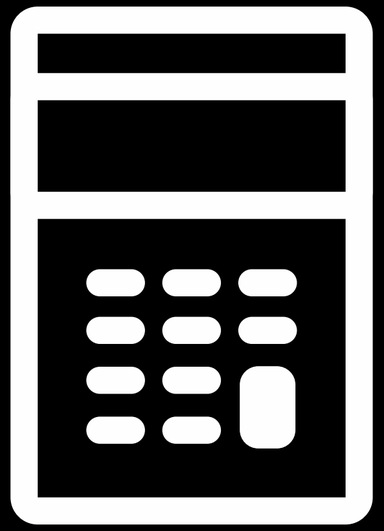
Credit Card Approval Prediction (machine learning project)

Presented to : Prof.Claudio Sartori
Presenter: Samral Tahirli

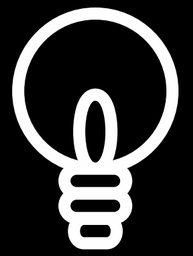
What is credit card approval prediction ?



Credit score cards are a common risk control method. It uses personal information and data submitted by credit card applicants to predict the probability of future defaults and credit card borrowings.Credit score cards are based on historical data.Vintage Analysis is one of the most popular ones.

What about we will talk today?

1	2	3	4	5
Step	**Step**	**Step**	**Step**	**Step**
Our task and Vintage analysis	Project data sets	Feature Engineering	Data analyzing	Machine learning



1

Task overview

We need build a machine learning model to predict if an applicant is 'good' or 'bad' client, different from other tasks, the definition of 'good' or 'bad' is not given. You should use some techique, such as Vintage analysis to construct our label. Also, unbalance data problem is a big problem in this task.

Vintage analysis

Vintage analysis measures the performance of a portfolio in different periods of time after the loan (or credit card) was granted. Performance can be measured in the form of cumulative charge-off rate, proportion of customers 30/60/90 days past due (DPD), utilization ratio, average balance etc.

2

Project data sets

We have 2 different data sets - Application record and Credit records.Both data sets contains duplicates values.

Application record keeps personal information while Credit records data set keeps credit records.

```
ar.head()
```

	ID	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_T
0	5008804	M	Y	Y	0	427
1	5008805	M	Y	Y	0	427
2	5008806	M	Y	Y	0	112
3	5008808	F	N	Y	0	270
4	5008809	F	N	Y	0	270

```
[5] ar.shape

(438557, 18)
```

```
cr.head()
```

	ID	STATUS
418975	5050603	1
832451	5117313	1
979818	5142362	1
947078	5135636	1
810410	5116036	1

```
[10] cr.shape

(1048575, 3)
```

3

Feature Engineering

When we check intersection of 2 datasets ,we can see that there are only 36457 rows are in both dataset.

30% of "OCCUPATION_TYPE" of application record data sets are null and we fill it while there is no any null values in credit record dataset.

```
[15] len(set(cr['ID']).intersection(set(ar['ID'])))

     36457

     ar.isnull().sum()/ar.shape[0]*100

     ID                     0.000000
     CODE_GENDER            0.000000
     FLAG_OWN_CAR           0.000000
     FLAG_OWN_REALTY        0.000000
     CNT_CHILDREN           0.000000
     AMT_INCOME_TOTAL       0.000000
     NAME_INCOME_TYPE       0.000000
     NAME_EDUCATION_TYPE    0.000000
     NAME_FAMILY_STATUS     0.000000
     NAME_HOUSING_TYPE      0.000000
     DAYS_BIRTH             0.000000
     DAYS_EMPLOYED          0.000000
     FLAG_MOBIL             0.000000
     FLAG_WORK_PHONE        0.000000
     FLAG_PHONE             0.000000
     FLAG_EMAIL             0.000000
     OCCUPATION_TYPE       30.601039
     CNT_FAM_MEMBERS        0.000000
     dtype: float64
```

3

Feature Engineering (application record)

We can see that 47 rows are duplicated in application data set and we remove them.

We need to change data type of some columns such as "FLAG_PHONE","CODE_GENDER" and etc

We create new columns such as "AGE" and "YEARS_EMPLOYED" show that client's age and how many years he/she works.Then we dropped "DAYS_BIRTH","DAYS_EMPLOYED" columns.

```
ar['FLAG_PHONE']=ar['FLAG_PHONE'].astype(str)
ar['FLAG_EMAIL']=ar['FLAG_EMAIL'].astype(str)
ar['FLAG_WORK_PHONE']=ar['FLAG_WORK_PHONE'].astype(str)
ar['CNT_FAM_MEMBERS']=ar['CNT_FAM_MEMBERS'].astype(int)
ar['FLAG_OWN_CAR']=ar['FLAG_OWN_CAR'].astype(str)
ar['CODE_GENDER']=ar['CODE_GENDER'].astype(str)
ar['FLAG_OWN_CAR']=ar['FLAG_OWN_CAR'].astype(str)
ar['FLAG_OWN_REALTY']=ar['FLAG_OWN_REALTY'].astype(str)

ar['AGE'] = ar['DAYS_BIRTH'].apply(lambda x: round(abs(x/365)))
ar.drop('DAYS_BIRTH', axis=1, inplace=True)
ar['YEARS_EMPLOYED'] = ar['DAYS_EMPLOYED'].apply(lambda x: round(abs(x/365),
ar.drop('DAYS_EMPLOYED', axis=1, inplace=True)
ar.drop('FLAG_MOBIL', axis=1, inplace=True)
```

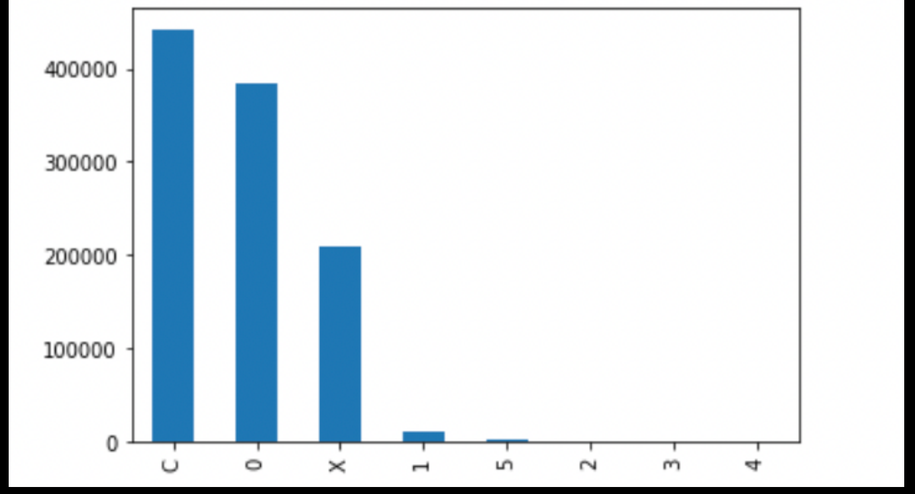
3

Feature Engineering (credit record)

Most of the "STATUS" column are "X","C","0".

We create new column such as "CUSTOMER_FOR_MONTHS" shows us how many months they are our clients.Then we group by ID and assign it to new dataframe - "credit_record"

```
cr["STATUS"].value_counts().plot.bar()
plt.show()
```




```
credit_record.head()
```

ID	CUSTOMER_FOR_MONTHS
5001711	3
5001712	18
5001713	21
5001714	14
5001715	59

3

Feature Engineering (credit record)

To create "target" column ,we use Vintage Analysis.Based on Vintage analysis,we ca say that if person has 30/60/90 days past due,she/he is "bad client".

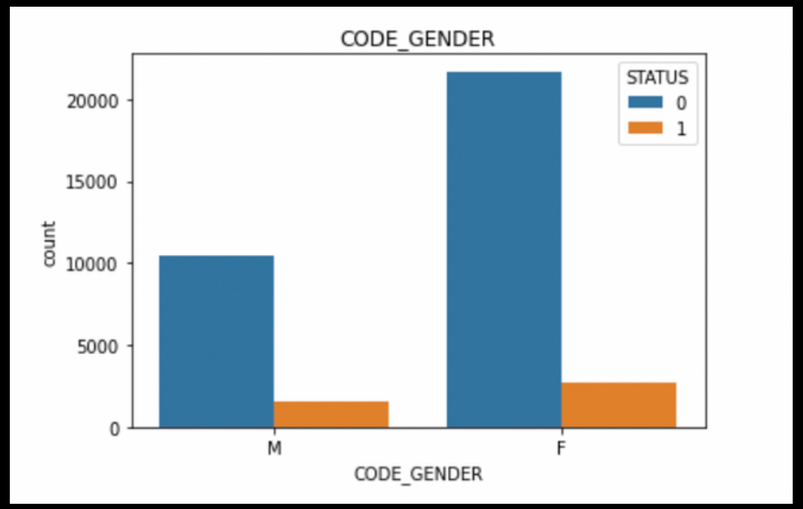
```
cr['STATUS'][cr["STATUS"] == 'C'] = 0
cr['STATUS'][cr["STATUS"] == 'X'] = 0
cr.head()
```

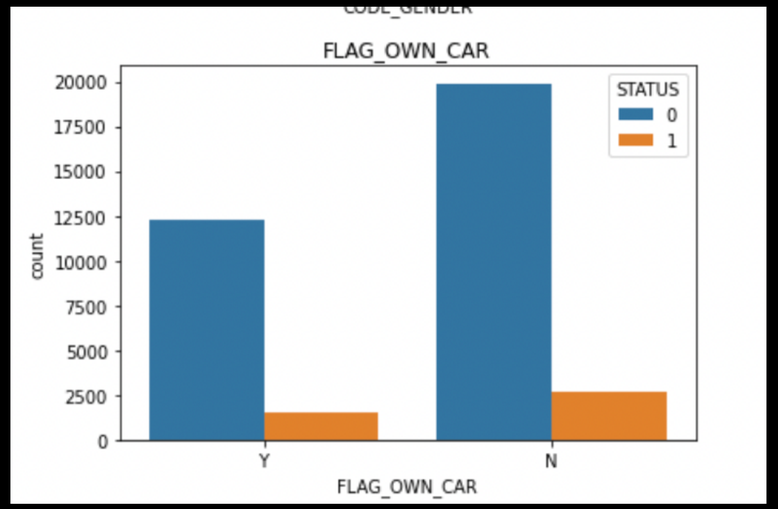
if "Status" column greater and equaly to 1 ,it means that clients is "bad client".For doing it,we changed data type of "Status" column.

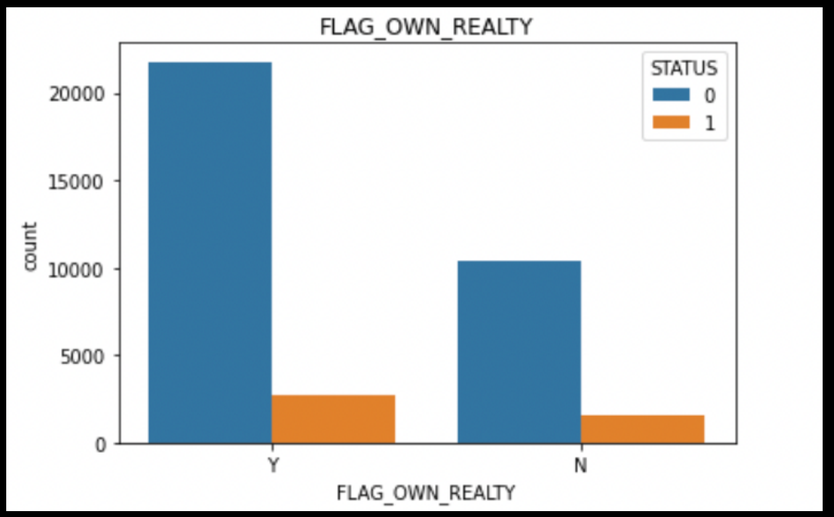
```
cr['STATUS'] = cr['STATUS'].apply(lambda x : 1 if x>=1 else 0)
cr.drop(['MONTHS_BALANCE'],axis = 1,inplace = True)
cr.drop(['CUSTOMER_FOR_MONTHS'],axis = 1,inplace = True)
cr.head()
```

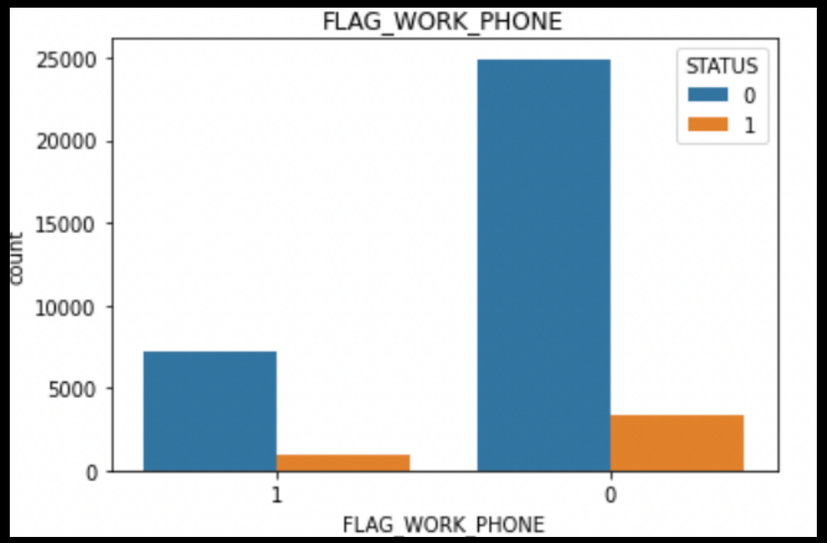
	ID	STATUS
418975	5050603	1
832451	5117313	1
979818	5142362	1
947078	5135636	1
810410	5116036	1

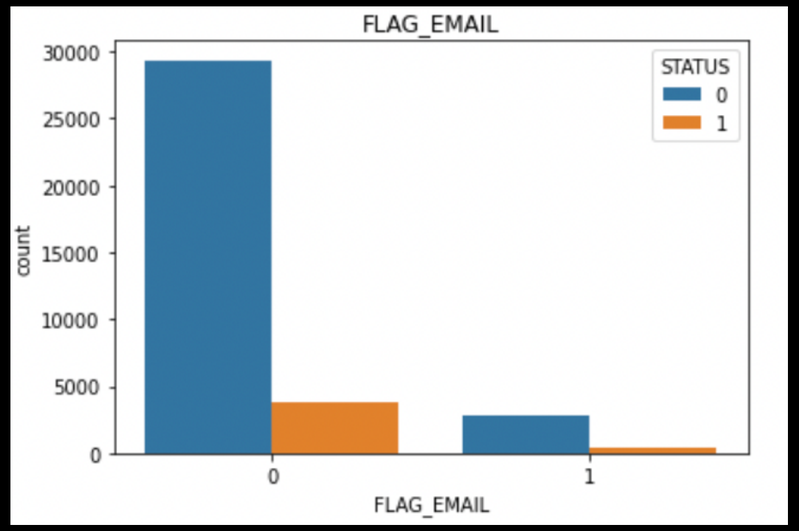
4 Data analyzing

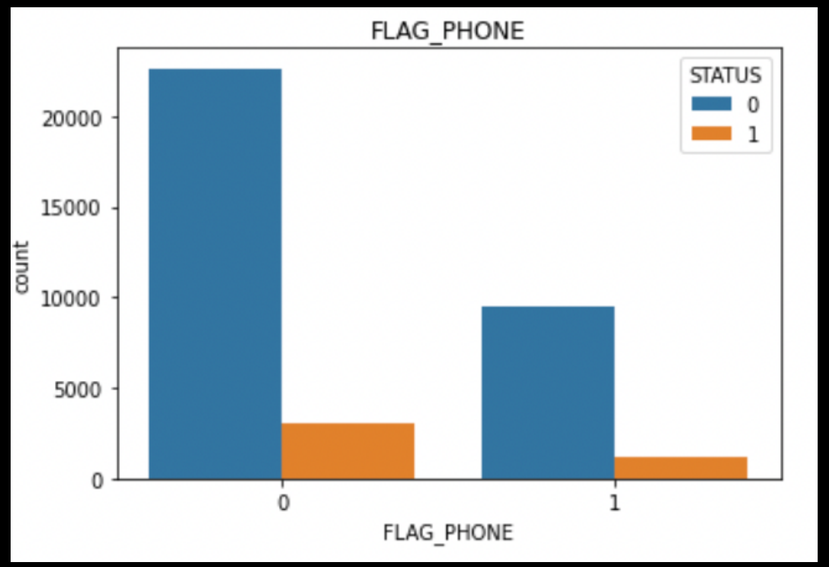




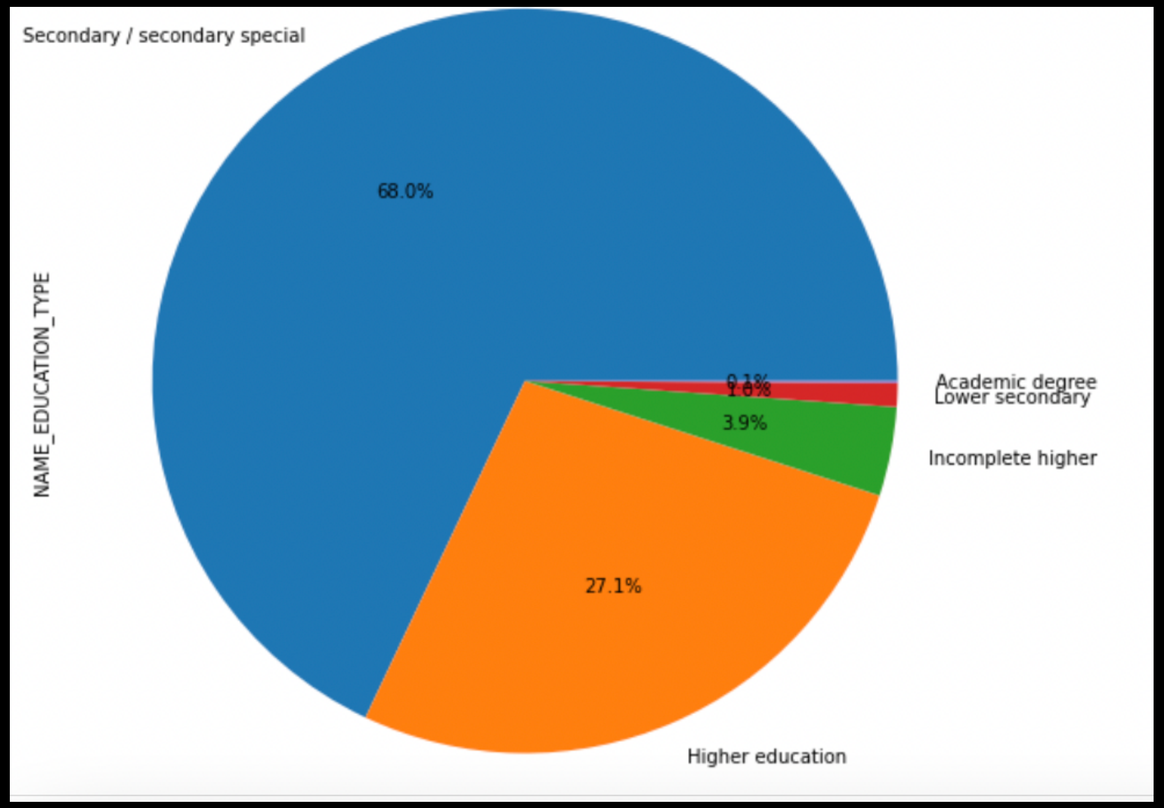


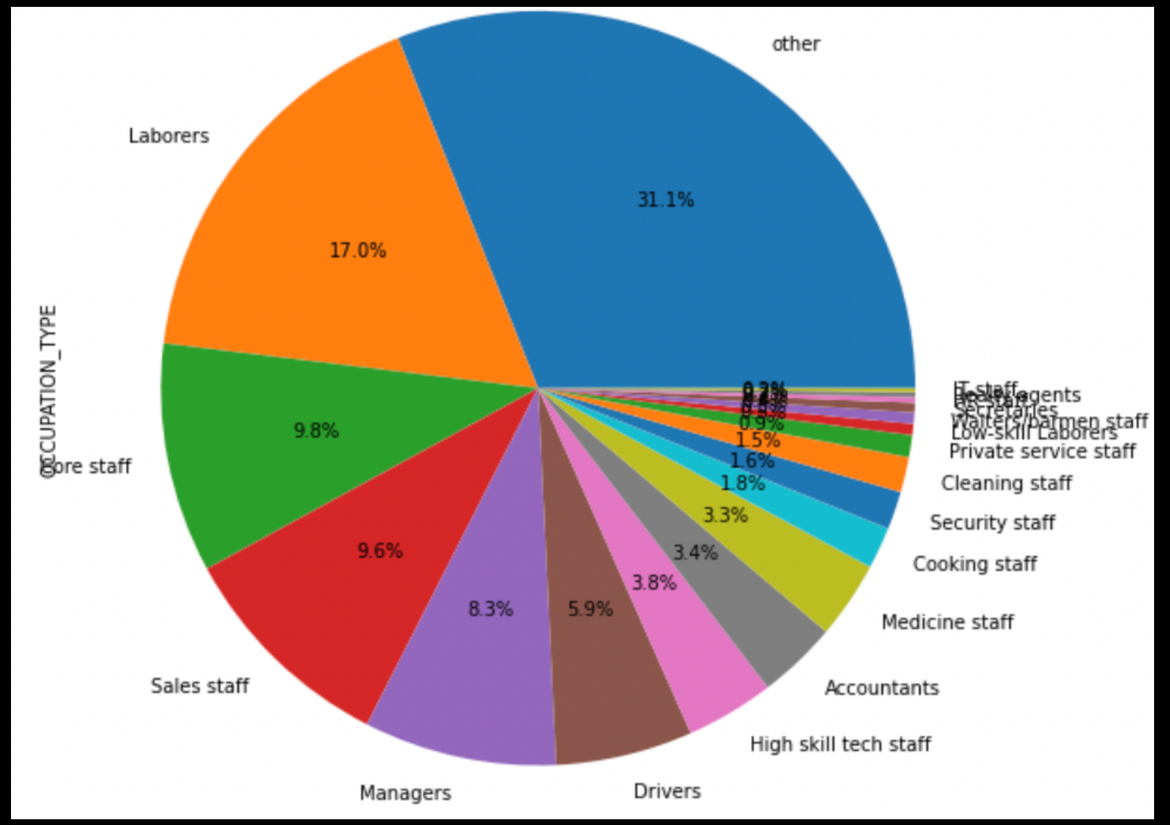




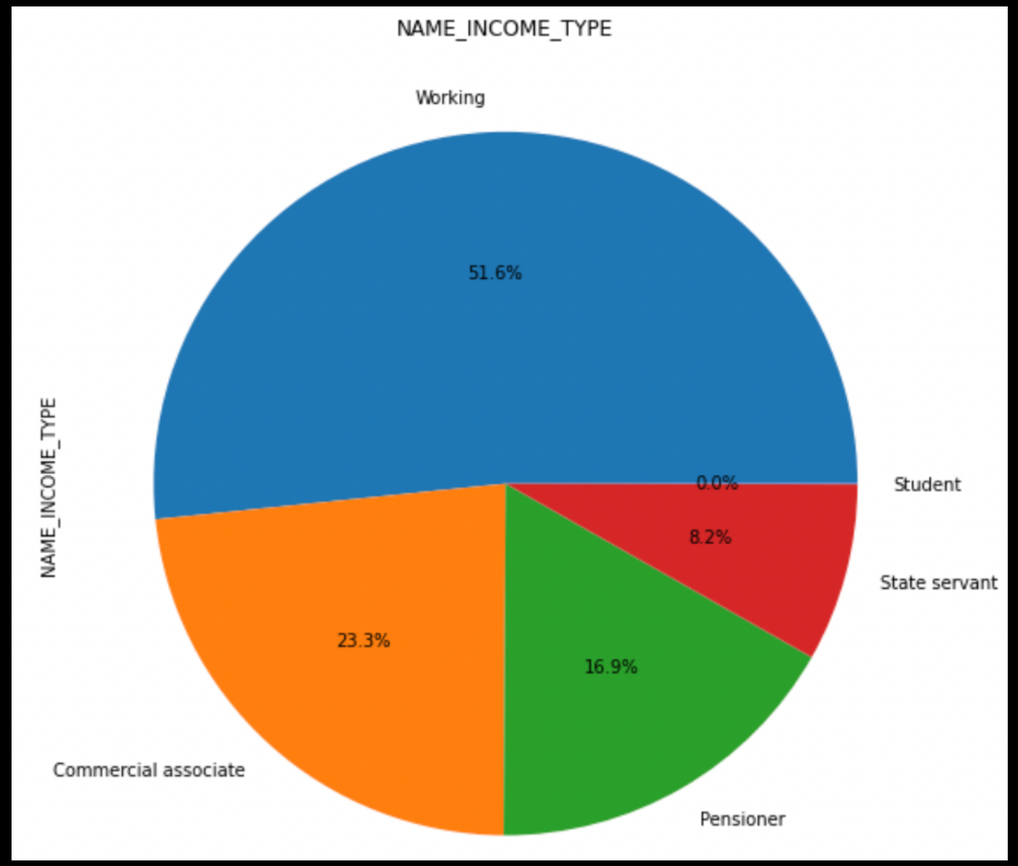


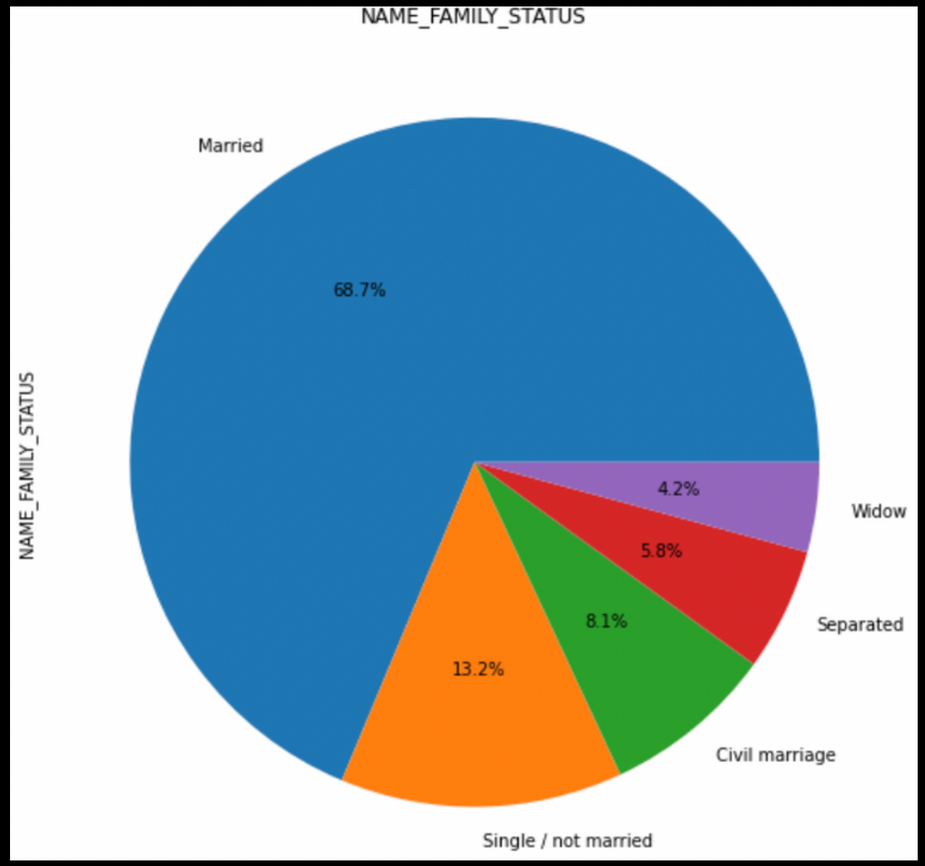
4 Data analyzing





4 Data analyzing





5 Machine Learning

1 — 2 — 3 — 4 — 5

Step

Drop "ID" column and to do get_dummies()

Step

Split date to X_train, X_test, Y_train, Y_test,To do standardscaling

Step

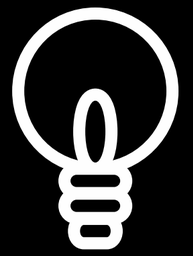
Modelling

Step

Oversampling and remodelling

Step

Model tunning



5 Machine Learning-Smote Tomek

Before
Smote Tomek

LogisticRegression

	precision	recall	f1-score	support
0	0.88	1.00	0.94	10570
1	1.00	0.01	0.01	1461
accuracy			0.88	12031
macro avg	0.94	0.50	0.47	12031
weighted avg	0.89	0.88	0.82	12031

DecisionTreeClassifier

	precision	recall	f1-score	support
0	0.90	0.90	0.90	10570
1	0.30	0.30	0.30	1461
accuracy			0.83	12031
macro avg	0.60	0.60	0.60	12031
weighted avg	0.83	0.83	0.83	12031

After
Smote Tomek

LogisticRegression

	precision	recall	f1-score	support
0	0.89	0.83	0.86	10570
1	0.16	0.23	0.19	1461
accuracy			0.75	12031
macro avg	0.52	0.53	0.52	12031
weighted avg	0.80	0.75	0.77	12031

DecisionTreeClassifier

	precision	recall	f1-score	support
0	0.91	0.89	0.90	10570
1	0.30	0.35	0.32	1461
accuracy			0.82	12031
macro avg	0.60	0.62	0.61	12031
weighted avg	0.83	0.82	0.83	12031

5 Machine Learning-SMOTE Tomek

Before Smote Tomek

KNeighborsClassifier

	precision	recall	f1-score	support
0	0.89	0.97	0.93	10570
1	0.43	0.15	0.22	1461
accuracy			0.87	12031
macro avg	0.66	0.56	0.57	12031
weighted avg	0.84	0.87	0.84	12031

RandomForestClassifier

	precision	recall	f1-score	support
0	0.90	0.96	0.93	10570
1	0.48	0.25	0.33	1461
accuracy			0.88	12031
macro avg	0.69	0.61	0.63	12031
weighted avg	0.85	0.88	0.86	12031

After Smote Tomek

KNeighborsClassifier

	precision	recall	f1-score	support
0	0.92	0.79	0.85	10570
1	0.25	0.51	0.34	1461
accuracy			0.76	12031
macro avg	0.59	0.65	0.60	12031
weighted avg	0.84	0.76	0.79	12031

RandomForestClassifier

	precision	recall	f1-score	support
0	0.91	0.94	0.93	10570
1	0.44	0.32	0.37	1461
accuracy			0.87	12031
macro avg	0.68	0.63	0.65	12031
weighted avg	0.85	0.87	0.86	12031

5 Machine Learning-RandomForestClassifier

Before model tunning

RandomForestClassifier

	precision	recall	f1-score	support
0	0.91	0.94	0.93	10570
1	0.44	0.32	0.37	1461
accuracy			0.87	12031
macro avg	0.68	0.63	0.65	12031
weighted avg	0.85	0.87	0.86	12031

After model tunning

	precision	recall	f1-score	support
0	0.91	0.95	0.93	10570
1	0.46	0.30	0.36	1461
accuracy			0.87	12031
macro avg	0.69	0.62	0.65	12031
weighted avg	0.85	0.87	0.86	12031

5 Machine Learning-KNN

Before model tunning

KNeighborsClassifier

	precision	recall	f1-score	support
0	0.92	0.79	0.85	10570
1	0.25	0.51	0.34	1461
accuracy			0.76	12031
macro avg	0.59	0.65	0.60	12031
weighted avg	0.84	0.76	0.79	12031

After model tunning

	precision	recall	f1-score	support
0	0.91	0.88	0.90	10570
1	0.32	0.40	0.35	1461
accuracy			0.82	12031
macro avg	0.62	0.64	0.63	12031
weighted avg	0.84	0.82	0.83	12031

Thank you for your attention!