WELCOME

Default of Credit Card Clients

THIS DATASET CONTAINS INFORMATION ON DEFAULT PAYMENTS, DEMOGRAPHIC FACTORS, CREDIT DATA, HISTORY OF PAYMENT, AND BILL STATEMENTS OF CREDIT CARD CLIENTS IN TAIWAN FROM APRIL 2005 TO SEPTEMBER 2005.

DATA SET INFORMATION:

This research aimed at the case of customers default payments in Taiwan, this study presented the novel Sorting Smoothing Method to estimate the real probability of default. With the real probability of default as the response variable (Y), and the predictive probability of default as the independent variable (X), **the simple linear regression result (Y = A + BX) shows that the forecasting model produced;** its regression intercept (A) is close to zero, and regression coefficient (B) to one, artificial neural network is the only one that can accurately estimate the real probability of default.

fx default payment next month

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_	PAY_	PAY_	PAY_	PAY_	PAY_	BILL_	BILL_	BILL_	BILL_	BILL_	BILL_	PAY_A	PAY_A	PAY_A	PAY_A	PAY_A	PAY_A	default payment next month
2	1	20000	2	2	1	24	2	2	-1	-1	-2	-2	3913	3102	689	0	0	0	0	689	0	0	0	0	1
3	2	120000	2	2	2	26	-1	2	0	0	0	2	2682	1725	2682	3272	3455	3261	0	1000	1000	1000	0	2000	1
4	3	90000	2	2	2	34	0	0	0	0	0	0	2923	1402	1355	1433	1494	1554	1518	1500	1000	1000	1000	5000	0
5	4	50000	2	2	1	37	0	0	0	0	0	0	4699	4823	4929	2831	2895	2954	2000	2019	1200	1100	1069	1000	0
6	5	50000	1	2	1	57	-1	0	-1	0	0	0	8617	5670	3583	2094	1914	1913	2000	36681	1000	9000	689	679	0
7	6	50000	1	1	2	37	0	0	0	0	0	0	6440	5706	5760	1939	1961	2002	2500	1815	657	1000	1000	800	0
8	7	500000	1	1	2	29	0	0	0	0	0	0	3679	4120	4450	5426	4830	4739	5500	40000	3800	2023	1375	1377	0
9	8	100000	2	2	2	23	0	-1	-1	0	0	-1	1187	380	601	221	-159	567	380	601	0	581	1687	1542	0
10	9	140000	2	3	1	28	0	0	2	0	0	0	1128	1409	1210	1221	1179	3719	3329	0	432	1000	1000	1000	0
11	10	20000	1	3	2	35	-2	-2	-2	-2	-1	-1	0	0	0	0	1300	1391	0	0	0	1300	1122	0	0
12	11	200000	2	3	2	34	0	0	2	0	0	-1	1107	9787	5535	2513	1828	3731	2306	12	50	300	3738	66	0
13	12	260000	2	1	2	51	-1	-1	-1	-1	-1	2	1226	2167	9966	8517	2228	1366	2181	9966	8583	2230	0	3640	0
14	13	630000	2	2	2	41	-1	0	-1	-1	-1	-1	1213	6500	6500	6500	6500	2870	1000	6500	6500	6500	2870	0	0
15	14	70000	1	2	2	30	1	2	2	0	0	2	6580	6736	6570	6678	3613	3689	3200	0	3000	3000	1500	0	1
16	15	250000	1	1	2	29	0	0	0	0	0	0	7088	6706	6356	5969	5687	5551	3000	3000	3000	3000	3000	3000	0
17	16	50000	2	3	3	23	1	2	0	0	0	0	5061	2917	2811	2877	2953	3021	0	1500	1100	1200	1300	1100	0
18	17	20000	1	1	2	24	0	0	2	2	2	2	1537	1801	1742	1833	1790	1910	3200	0	1500	0	1650	0	1
19	18	320000	1	1	1	49	0	0	0	-1	-1	-1	2532	2465	1946	7007	5856	1955	1035	10000	7594	2000	1955	50000	0
20	19	360000	2	1	1	49	1	-2	-2	-2	-2	-2	0	0	0	0	0	0	0	0	0	0	0	0	0
21	20	180000	2	1	2	29	1	-2	-2	-2	-2	-2	0	0	0	0	0	0	0	0	0	0	0	0	0
22	21	130000	2	3	2	39	0	0	0	0	0	-1	3835	2768	2448	2061	1180	930	3000	1537	1000	2000	930	3376	0
23	22	120000	2	2	1	39	-1	-1	-1	-1	-1	-1	316	316	316	0	632	316	316	316	0	632	316	0	1
24	23	70000	2	2	2	26	2	0	0	2	2	2	4108	4244	4502	4400	4690	4601	2007	3582	0	3601	0	1820	1
25	24	450000	2	1	1	40	-2	-2	-2	-2	-2	-2	5512	1942	1473	560	0	0	1942	1473	560	0	0	1128	1
26	25	90000	1	1	2	23	0	0	0	-1	0	0	4744	7070	0	5398	6360	8292	5757	0	5398	1200	2045	2000	0

Welcome to project!

TODAY'S CONTENTS

01

APACHE SPARK

In computing, extract, transform, load (ETL) is the general procedure of copying data from one or more sources into a destination system which represents the data differently from the source(s).

02

DASHBOARD

A type of graphical user interface which often provides at-a-glance views of key performance indicators (KPIs) relevant to a particular objective or business process and considered a form of data visualization.

EXTRACTION:

Data extraction involves extracting data from homogeneous or heterogeneous sources;

----- Extract -----

[4]:
```
sc = pyspark.SparkContext('local[*]')
```

[5]:
```
# sc.stop()
spark = SparkSession.\
        builder.\
        appName("hello pyspark").\
        master("spark://spark-master:7077").\
        config("spark.executor.memory", "512m").\
        getOrCreate()
```

[6]:
```
df = spark.read.csv(path="default of credit card clients.csv", sep=",", header=True).cache()
```

[7]:
```
df.show()
```

```
+---+---------+---+---------+--------+---+-----+-----+-----+-----+-----+-----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------------+
| ID|LIMIT_BAL|SEX|EDUCATION|MARRIAGE|AGE|PAY_0|PAY_2|PAY_3|PAY_4|PAY_5|PAY_6|BILL_AMT1|BILL_AMT2|BILL_AMT3|BILL_AMT4|BILL_AMT5|BILL_AMT6|PAY_AMT1|PAY_AMT2|PAY_AMT3|PAY_AMT4|PAY_AMT5|PAY_AMT6|default payment next month|
+---+---------+---+---------+--------+---+-----+-----+-----+-----+-----+-----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------------+
|  1|    20000|  2|        2|       1| 24|    2|    2|   -1|   -1|   -2|   -2|     3913|     3102|      689|        0|        0|        0|        0|      689|        0|        0|        0|        0|                         1|
|  2|   120000|  2|        2|       2| 26|   -1|    2|    0|    0|    0|    2|     2682|     1725|     2682|     3272|     3455|     3261|        0|     1000|     1000|     1000|        0|     2000|                         1|
|  3|    90000|  2|        2|       2| 34|    0|    0|    0|    0|    0|    0|    29239|    14027|    13559|    14331|    14948|    15549|     1518|     1500|     1000|     1000|     1000|     5000|                         0|
|  4|    50000|  2|        2|       1| 37|    0|    0|    0|    0|    0|    0|    46990|    48233|    49291|    28314|    28959|    29547|     2000|     2019|     1200|     1100|     1069|     1000|                         0|
|  5|    50000|  1|        2|       1| 57|   -1|    0|   -1|    0|    0|    0|     8617|     5670|    35835|    20940|    19146|    19131|     2000|    36681|    10000|     9000|      689|      679|                         0|
|  6|    50000|  1|        1|       2| 37|    0|    0|    0|    0|    0|    0|    64400|    57069|    57608|    19394|    19619|    20024|     2500|     1815|      657|     1000|     1000|      800|                         0|
|  7|   500000|  1|        1|       2| 29|    0|    0|    0|    0|    0|    0|   367965|   412023|   445007|   542653|   483003|   473944|    55000|    40000|    38000|    20239|    13750|    13770|                         0|
|  8|   100000|  2|        2|       2| 23|    0|   -1|   -1|    0|    0|   -1|    11876|      380|      601|      221|     -159|      567|      380|      601|        0|      581|     1687|     1542|                         0|
|  9|   140000|  2|        3|       1| 28|    0|    0|    2|    0|    0|    0|    11285|    14096|    12108|    12211|    11793|     3719|     3329|        0|      432|     1000|     1000|     1000|                         0|
| 10|    20000|  1|        3|       2| 35|   -2|   -2|   -2|   -2|   -1|   -1|        0|        0|        0|        0|    13007|    13912|        0|        0|        0|    13007|     1122|        0|                         0|
| 11|   200000|  2|        3|       2| 34|    0|    0|    2|    0|    0|   -1|    11073|     9787|     5535|     2513|     1828|     3731|     2306|       12|       50|      300|     3738|       66|                         0|
| 12|   260000|  2|        1|       2| 51|   -1|   -1|   -1|   -1|   -1|    2|    12261|    21670|     9966|     8517|    22287|    13668|    21818|     9966|     8583|    22301|        0|     3640|                         0|
| 13|   630000|  2|        2|       2| 41|   -1|    0|   -1|   -1|   -1|   -1|    12137|     6500|     6500|     6500|     6500|     2870|     1000|     6500|     6500|     6500|     2870|        0|                         0|
| 14|    70000|  1|        2|       2| 30|    1|    2|    2|    0|    0|    2|    65802|    67369|    65701|    66782|    36137|    36894|     3200|        0|     3000|     3000|     1500|        0|                         1|
| 15|   250000|  1|        1|       2| 29|    0|    0|    0|    0|    0|    0|    70887|    67060|    63561|    59696|    56875|    55512|     3000|     3000|     3000|     3000|     3000|     3000|                         0|
| 16|    50000|  2|        3|       3| 23|    1|    2|    0|    0|    0|    0|    50614|    29173|    28116|    28771|    29531|    30211|        0|     1500|     1100|     1200|     1300|     1100|                         0|
| 17|    20000|  1|        1|       2| 24|    0|    0|    2|    2|    2|    2|    15376|    18010|    17428|    18338|    17905|    19104|     3200|        0|     1500|        0|     1650|        0|                         1|
| 18|   320000|  1|        1|       1| 49|    0|    0|    0|   -1|   -1|   -1|   253286|   246536|   194663|    70074|     5856|   195599|    10358|    10000|    75940|    20000|   195599|    50000|                         0|
| 19|   360000|  2|        1|       1| 49|    1|   -2|   -2|   -2|   -2|   -2|        0|        0|        0|        0|        0|        0|        0|        0|        0|        0|        0|        0|                         0|
| 20|   180000|  2|        1|       2| 29|    1|   -2|   -2|   -2|   -2|   -2|        0|        0|        0|        0|        0|        0|        0|        0|        0|        0|        0|        0|                         0|
+---+---------+---+---------+--------+---+-----+-----+-----+-----+-----+-----+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+--------------------------+
only showing top 20 rows
```

TRANSFORMATION:

Data transformation processes data by data cleaning and transforming them into a proper storage format/structure for the purposes of querying and analysis;

----- Transform -----

Remove Column

[10]:
```
df = df.drop("PAY_0","PAY_2","PAY_3","PAY_4","PAY_5","PAY_6")
```

[11]:
```
df = df.drop("BILL_AMT1","BILL_AMT2","BILL_AMT3","BILL_AMT4","BILL_AMT5","BILL_AMT6")
```

[12]:
```
df = df.drop("PAY_AMT1","PAY_AMT2","PAY_AMT3","PAY_AMT4","PAY_AMT5","PAY_AMT6")
```

[13]:
```
df.show()
```

```
+---+---------+---+---------+--------+---+--------------------------+
| ID|LIMIT_BAL|SEX|EDUCATION|MARRIAGE|AGE|default payment next month|
+---+---------+---+---------+--------+---+--------------------------+
|  1|    20000|  2|        2|       1| 24|                         1|
|  2|   120000|  2|        2|       2| 26|                         1|
|  3|    90000|  2|        2|       2| 34|                         0|
|  4|    50000|  2|        2|       1| 37|                         0|
|  5|    50000|  1|        2|       1| 57|                         0|
|  6|    50000|  1|        1|       2| 37|                         0|
|  7|   500000|  1|        1|       2| 29|                         0|
|  8|   100000|  2|        2|       2| 23|                         0|
|  9|   140000|  2|        3|       1| 28|                         0|
| 10|    20000|  1|        3|       2| 35|                         0|
| 11|   200000|  2|        3|       2| 34|                         0|
| 12|   260000|  2|        1|       2| 51|                         0|
| 13|   630000|  2|        2|       2| 41|                         0|
| 14|    70000|  1|        2|       2| 30|                         1|
| 15|   250000|  1|        1|       2| 29|                         0|
| 16|    50000|  2|        3|       3| 23|                         0|
| 17|    20000|  1|        1|       2| 24|                         1|
| 18|   320000|  1|        1|       1| 49|                         0|
| 19|   360000|  2|        1|       1| 49|                         0|
| 20|   180000|  2|        1|       2| 29|                         0|
+---+---------+---+---------+--------+---+--------------------------+
only showing top 20 rows
```

TRANSFORMATION:

Data transformation processes data by data cleaning and transforming them into a proper storage format/structure for the purposes of querying and analysis;

----- Transform -----

Count the missing values in a column

[16]:
```
for col in df.columns:
    print(df.filter(df[col].isNull()).count())
```

```
0
0
0
0
0
0
0
0
```

SQL (Transform)

[17]:
```
df.createOrReplaceTempView("clients")
```

[42]:
```
spark.sql("SELECT * FROM clients").show()
```

```
+---+---------+---+---------+--------+---+--------------------------+
| ID|LIMIT_BAL|SEX|EDUCATION|MARRIAGE|AGE|default payment next month|
+---+---------+---+---------+--------+---+--------------------------+
|  1|    20000|  2|        2|       1| 24|                         1|
|  2|   120000|  2|        2|       2| 26|                         1|
|  3|    90000|  2|        2|       2| 34|                         0|
|  4|    50000|  2|        2|       1| 37|                         0|
|  5|    50000|  1|        2|       1| 57|                         0|
|  6|    50000|  1|        1|       2| 37|                         0|
|  7|   500000|  1|        1|       2| 29|                         0|
|  8|   100000|  2|        2|       2| 23|                         0|
|  9|   140000|  2|        3|       1| 28|                         0|
| 10|    20000|  1|        3|       2| 35|                         0|
| 11|   200000|  2|        3|       2| 34|                         0|
| 12|   260000|  2|        1|       2| 51|                         0|
| 13|   630000|  2|        2|       2| 41|                         0|
| 14|    70000|  1|        2|       2| 30|                         1|
| 15|   250000|  1|        1|       2| 29|                         0|
```

TRANSFORMATION:

Data transformation processes data by data cleaning and transforming them into a proper storage format/structure for the purposes of querying and analysis;

----- Transform -----

```
Pandas (Transform)
```

[19]:
```
df= spark.sql("SELECT * FROM clients").toPandas()
```

[21]:
```
df.columns
```

[21]:
```
Index(['ID', 'LIMIT_BAL', 'SEX', 'EDUCATION', 'MARRIAGE', 'AGE',
       'default payment next month'],
      dtype='object')
```

[22]:
```
df.head()
```

[22]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	default payment next month
0	1	20000	2	2	1	24	1
1	2	120000	2	2	2	26	1
2	3	90000	2	2	2	34	0
3	4	50000	2	2	1	37	0
4	5	50000	1	2	1	57	0

[20]:
```
df.tail()
```

[20]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	default payment next month
29995	29996	220000	1	3	1	39	0
29996	29997	150000	1	3	2	43	0
29997	29998	30000	1	2	2	37	1
29998	29999	80000	1	3	1	41	1
29999	30000	50000	1	2	1	46	1

TRANSFORMATION:

Data transformation processes data by data cleaning and transforming them into a proper storage format/structure for the purposes of querying and analysis;

----- Transform -----

[22]:
```
df.head()
```

[22]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	default payment next month
0	1	20000	2	2	1	24	1
1	2	120000	2	2	2	26	1
2	3	90000	2	2	2	34	0
3	4	50000	2	2	1	37	0
4	5	50000	1	2	1	57	0

[23]:
```
df['SEX'].replace({'1':'Male','2':'Female'}, inplace=True)
```

[24]:
```
df['EDUCATION'].replace({'1':'Graduate school','2':'University','3':'High school','4':'others','5':'unknown','6':'unknown','0':'unknown'}, inplace=True)
```

[25]:
```
df['MARRIAGE'].replace({'1':'Married','2':'Single','3':'others','0':'unknown'}, inplace=True)
```

[26]:
```
df['default payment next month'].replace({'1':'Yes','0':'No'}, inplace=True)
```

[43]:
```
df.head()
```

[43]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	default payment next month
0	1	20000	Female	University	Married	24	Yes
1	2	120000	Female	University	Single	26	Yes
2	3	90000	Female	University	Single	34	No
3	4	50000	Female	University	Married	37	No
4	5	50000	Male	University	Married	57	No

LOADING:

Data loading describes the insertion of data into the final target database such as an operational data store, a data mart, data lake or a data warehouse.

----- Load -----

[28]:
```
from sqlalchemy import create_engine
```

[29]:
```
username = "root"
password = "dstb"
port = 3306
database = "dstb_db"
```

[30]:
```
pymysql.install_as_MySQLdb()
```

[31]:
```
engine = create_engine('mysql+mysqldb://%s:%s@db:%i/%s'%(username, password, port, database))
```

[32]:
```
df.to_sql("clients", engine, if_exists="replace")
```

```
phpMyAdmin - > http://localhost:8000
```

```
Query from MariaDB
```

[33]:
```
%load_ext sql
```

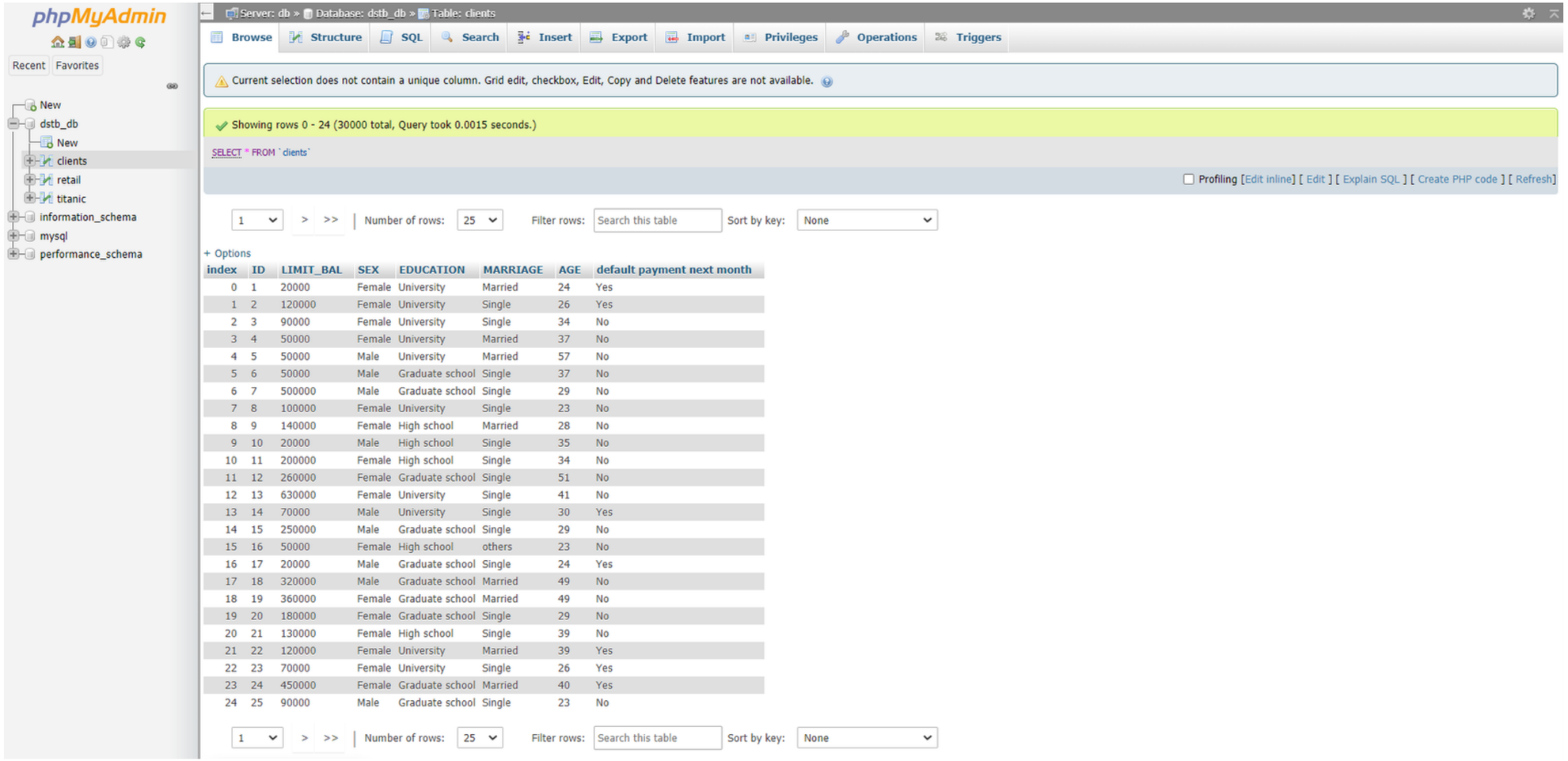
[34]:
```
%sql mysql://root:dstb@db:3306/dstb_db
```

[35]:
```
result = %sql select * from clients
```

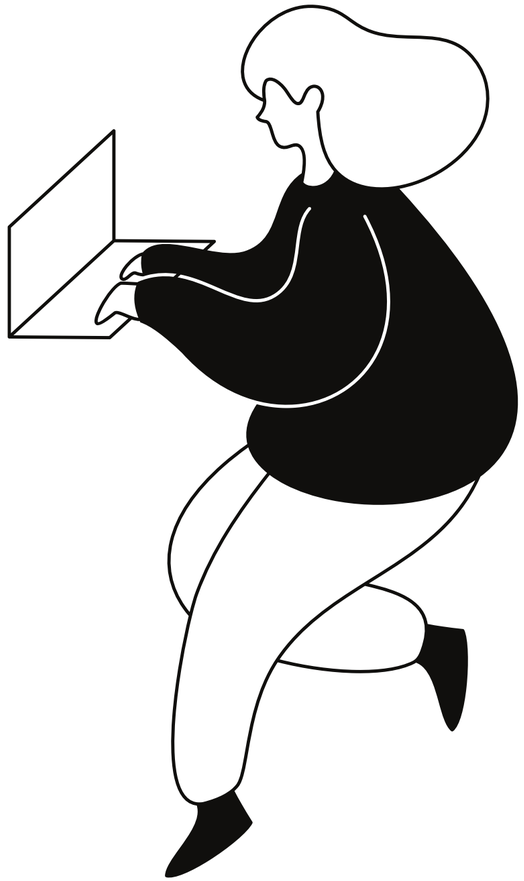
```
 * mysql://root:***@db:3306/dstb_db
30000 rows affected.
```

LOADING:

Data loading describes the insertion of data into the final target database such as an operational data store, a data mart, data lake or a data warehouse.



index	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	default payment next month
0	1	20000	Female	University	Married	24	Yes
1	2	120000	Female	University	Single	26	Yes
2	3	90000	Female	University	Single	34	No
3	4	50000	Female	University	Married	37	No
4	5	50000	Male	University	Married	57	No
5	6	50000	Male	Graduate school	Single	37	No
6	7	500000	Male	Graduate school	Single	29	No
7	8	100000	Female	University	Single	23	No
8	9	140000	Female	High school	Married	28	No
9	10	20000	Male	High school	Single	35	No
10	11	200000	Female	High school	Single	34	No
11	12	260000	Female	Graduate school	Single	51	No
12	13	630000	Female	University	Single	41	No
13	14	70000	Male	University	Single	30	Yes
14	15	250000	Male	Graduate school	Single	29	No
15	16	50000	Female	High school	others	23	No
16	17	20000	Male	Graduate school	Single	24	Yes
17	18	320000	Male	Graduate school	Married	49	No
18	19	360000	Female	Graduate school	Married	49	No
19	20	180000	Female	Graduate school	Single	29	No
20	21	130000	Female	High school	Single	39	No
21	22	120000	Female	University	Married	39	Yes
22	23	70000	Female	University	Single	26	Yes
23	24	450000	Female	Graduate school	Married	40	Yes
24	25	90000	Male	Graduate school	Single	23	No



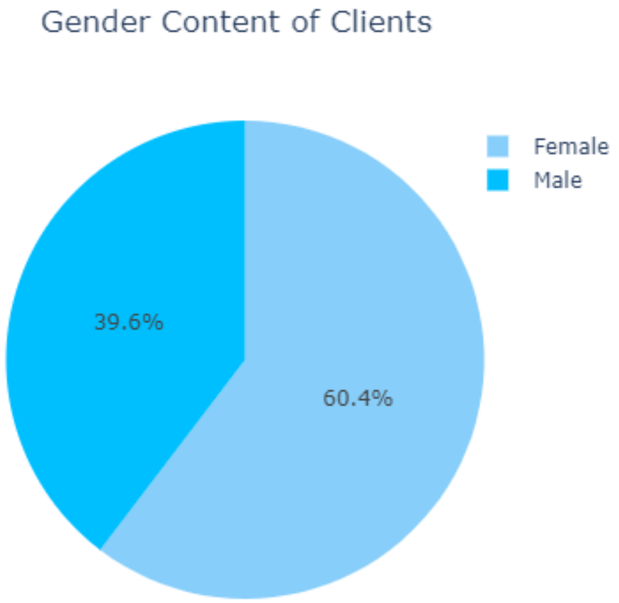
NEXT

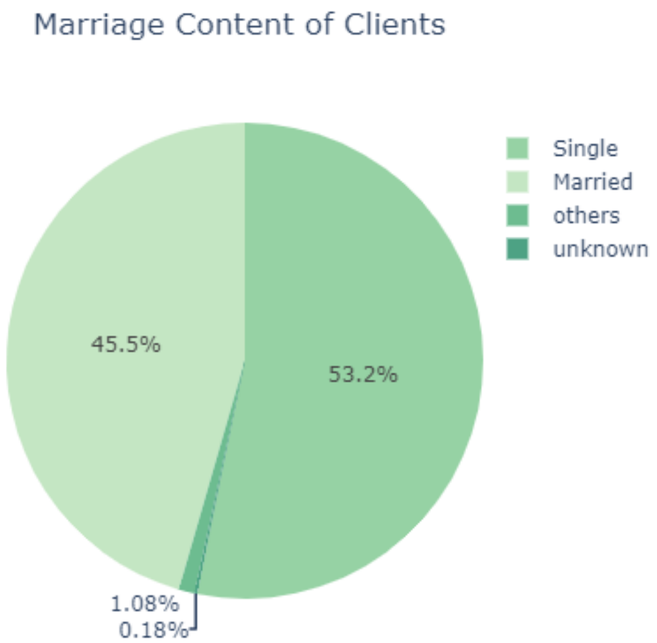
Dashboard

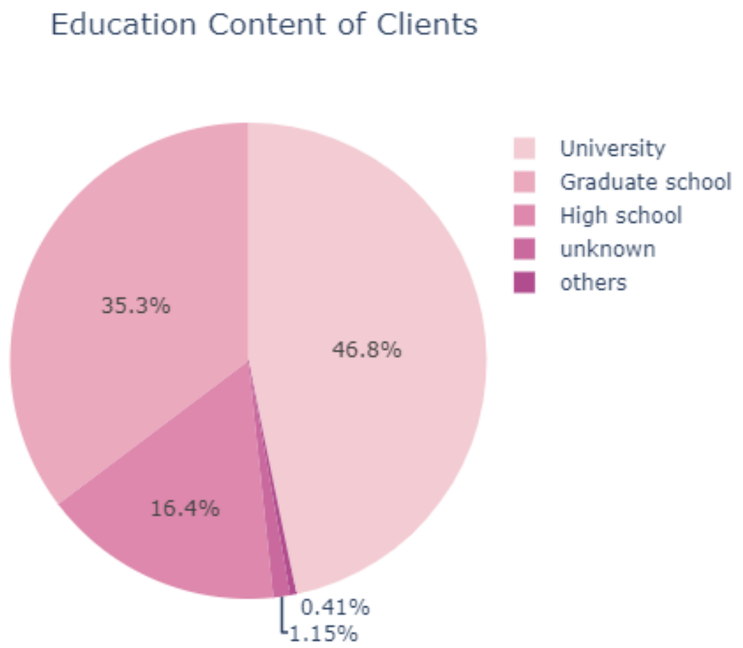
OF DEFAULT OF CREDIT CARD CLIENTS

In other usage, "dashboard" is another name for "progress report" or "report" and considered a form of data visualization.The “dashboard” is often accessible by a web browser and is usually linked to regularly updating data sources.

CREDIT CARD CLIENTS DASHBOARD







Most of our clients are

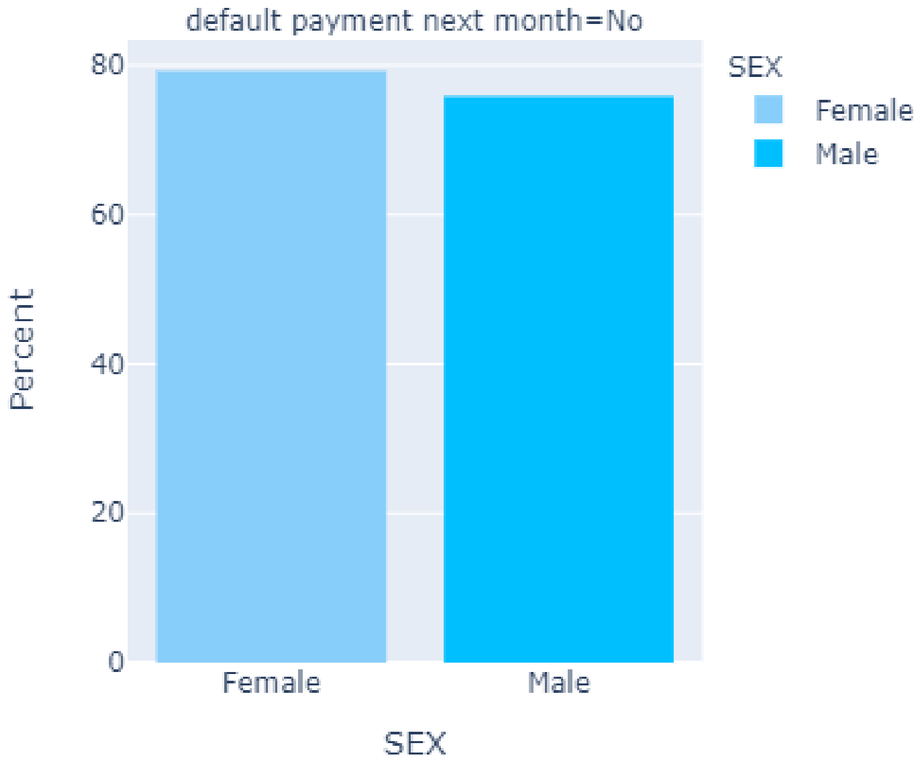
FEMALE

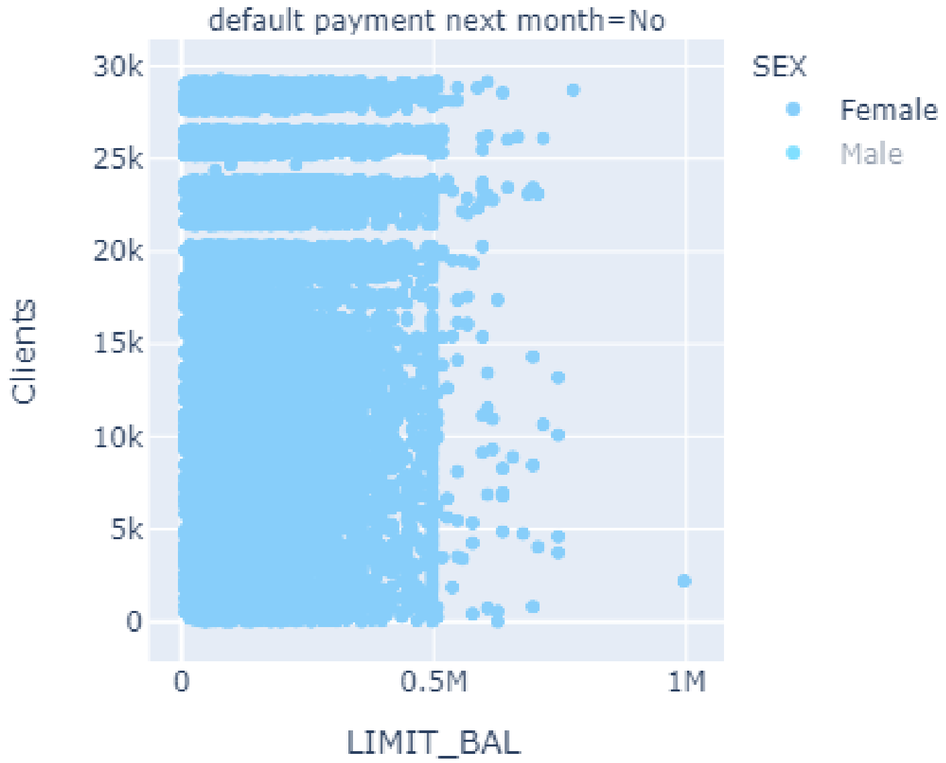
SINGLE

UNIVERSITY

DEFAULT PAYMENT NEXT MONTH = NO

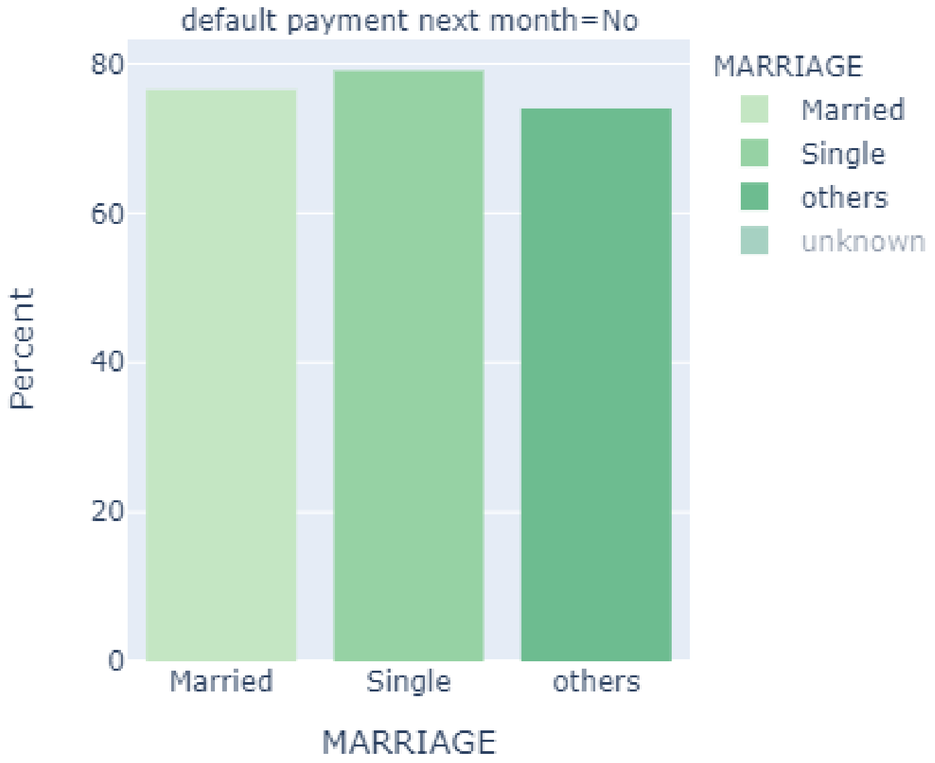
GENDER CONTENT OF CLIENTS

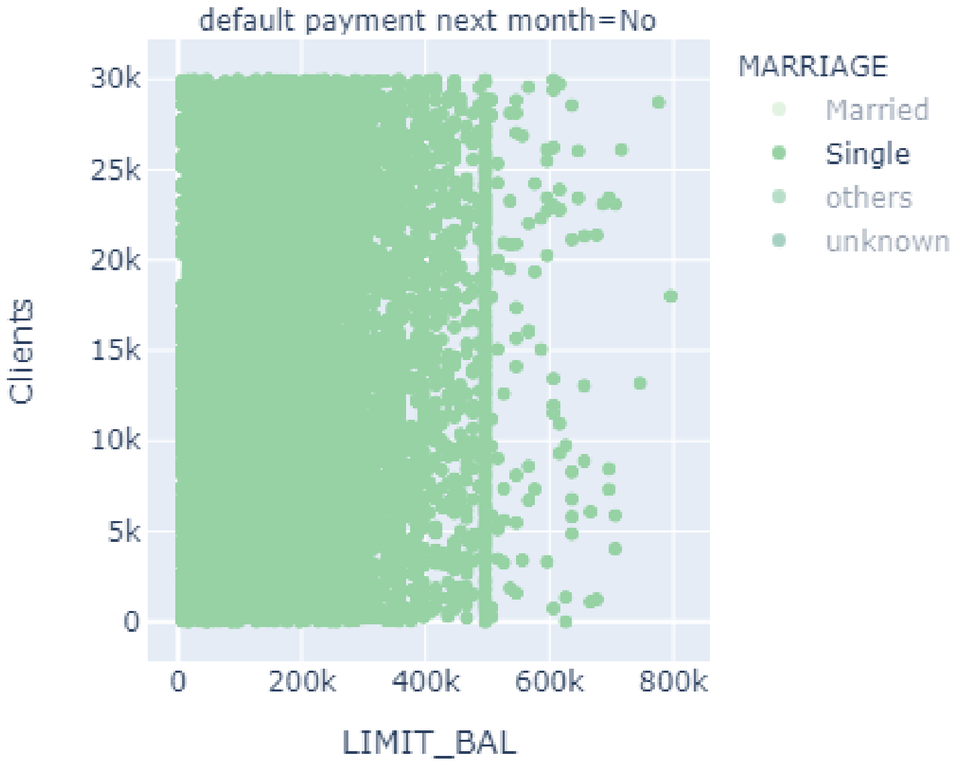




DEFAULT PAYMENT NEXT MONTH = NO

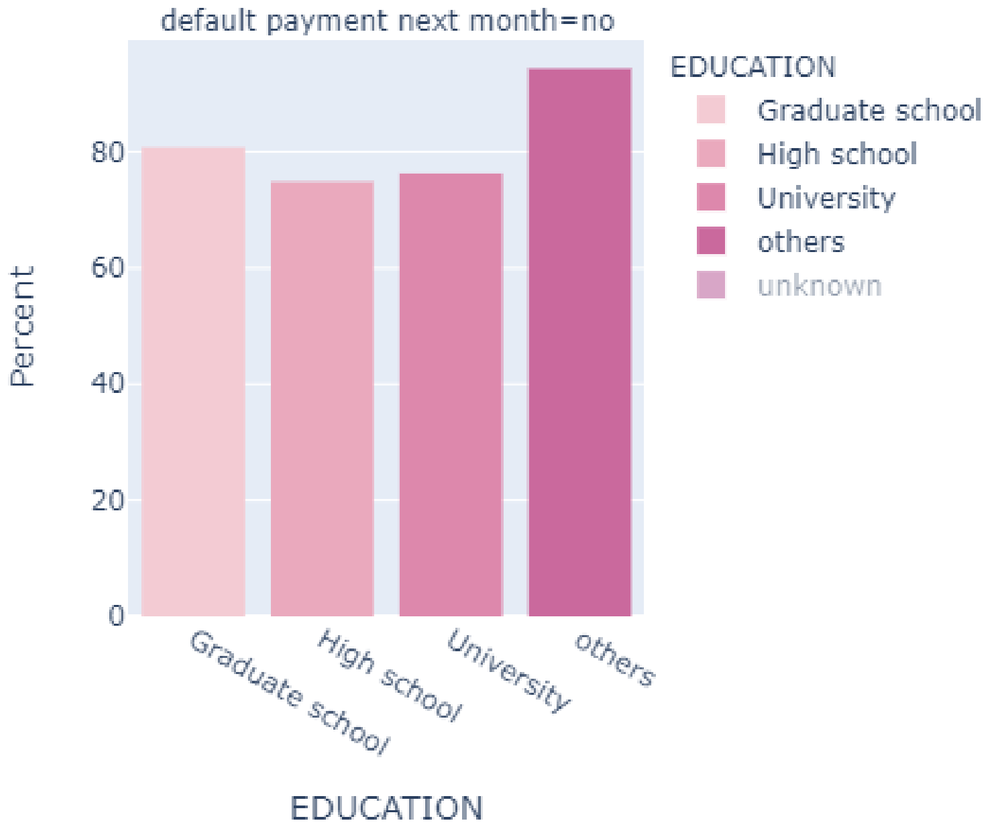
MARRIAGE CONTENT OF CLIENTS

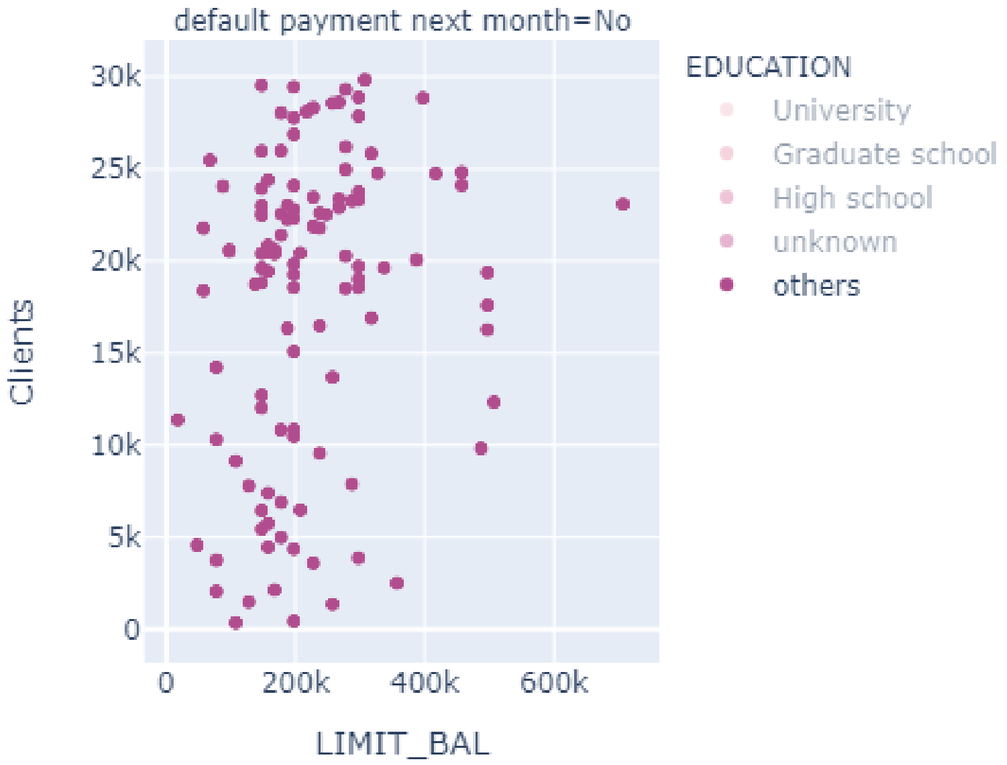




DEFAULT PAYMENT NEXT MONTH = NO

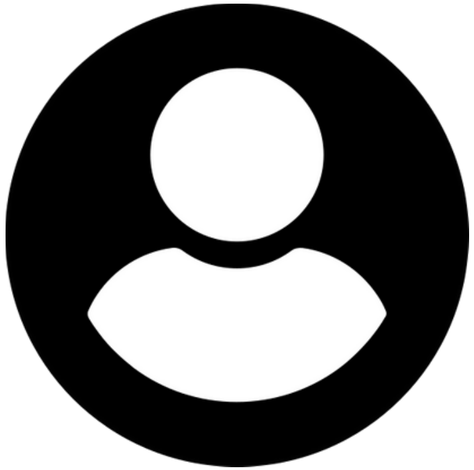
EDUCATION CONTENT OF CLIENTS





Who not default payment next month?

FEMALE

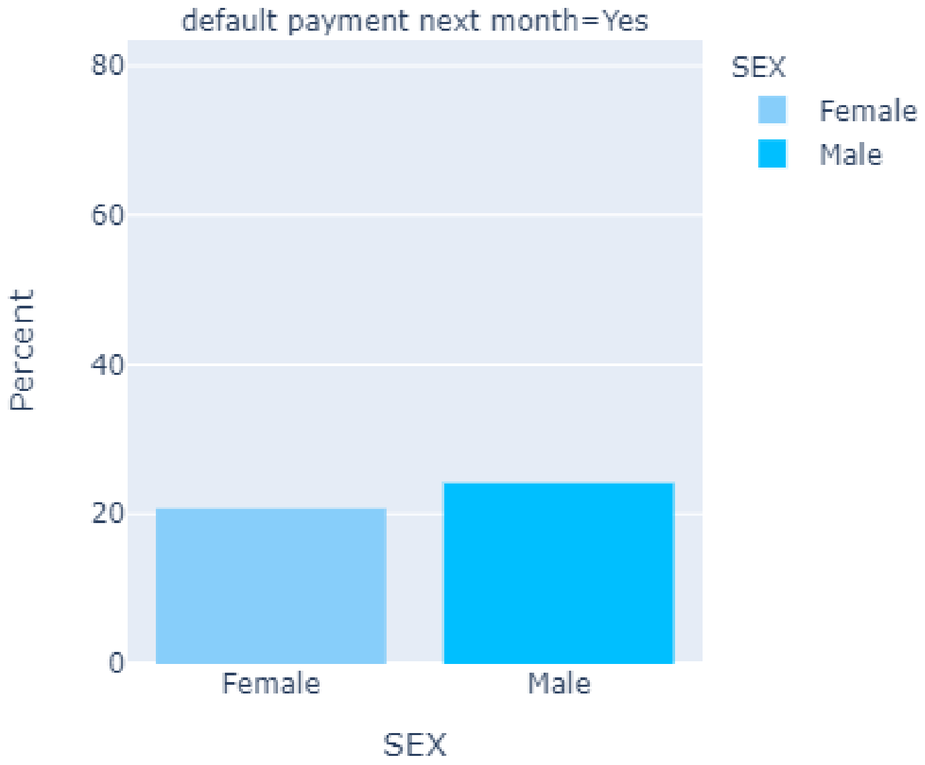


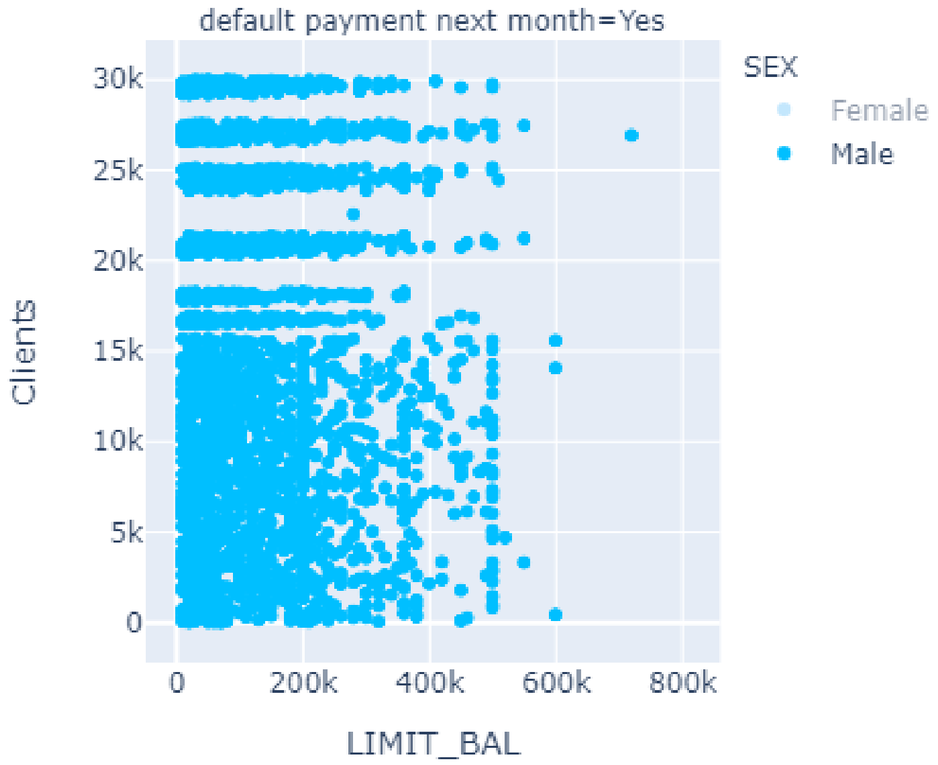
SINGLE

OTHER

DEFAULT PAYMENT NEXT MONTH = YES

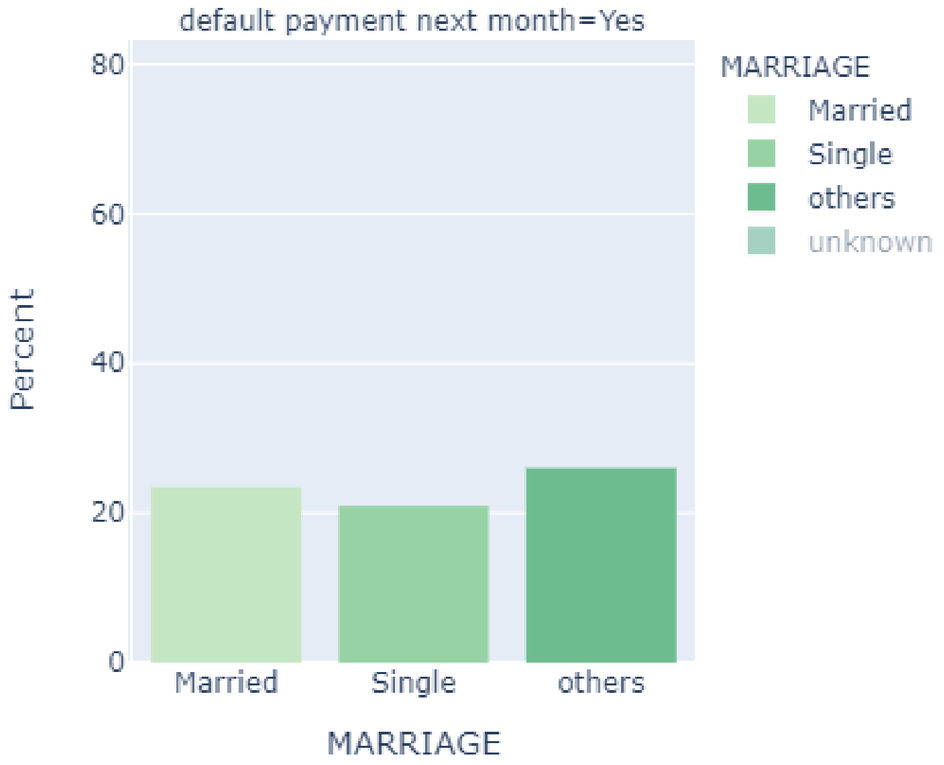
GENDER CONTENT OF CLIENTS

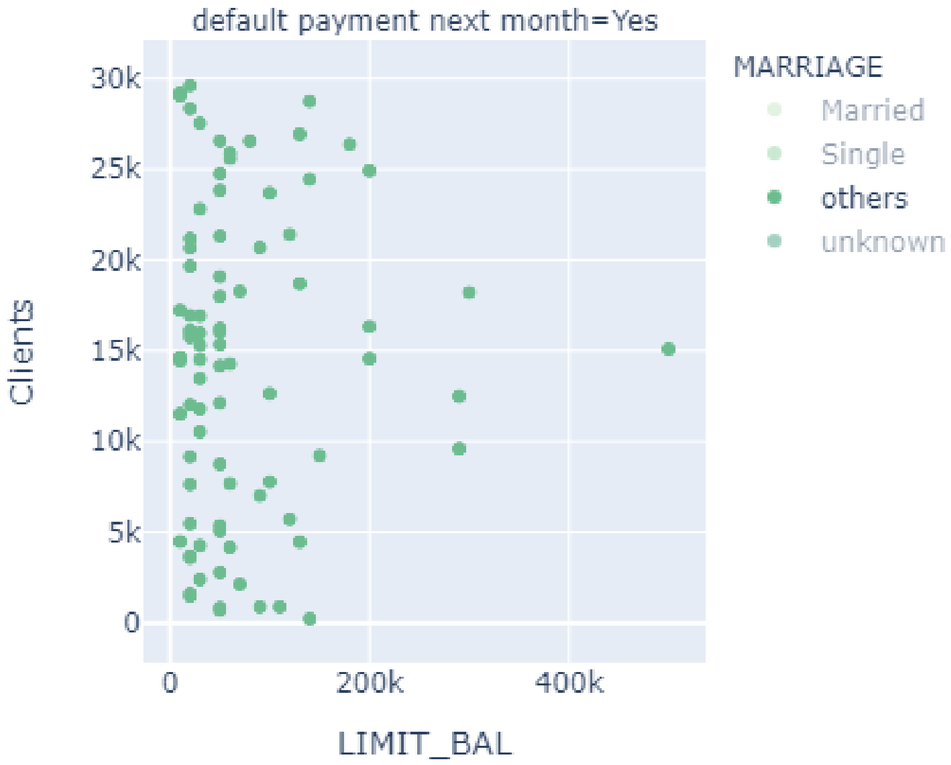




DEFAULT PAYMENT NEXT MONTH = YES

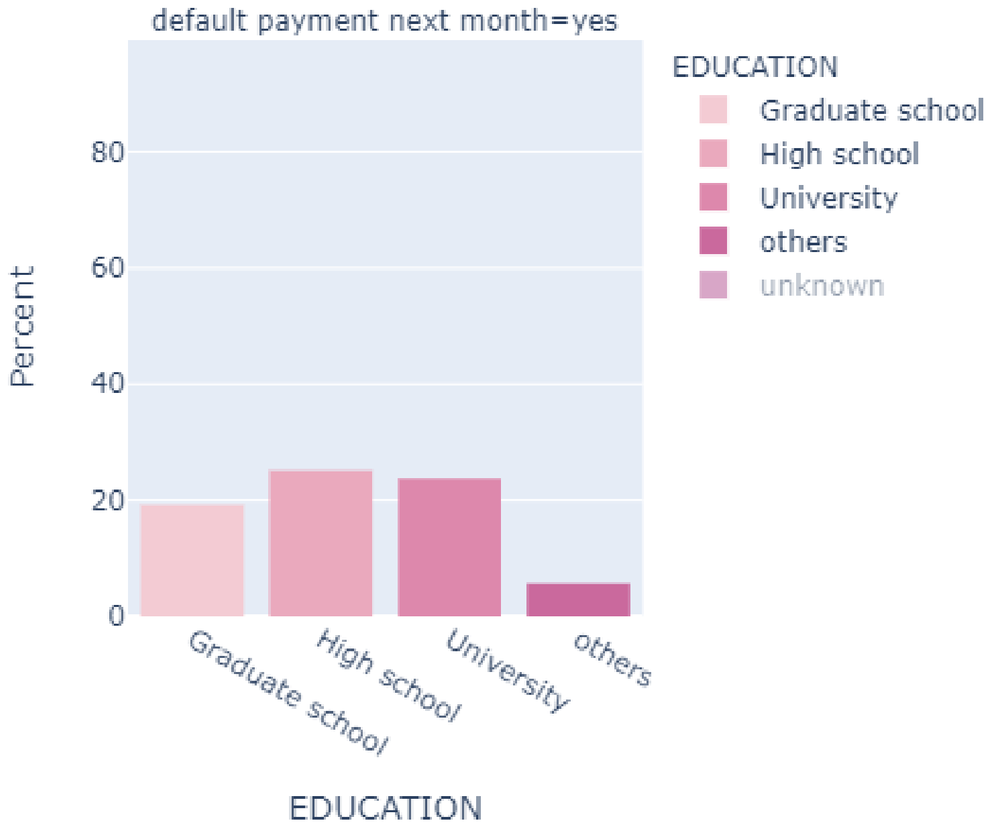
MARRIAGE CONTENT OF CLIENTS

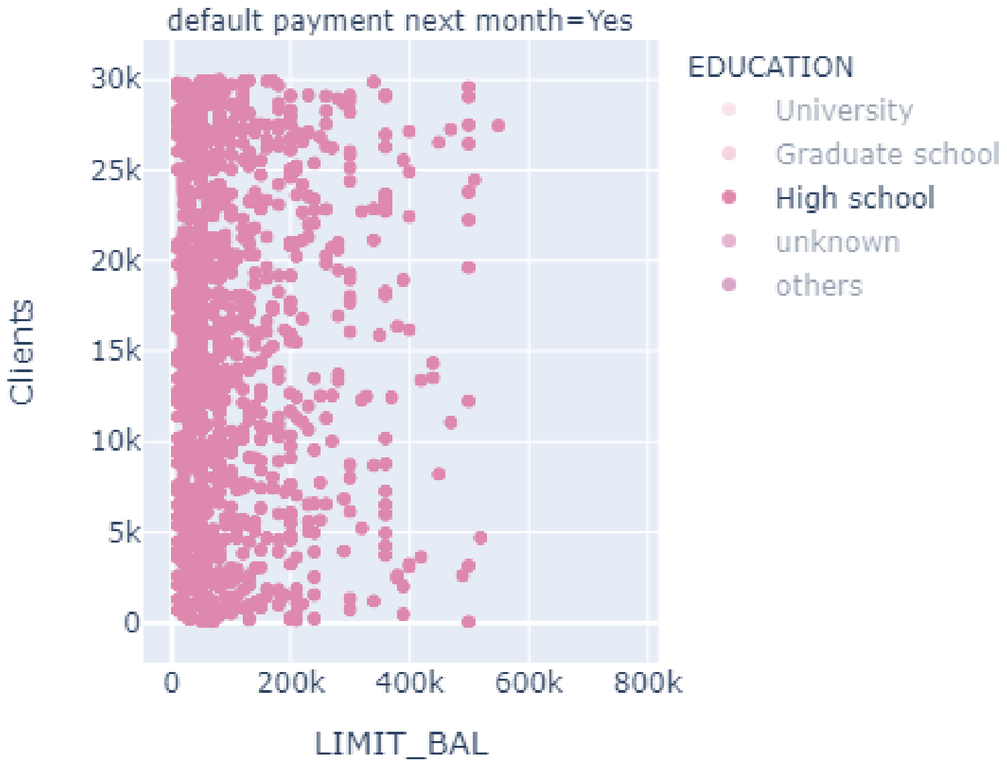




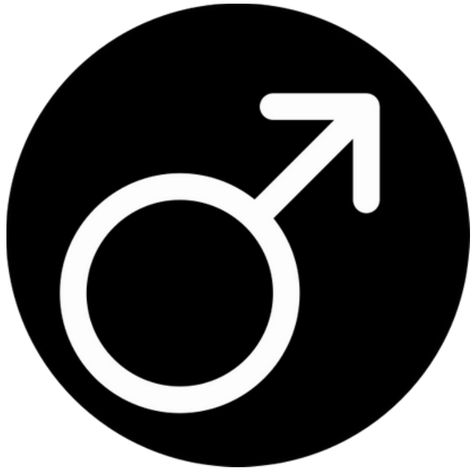
DEFAULT PAYMENT NEXT MONTH = YES

EDUCATION CONTENT OF CLIENTS





Who default payment next month?



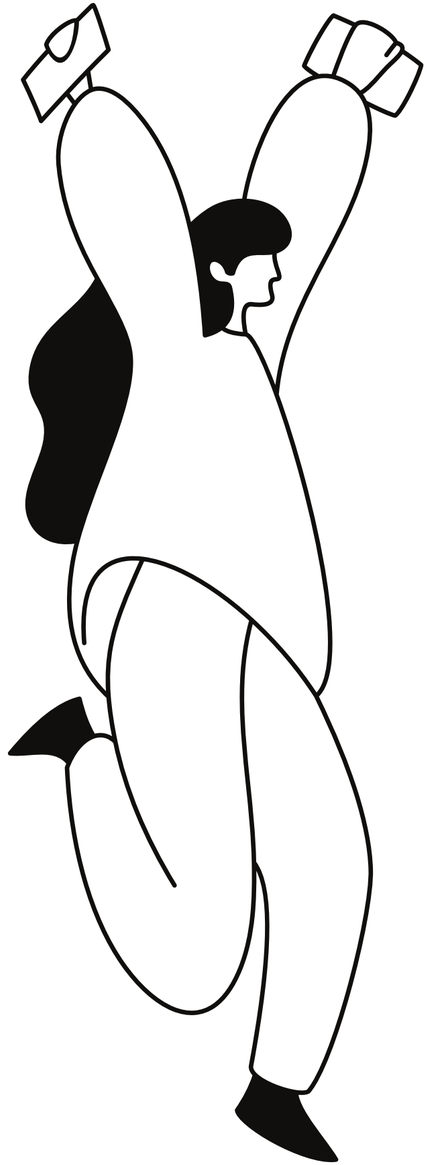
MALE

OTHER

HIGH SCHOOL

Marketing Plan

Well that’s it from me. Thanks very much.



PRESENTED BY

620710405 นางสาวณัฐธิดา ลาภธนชัย
620710745 นางสาวอาทิตยา ชมทอง