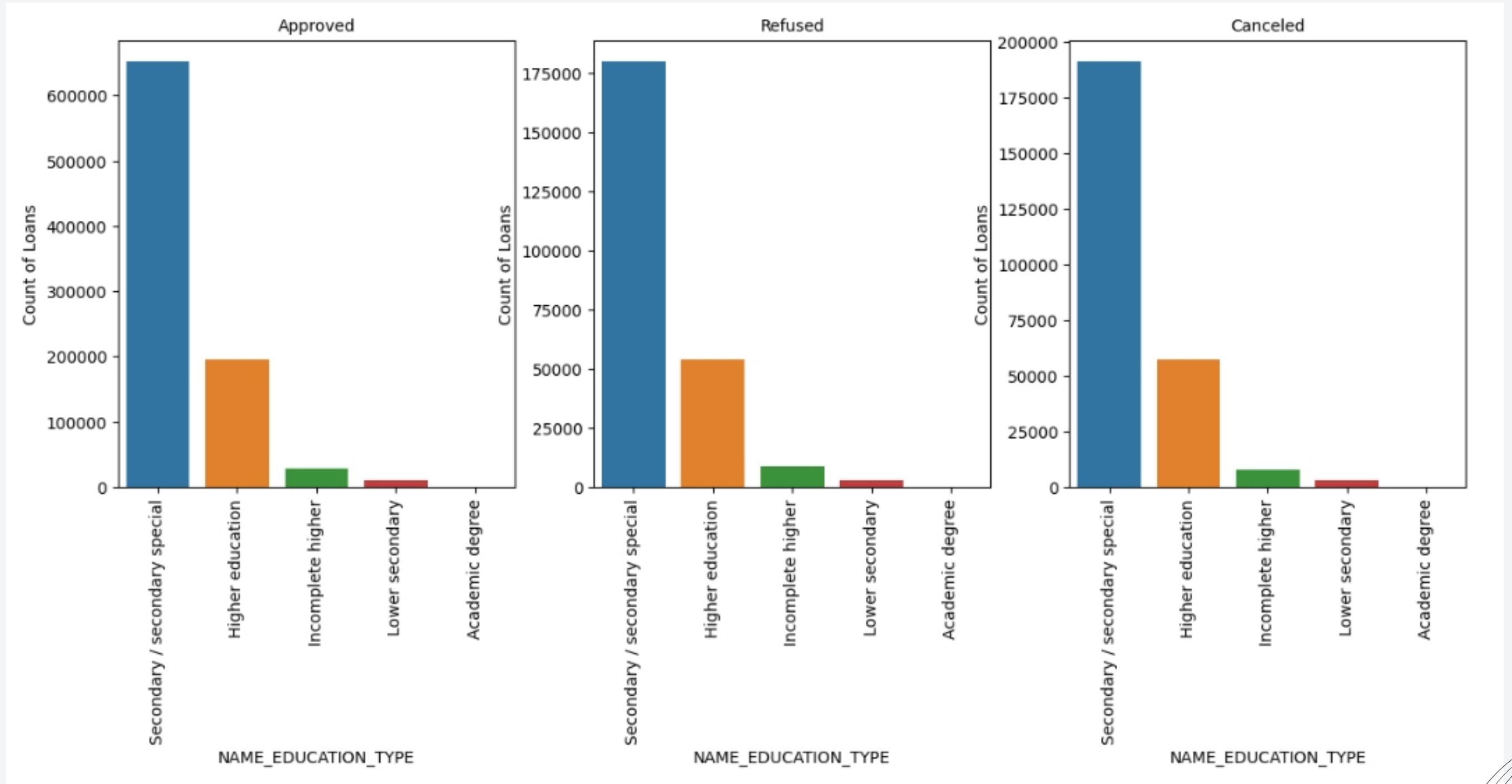


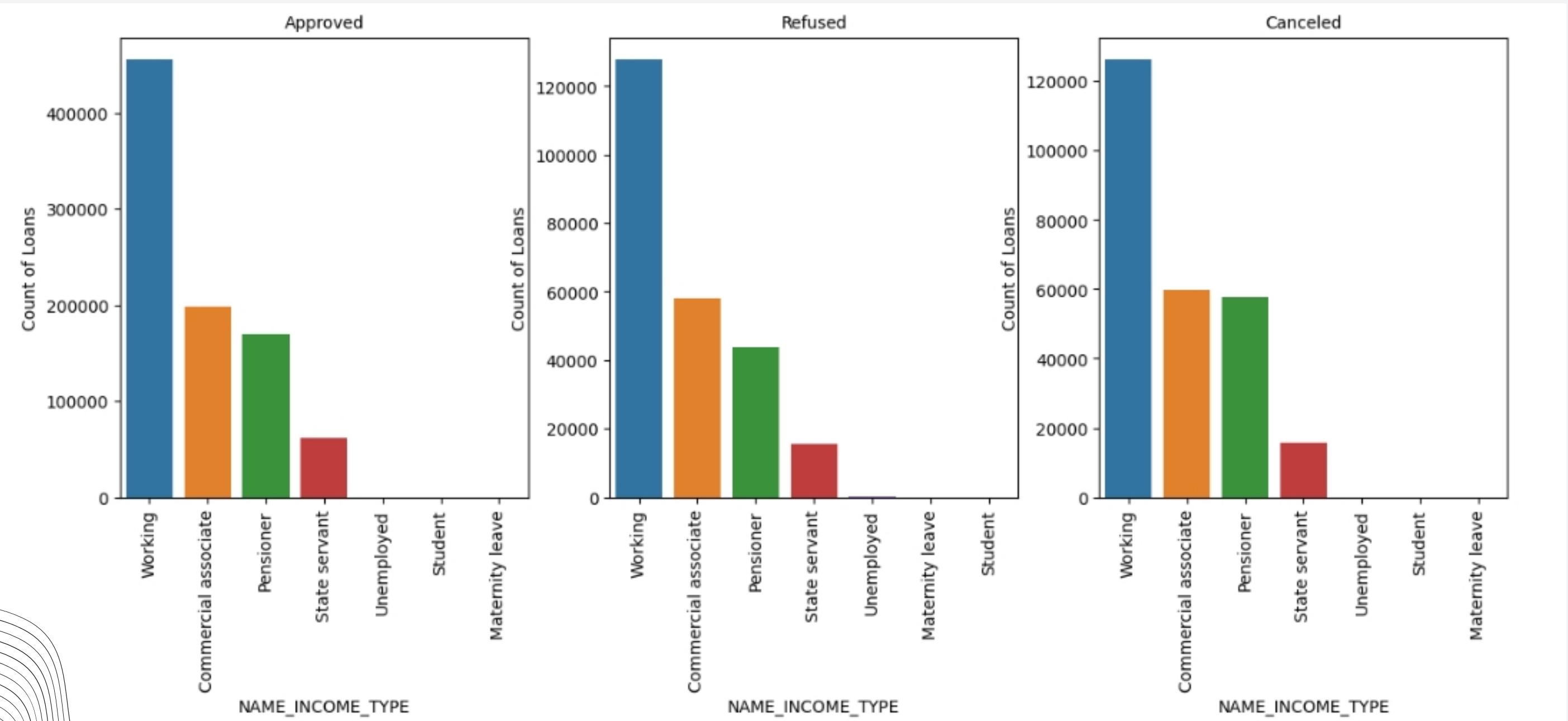
MERGE

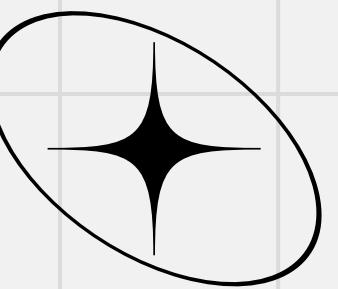
APPLICATION uses
PREVIOUS APPLICATION











MEMBER

6404053630059	วรรณดา	มวลงศ
6404053630083	เกศมนี	ปวุฒิพันธุ
6404053630130	ณัฏฐริกา	ภาคร์โยคิน
6404053630202	วรรณ	วงศ์กัตติเสนีย