Northeastern

College of Professional Studies
Northeastern University San Jose

MPS Analytics

Course: ALY6020

Assignment:

Module 3 – Midweek Project

Submitted on:

October 12, 2023

Submitted to:

Professor: Ahmadi Behzad

Submitted by:

Heejae Roh

Introduction

The main point of this dataset is whether a personal loan has been accepted or not. Other columns contain information about the applicant's income or banking usage patterns. I will run logistic regression based on this dataset and proceed with how to determine features in logistic regression and how to review the impact of features.

Answering Questions

1. What were the three most significant variables?

Coefficient and p-value table of Logistic Regression after elimination

Column Name	coefficient	p-value	Rank of P-value
age	-0.5088	6.025934e-114	2
experience	0.5117	2.100041e-94	3
income	0.0542	4.616968e-115	1
family	0.6187	2.832606e-17	5
education	1.6394	9.210361e-48	4
securities_account	0.4303	6.515244e-02	6
online	-0.0629	6.808598e-01	8
creditcard	-0.2099	2.104415e-01	7

Three most significant variables: income > age > experience

P-value Ascending order: income < age < experience

The most significant variable according to p-value is income. Income recorded the lowest p-value at 4.61e-115. Next is age. Age has the next lowest p-value at 6.02e-114. For education, in the data description, 1 indicates undergrad, 2 indicates graduate, and 3 indicates advanced/professional. education recorded the third lowest p-value. Personal loan, which is a binary variable and has positive correlation between income and income has the greatest significance, followed by negative correlation age, and thirdly, positive experience.

2. Of those three, which had the most negative influence on loan acceptance?

Coefficient and p-value table of Logistic Regression after elimination

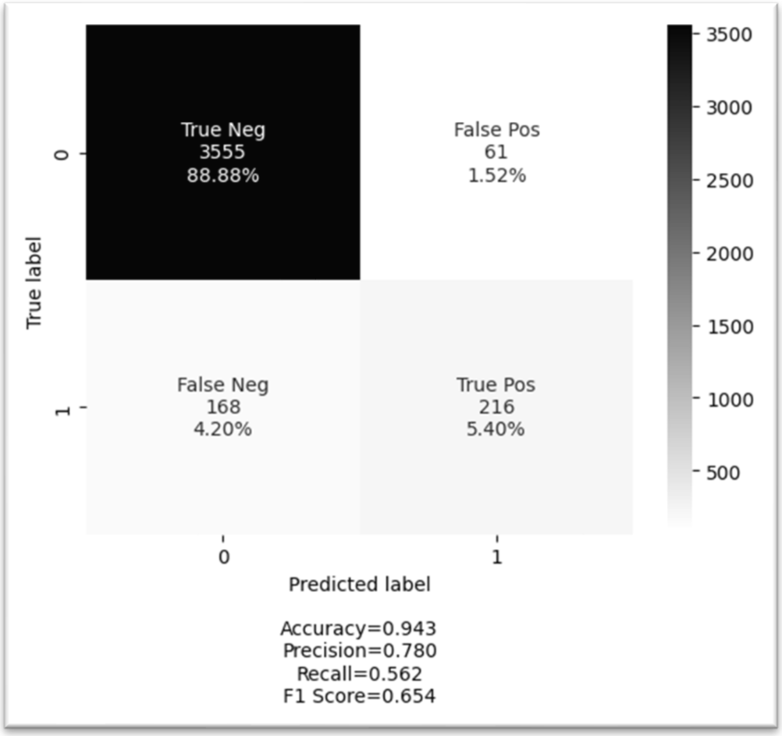
Column Name	coefficient	p-value	Rank of P-value
age	-0.5088	6.025934e-114	2
experience	0.5117	2.100041e-94	3
income	0.0542	4.616968e-115	1
family	0.6187	2.832606e-17	5
education	1.6394	9.210361e-48	4
securities_account	0.4303	6.515244e-02	6
online	-0.0629	6.808598e-01	8
creditcard	-0.2099	2.104415e-01	7

The most negative influence: age

P-value Ascending order in negative coefficient: age < online < creditcard

Variables with negative influence can also be known through p-value. Unlike the table above, this table shows the result of excluding mortgage with p-values higher than 0.25 and then performing logistic regression according to the rule of thumbs. age recorded the lowest p-value, but the coefficient was negative. Among the columns with negative coefficients, online recorded the next lowest p-value, and overall, it had the 8th lowest p-value among the 10 variables, showing that its significance is lower than that of other many positive variables. Next, creditcard had the third lowest p-value, or significance, among variables with negative coefficients.

3. How accurate was the model overall and what was the precision rate?



A confusion matrix was constructed by dividing the trained model into testset and predicted value. Accuracy recorded 0.943. Precision (among positively predicted, true positive) recorded 0.780. Since most of this data is negative (0), that is, data that has not been loaned, we need to focus more on precision rather than recall. Accordingly, the recall showed a relatively large number of false negatives (147), but the number of true positives was higher, recording a recall of 0.562. In a situation where there are many trainset from which negatives can be selected, it can be said that the positive data was predicted relatively well, although the recall recorded as 0.562. The F1-score recorded 0.654.

Dataset Understanding

The dataset is about binary personal loan and factors that can be used to decide accepting loan or not. There are 12 columns and 5000 entries. It is assumed that various data were extracted to predict whether a personal loan will be approved. Judging by the number of 5,000, it appears that the number was determined and extracted.

Description of the variables/features in the dataset.

#	column name	Description
1	**ID**	Customer Id
2	**Age**	Customer's age in completed years
3	**Experience**	#years of professional experience
4	**Income**	Annual income of the customer ($000)
5	**ZIPCode**	Home Address ZIP code.
6	**Family**	Family size of the customer
7	**CCAvg**	Avg. spending on credit cards per month ($000)
8	**Education**	Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
9	**Mortgage**	Value of house mortgage if any. ($000)
10	**Personal Loan**	Did this customer accept the personal loan offered in the last campaign?
11	**Securities Account**	Does the customer have a securities account with the bank?
12	**CD Account**	Does the customer have a certificate of deposit (CD) account with the bank?
13	**Online**	Does the customer use internet banking facilities?
14	**CreditCard**	Does the customer use a credit card issued by UniversalBank?

Headtail of Dataset 1

id	age	experience	income	zip_code	family	ccavg	education
1	25	1	49	91107	4	1.6	1
2	3	19	34	90089	3	1.5	1
3	39	15	11	94720	1	1.0	1
…				…			
4998	63	39	24	93023	2	0.3	3
4999	65	40	49	90034	3	0.5	2
5000	28	4	83	92612	3	0.8	1

Headtail of Dataset 2

id	mortgage	loan	securities_account	cd_account	online	creditcard
1	0	0	1	0	0	0
2	0	0	1	0	0	0
3	0	0	0	0	0	0
…			…			
4998	0	0	0	0	0	0
4999	0	0	0	0	1	0
5000	0	0	0	0	1	1

Data Cleansing

1. I first checked if there was a missing_value in the data, and there was no missing value in the data. All data were numerical data.
2. Next, we checked the number of unique data. personal loan, Securities Account, CD Account, Online, CreditCard appeared as binary data containing 0 and 1. Family and Education have 4 and 3 categories respectively.
3. All column names have been changed to lowercase.
4. Prior to analysis, ids, which are automatically generated consecutive numbers, were first excluded. After searching again for the meaning of zip code, we decided that it would be difficult to use it in analysis as a numerical variable, so we excluded it. ZIP Codes are numbered with the first digit representing a certain group of U.S. states, the second and third digits together representing a region in that group (or perhaps a large city) and the fourth and fifth digits representing a group of delivery addresses within that region (wikipedia, 2023).
5. Rule of thumb: select all the variables whose p-value < 0.25 along with the variables of knownclinical importance (utdallas, n.d.). After analysis excluding id and zip_code, the mortgage column was excluded with a p-value of 0.405, which is higher than 0.25, according to the rule of thumb.
6. After checking correlation between variables: ccvag (numerical), and cd_account (binary) are deleted.

Exploratory Data Analysis

Descriptive Analysis of Dataset 1

	age	experience	income	zip_code	family	ccavg
count	5000	5000	5000	5000	5000	5000
mean	45.34	20.10	73.77	93152.50	2.39	1.94
std	11.46	11.47	46.03	2121.85	1.15	1.75
min	23.00	-3.00	8.00	9307.00	1.00	0.00
25%	35.00	10.00	39.00	91911.00	1.00	0.70
50%	45.00	20.00	64.00	93437.00	2.00	1.50
75%	55.00	30.00	98.00	94608.00	3.00	2.50
max	67.00	43.00	224.00	96651.00	4.00	10.00

Descriptive Analysis of Dataset 2

	education	mortgage	personal_loan	securities_account
count	5000	5000	5000	5000
mean	1.88	56.50	0.10	0.10
std	0.84	101.71	0.29	0.31
min	1.00	0.00	0.00	0.00
25%	1.00	0.00	0.00	0.00
50%	2.00	0.00	0.00	0.00
75%	3.00	101.00	0.00	0.00
max	10.00	3.00	635.00	1.00

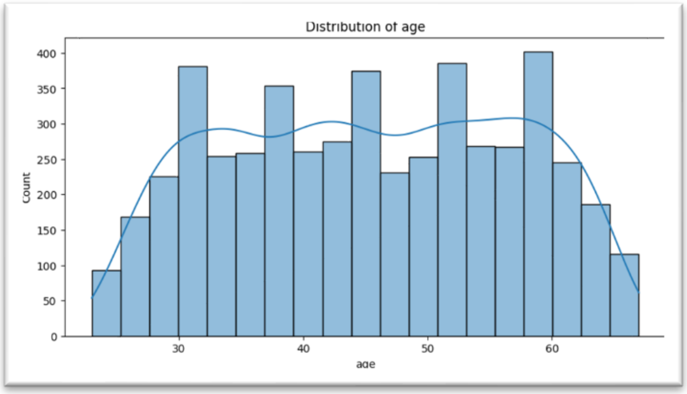
Descriptive Analysis of Dataset 3

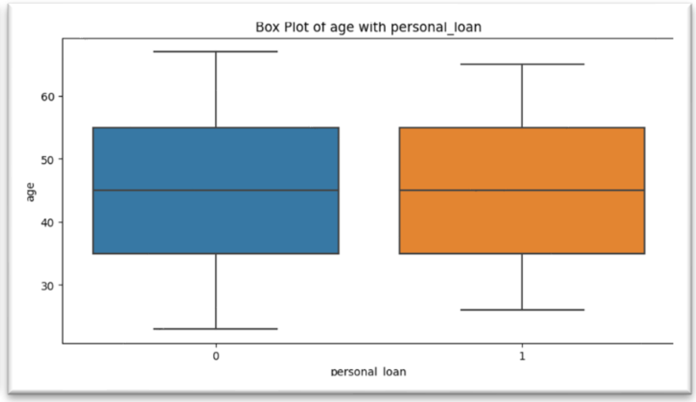
	cd_account	online	creditcard
count	5000	5000	5000
mean	0.06	0.60	0.29
std	0.00	0.00	0.00
min	0.00	0.00	0.00
25%	0.00	1.00	0.00
50%	0.00	1.00	0.00
75%	0.00	1.00	1.00
max	1.00	1.00	1.00

1. All variables have 5000 counts. Among them, 'id' is an automatically assigned number from 1 to 5000.
2. age showed a mean of 45.34. experience has a mean of 20.1.
3. Income showed 73.77. This represents an annual income of 73K according to the description.
4. family indicates the number of family members with numbers from 1 to 4.
5. ccvg showed a mean of 1.94, and like income, it indicates monthly spending in credit card of 1.94K.
6. For education, the mean was 1.88, with educational levels ranging from undergrad to graduate close to graduate.
7. mortgage refers to a home equity loan, and the mean was 56.5K.
8. Personal loan and securities_account are binary variables, and the mean is 0.1, so you can see that there are many more values that are 0.
9. cd_account recorded an even lower value of 0.06. On the other hand, online recorded 0.6, showing that there are more users using online banking.
10. Regarding whether a credit card was issued by UniversalBank, the mean was 0.29, showing that there were more people who did not have a credit card issued.

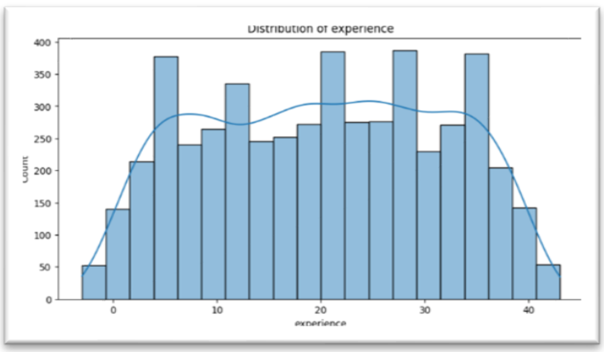
Data Visualizations

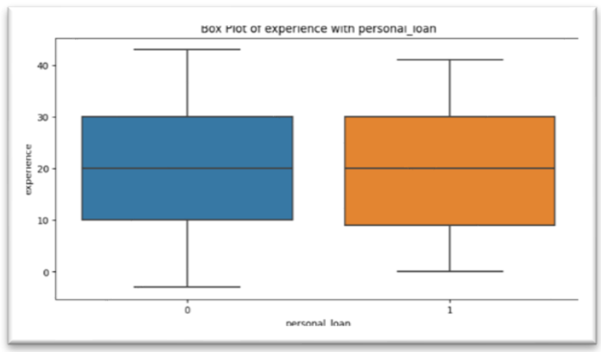
Histograms, Pie chart, and box plot of primary attribute for logistic regression





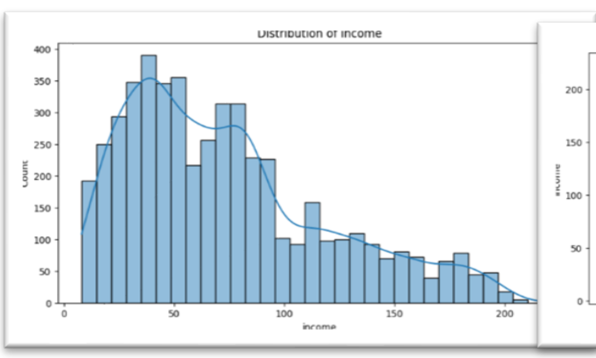
Age is distributed fairly evenly by age, but the three largest groups are clustered in the early 30s, late 30s, mid 40s, early 50s, and late 50s. The box plot according to personal_loan does not show much difference.

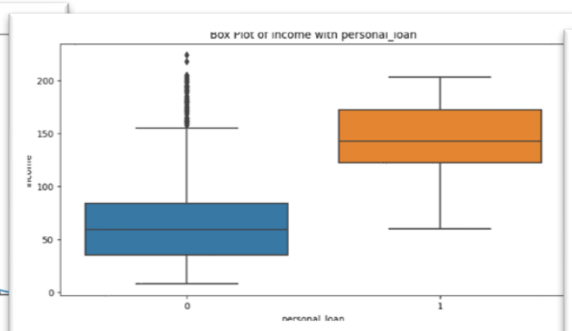


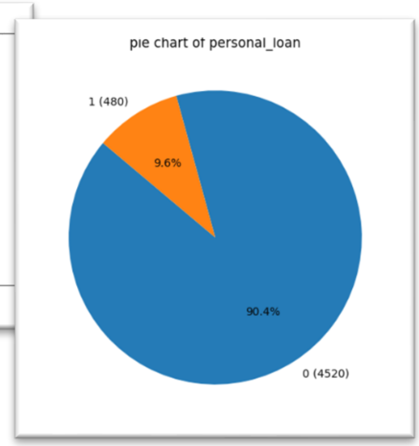


The experience histogram shows an almost similar form to the age distribution. Only the range is different, but the histograms have many similar shapes. Later in assumption, it seems necessary to check the correlation between the two data.

The income histogram shows a right skewed appearance. In the box plot on the right, personal loans show







a large difference between 0 and 1, and at 0, values that need to be considered for outliers are visible, but we will not process outliers in this analysis. Outliers have a significant impact on model analysis in regression, so more detailed analysis is required.

In the pie chart on the far right, '0', the percentage of personal loans not approved, recorded 90.4% or 4,520 values, and '1', the percentage of approved personal loans, recorded 9.6% with 480 values. We can see that our target variable is unbalanced data.

Logistic Regression Assumptions

Any logistic regression example in Python is incomplete without addressing model assumptions in the analysis. The important assumptions of the logistic regression model include:

Assumptions of Logistic Regression (Lillian, 2018)

A1. Target variable is binary

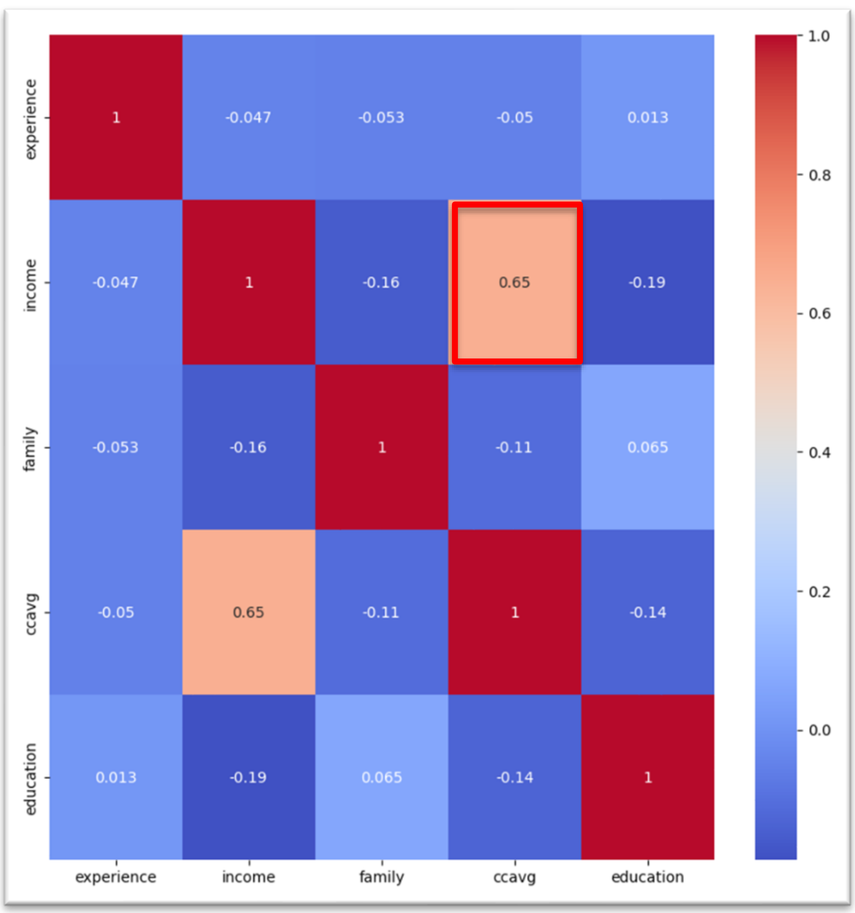
A2. Predictive features are interval (continuous) or categorical

A3. Features are independent of one another

A4. Sample size is adequate – Rule of thumb: 50 records per predictor

Since most of the other assumptions were satisfied, I will compare the correlation between variables to confirm the third assumption. Numerical variables will use a correlation matrix, and binary variables will be analyzed using chi-square test values.

Correlation Matrix



In physics and chemistry, a correlation coefficient should be lower than -0.9 or higher than 0.9 for the correlation to be considered meaningful, while in social sciences the threshold could be as high as -0.5 and as low as 0.5 (Jason, 2023). Therefore, I will ultimately perform logistic regression excluding ccavg from the table above. This is because income has a significant influence on personal_loan and exceeds the value of 0.5.

	chi-square p-value	correlated or not
('securities_account', 'cd_account')	2.32e-110	'correlated'
('securities_account', 'online')	0.3976	'not-correlated'
('securities_account', 'creditcard')	0.3115	'not-correlated'
('cd_account', 'online')	3.52e-35	'correlated'
('online', 'creditcard')	0.7902	'not-correlated'

Binary variables were checked for correlation with each other through the chi-square test. Since cd_account is correlated with two columns, we will exclude it from the final analysis and run logistic regression.

Result of Logistic Regression

Result 1: before elimination

```
                           Logit Regression Results
==============================================================================
Dep. Variable:          personal_loan   No. Observations:                 4000
Model:                          Logit   Df Residuals:                     3989
Method:                           MLE   Df Model:                           10
Date:                Thu, 12 Oct 2023   Pseudo R-squ.:                  0.5910
Time:                        05:29:50   Log-Likelihood:                -517.35
converged:                       True   LL-Null:                       -1264.8
Covariance Type:            nonrobust   LLR p-value:                     0.000
======================================================================================
                         coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------------
age                   -0.5281      0.026    -20.470      0.000      -0.579      -0.478
experience             0.5310      0.028     18.641      0.000       0.475       0.587
income                 0.0552      0.003     18.598      0.000       0.049       0.061
family                 0.7417      0.083      8.919      0.000       0.579       0.905
ccavg                  0.0751      0.043      1.731      0.083      -0.010       0.160
education              1.6952      0.125     13.551      0.000       1.450       1.940
mortgage               0.0004      0.001      0.705      0.481      -0.001       0.002
securities_account    -1.0407      0.320     -3.253      0.001      -1.668      -0.414
cd_account             3.8781      0.362     10.707      0.000       3.168       4.588
online                -0.8209      0.175     -4.683      0.000      -1.164      -0.477
creditcard            -1.2173      0.226     -5.394      0.000      -1.660      -0.775
================================================================================
```

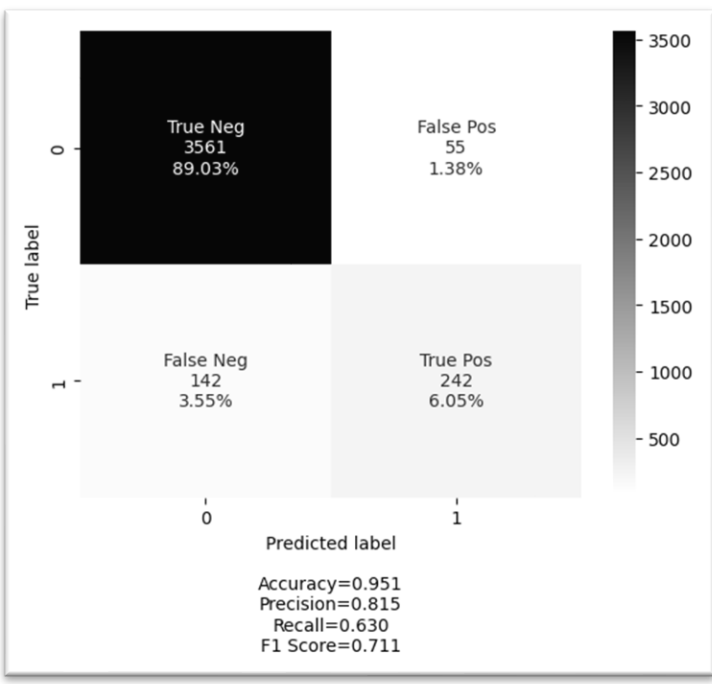
testset and trainset were separated at a ratio of 0.2 to 0.8. Therefore, we constructed logistic regression with 4000 trainset. This is the first result including all variables. The P-value of mortgage is higher than the rule of thumb of 0.25. Hence, mortgage is excluded from the final analysis. The ccavg and cd_account columns are 'A3.' in correlation check. Because these columns violate the assumption of logistic regression 'Features are independent of one another', it will be excluded from the final analysis even if the p-value is low.

Result 2: after elimination of mortgage, ccavg, and cd_account

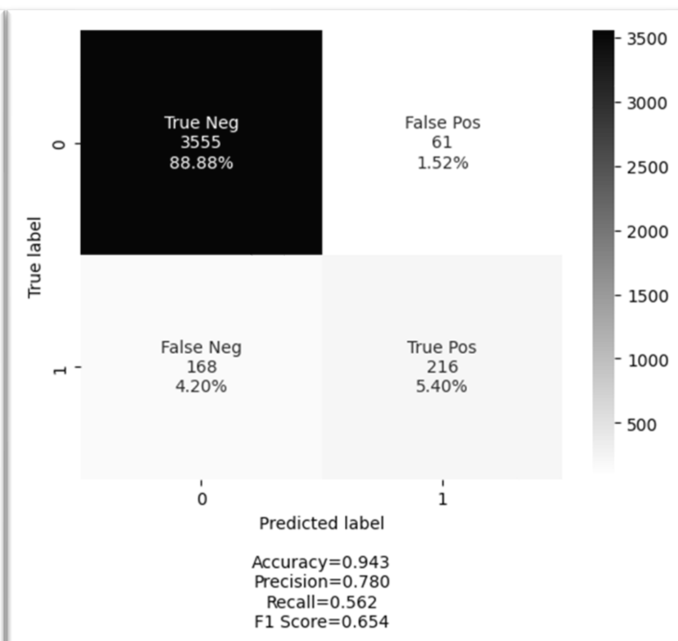
```
                           Logit Regression Results
==============================================================================
Dep. Variable:          personal_loan   No. Observations:                 4000
Model:                          Logit   Df Residuals:                     3992
Method:                           MLE   Df Model:                            7
Date:                Thu, 12 Oct 2023   Pseudo R-squ.:                  0.5167
Time:                        05:29:51   Log-Likelihood:                -611.35
converged:                       True   LL-Null:                       -1264.8
Covariance Type:            nonrobust   LLR p-value:                 5.244e-278
=======================================================================================
                          coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------
age                    -0.5088      0.022    -22.687      0.000      -0.553      -0.465
experience              0.5117      0.025     20.613      0.000       0.463       0.560
income                  0.0542      0.002     22.800      0.000       0.050       0.059
family                  0.6187      0.073      8.453      0.000       0.475       0.762
education               1.6394      0.113     14.519      0.000       1.418       1.861
securities_account      0.4303      0.233      1.844      0.065      -0.027       0.888
online                 -0.0629      0.153     -0.411      0.681      -0.363       0.237
creditcard             -0.2099      0.168     -1.252      0.210      -0.538       0.119
=======================================================================================
```

The logistic regression constructed based on all variables previously recorded Accuracy 0.951, precision 0.815, recall 0.630, and F1 Score of 0.711. Looking at the confusion matrix below, after removing the three variables, accuracy was 0.943, precision was 0.780, recall was 0.562, and F1 score was 0.654. Accuracy is not a good metric to use when you have class imbalance (Joos, 2001).

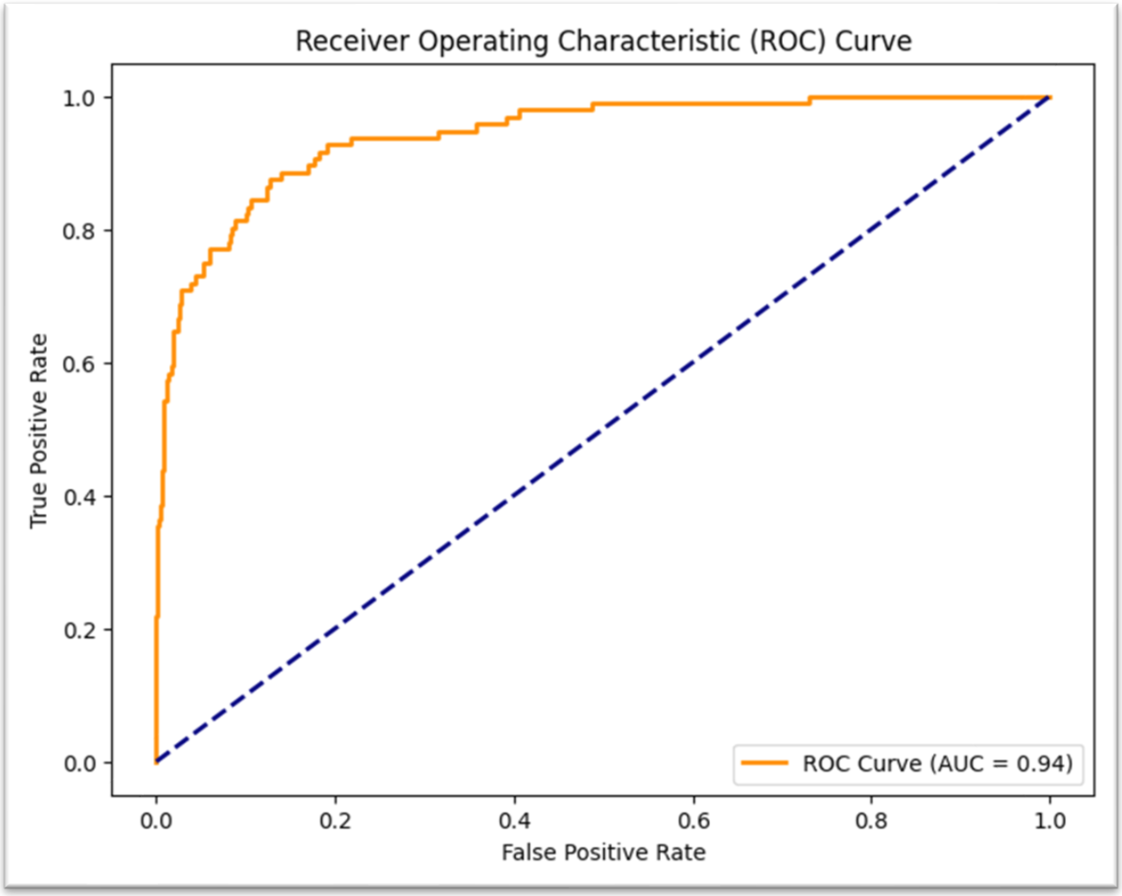
Before Feature selection



After Feature Selection



The logistic regression constructed based on all variables previously recorded Accuracy 0.951, precision 0.815, recall 0.630, and F1 Score of 0.711. Looking at the confusion matrix below, after removing the three variables, accuracy was 0.943, precision was 0.780, recall was 0.562, and F1 score was 0.654. Accuracy is not a good metric to use when you have class imbalance. One way to solve class imbalance problems is to work on your sample. Another way to solve class imbalance problems is to use better accuracy metrics like the F1 score (Joos, 2001).



Conclusion

In this analysis, I used logistic regression to find out which variables had the greatest influence on the factors that determine personal_loan. In addition, we also looked for ways to find out which variables have a negative effect. In the process, I determined whether the assumptions of logistic regression were satisfied and checked how to measure the results of logistic regression with imbalanced data.

Reference

Joos, Korstanje. (2021, August 31). The F1 score. Medium. Retrieved from https://towardsdatascience.com/the-f1-score-bec2bbc38aa6

sefidian. (n.d.). Measure the correlation between numerical and categorical variables and the correlation between two categorical variables in Python: Chi-Square and ANOVA. Retrieved from http://www.sefidian.com/2020/08/02/measure-the-correlation-between-numerical-and-categoricalvariables-and-the-correlation-between-two-categorical-variables-in-python-chi-square-andanova/#:~:text=The%20ANOVA%20test%20is%20used,variables%20for%20each%20categorical%20value.

ML Explained. (2023). Calculate correlation among categorical variables in Python. YouTube. Retrieved from https://www.youtube.com/watch?v=fzzUfa0-VsE

Lillian Pierson, P.E. (2018). Logistic regression example in Python. DATA MANIA. Retrieved from https://www.data-mania.com/blog/logistic-regression-example-in-python/

Jason, Fernando. (2023, May 12). The correlation coefficient:wWhat It Is, What It tells investors. Investopedia. Retrieved from https://www.investopedia.com/terms/c/correlationcoefficient.asp#:~:text=In%20physics%20and%20chemistry%2C%20a,and%20as%20low%20as%200.5.

wikipedia. (2023). ZIP Code. Retrieved from https://en.wikipedia.org/wiki/ZIP_Code#:~:text=ZIP%20Codes%20are%20numbered%20with,delivery%20addresses%20within%20that%20region.

utdallas. (n.d.). Building and applying logistic regression models. Retrieved from chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://personal.utdallas.edu/~pkc022000/6390/SP06/NOTES/Logistic_Regression_4.pdf

Kenneth, Leung. (2021, October 4). Assumptions of logistic regression, clearly explained. Medium. Retrieved from https://towardsdatascience.com/assumptions-of-logistic-regression-clearly-explained-44d85a22b290#:~:text=Logistic%20regression%20does%20not%20require,but%20not%20for%20logistic%20regression.

RITHP. (2023). Logistic regression for feature selection: selecting the right features for your model. Medium. Retrieved from https://medium.com/@rithpansanga/logistic-regression-for-feature-selection-selecting-the-right-features-for-your-model-410ca093c5e0

Susan, Li. (2017, September 28). Building A logistic regression in python, step by step. Medium. Retrieved from https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

pandas. (n.d.). pandas.read_excel. Retrieved from https://pandas.pydata.org/pandas-docs/version/0.25.2/reference/api/pandas.read_excel.html

DTrimarchi10. (n.d.). confusion_matrix. github. Retrieved from https://github.com/DTrimarchi10/confusion_matrix/blob/master/cf_matrix.py

Appendix (Python code):

```python
import pandas as pd
import numpy as np
from sklearn import datasets
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from collections import Counter

"""### DATA Import"""

df = pd.read_excel('Bank_Personal_Loan_Modelling.xlsx', sheet_name='Data')

df.head()

df.tail()

"""## Visualization & Understanding Dataset"""

def missing_values(df):
    missing_number = df.isnull().sum().sort_values(ascending=False)
    missing_percent = (df.isnull().sum()/df.isnull().count()).sort_values(ascending=False)
    missing_values = pd.concat([missing_number, missing_percent], axis=1, keys=['Missing_Number',
'Missing_Percent'])
    return missing_values[missing_values['Missing_Number']>0]

def first_looking(df):
    print(colored("Shape:", attrs=['bold']), df.shape,'\n',
          colored('-'*79, 'red', attrs=['bold']),
          colored("\nInfo:\n", attrs=['bold']), sep='')
    print(df.info(), '\n',
          colored('-'*79, 'red', attrs=['bold']), sep='')
    print(colored("Number of Uniques:\n", attrs=['bold']), df.nunique(),'\n',
          colored('-'*79, 'red', attrs=['bold']), sep='')
    print(colored("Missing Values:\n", attrs=['bold']), missing_values(df),'\n',
          colored('-'*79, 'red', attrs=['bold']), sep='')
    print(colored("All Columns:", attrs=['bold']), list(df.columns),'\n',
          colored('-'*79, 'red', attrs=['bold']), sep='')

    df.columns= df.columns.str.lower().str.replace('&', '_').str.replace(' ', '_')

    print(colored("Columns after rename:", attrs=['bold']), list(df.columns),'\n',
              colored('-'*79, 'red', attrs=['bold']), sep='')

import colorama
from colorama import Fore, Style  # maakes strings colored
from termcolor import colored

missing_values(df)

first_looking(df)

df.describe()

pip install ydata-profiling

import ydata_profiling

df.profile_report()

"""## Data Visualization"""

import seaborn as sns
```

```python
df1 = df.copy()

# wheelbase histogram
#distribution of capital_gain
plt.figure(figsize=(10, 5))
sns.histplot(x=df1['age'], kde=True)

plt.title("Distribution of age")

# Box plot with personal_loan
plt.figure(figsize=(10, 5))
sns.boxplot(x='personal_loan', y='age', data=df1)

plt.title("Box Plot of age with personal_loan")

# compressionratio histogram
#distribution of compressionratio
plt.figure(figsize=(10, 5))
sns.histplot(x=df1['experience'], kde=True)

plt.title("Distribution of experience")

# Box plot with personal_loan
plt.figure(figsize=(10, 5))
sns.boxplot(x='personal_loan', y='experience', data=df1)

plt.title("Box Plot of experience with personal_loan")

# income histogram
#distribution of income
plt.figure(figsize=(10, 5))
sns.histplot(x=df1['income'], kde=True)

plt.title("Distribution of income")

# Box plot with personal_loan
plt.figure(figsize=(10, 5))
sns.boxplot(x='personal_loan', y='income', data=df1)

plt.title("Box Plot of income with personal_loan")

plt.figure(figsize=(10, 5))
ax = sns.boxplot(y='income', data=df1)

minimum = df1['income'].min()
maximum = df1['income'].max()
median = df1['income'].median()
q1 = df1['income'].quantile(0.25)
q3 = df1['income'].quantile(0.75)

ax.text(0.8, minimum, f"Min: {minimum}", ha='center', va='center', color='white',
bbox=dict(facecolor='blue', edgecolor='blue'))
ax.text(0.8, q1, f"Q1: {q1}", ha='center', va='center', color='white', bbox=dict(facecolor='blue',
edgecolor='blue'))
ax.text(0.8, median, f"Median: {median}", ha='center', va='center', color='white',
bbox=dict(facecolor='blue', edgecolor='blue'))
ax.text(0.8, q3, f"Q3: {q3}", ha='center', va='center', color='white', bbox=dict(facecolor='blue',
edgecolor='blue'))
ax.text(0.8, maximum, f"Max: {maximum}", ha='center', va='center', color='white',
bbox=dict(facecolor='blue', edgecolor='blue'))

plt.title("Box Plot of income with Five-Number Summary")
```

```python
counts = df1['personal_loan'].value_counts()

# Extract labels and sizes for the pie chart
labels = counts.index.tolist()
sizes = counts.values

# Create a pie chart
plt.figure(figsize=(6, 6))  # Optional: Set the figure size
plt.pie(sizes, labels=[f'{label} ({count})' for label, count in zip(labels, sizes)], autopct='%.1f%%',
startangle=140)

# Display the pie chart
plt.title('pie chart of personal_loan')
plt.show()

"""#### Correlation matrix"""

df_num=df1[['experience', 'income', 'family', 'ccavg', 'education', ]]

car_corr_matrix=df_num.corr()

plt.figure(figsize=(10, 10))
sns.heatmap(car_corr_matrix, cmap='coolwarm', annot=True)

"""#### Correlation between binary variables"""

from scipy.stats import chi2_contingency

cross_tab = pd.crosstab(index=df['securities_account'], columns=df['cd_account'])
cross_tab

chi_sq_result = chi2_contingency(cross_tab,)

p, x = chi_sq_result[1], "reject" if chi_sq_result[1] < 0.05 else "accept"

print(f"The p-value is {chi_sq_result[1]} and hence we {x} the null hypothesis with {chi_sq_result[2]}
degrees of freedom")

def is_correlated(x,y):
  ct=pd.crosstab(index=df[x], columns=df[y])
  chi_sq_result = chi2_contingency(ct,)
  p, x = chi_sq_result[1], "correlated" if chi_sq_result[1] < 0.05 else "not-correlated"
  return p, x

is_correlated('securities_account', 'cd_account')

is_correlated('securities_account', 'online')

is_correlated('securities_account', 'creditcard')

is_correlated('cd_account', 'online')
# Delete cd_account

is_correlated('online', 'creditcard')

"""## Analysis
### Saving before changing df as df1
"""
df1 = df.copy()

"""### Logistics regression model fitting"""
```

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.inspection import permutation_importance
from sklearn import metrics
from scipy.stats import norm
import statsmodels.api as sm

y=df[['personal_loan']]
x=df.drop(['personal_loan', 'id', 'zip_code'], axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

x_train

Y=y_train
X=x_train
model = sm.Logit(endog=Y, exog=X).fit()

print(model.summary())

pred = model.predict(X)

binary_predictions = round(pred)

# Create a logistic regression model
model = sm.Logit(endog=Y, exog=X).fit()
p_values = model.pvalues
print(p_values)

ranked_features = p_values.sort_values(ascending=True)
print(ranked_features)

from sklearn.metrics._plot.confusion_matrix import confusion_matrix
# https://realpython.com/logistic-regression-python/
confusion = confusion_matrix(Y, binary_predictions)

def make_confusion_matrix(cf,
                          group_names=None,
                          categories='auto',
                          count=True,
                          percent=True,
                          cbar=True,
                          xyticks=True,
                          xyplotlabels=True,
                          sum_stats=True,
                          figsize=None,
                          cmap='Blues',
                          title=None):
    '''
    This function will make a pretty plot of an sklearn Confusion Matrix cm using a Seaborn heatmap
visualization.

    Arguments
    ---------
    cf:            confusion matrix to be passed in

    group_names:   List of strings that represent the labels row by row to be shown in each square.

    categories:    List of strings containing the categories to be displayed on the x,y axis. Default
is 'auto'

    count:         If True, show the raw number in the confusion matrix. Default is True.

    normalize:     If True, show the proportions for each category. Default is True.
```

```
    cbar:          If True, show the color bar. The cbar values are based off the values in the
confusion matrix.
                   Default is True.

    xyticks:       If True, show x and y ticks. Default is True.

    xyplotlabels:  If True, show 'True Label' and 'Predicted Label' on the figure. Default is True.

    sum_stats:     If True, display summary statistics below the figure. Default is True.

    figsize:       Tuple representing the figure size. Default will be the matplotlib rcParams value.

    cmap:          Colormap of the values displayed from matplotlib.pyplot.cm. Default is 'Blues'
                   See http://matplotlib.org/examples/color/colormaps_reference.html

    title:         Title for the heatmap. Default is None.

    '''


    # CODE TO GENERATE TEXT INSIDE EACH SQUARE
    blanks = ['' for i in range(cf.size)]

    if group_names and len(group_names)==cf.size:
        group_labels = ["{}\n".format(value) for value in group_names]
    else:
        group_labels = blanks

    if count:
        group_counts = ["{0:0.0f}\n".format(value) for value in cf.flatten()]
    else:
        group_counts = blanks

    if percent:
        group_percentages = ["{0:.2%}".format(value) for value in cf.flatten()/np.sum(cf)]
    else:
        group_percentages = blanks

    box_labels = [f"{v1}{v2}{v3}".strip() for v1, v2, v3 in
zip(group_labels,group_counts,group_percentages)]
    box_labels = np.asarray(box_labels).reshape(cf.shape[0],cf.shape[1])


    # CODE TO GENERATE SUMMARY STATISTICS & TEXT FOR SUMMARY STATS
    if sum_stats:
        #Accuracy is sum of diagonal divided by total observations
        accuracy  = np.trace(cf) / float(np.sum(cf))

        #if it is a binary confusion matrix, show some more stats
        if len(cf)==2:
            #Metrics for Binary Confusion Matrices
            precision = cf[1,1] / sum(cf[:,1])
            recall    = cf[1,1] / sum(cf[1,:])
            f1_score  = 2*precision*recall / (precision + recall)
            stats_text = "\n\nAccuracy={:0.3f}\nPrecision={:0.3f}\nRecall={:0.3f}\nF1
Score={:0.3f}".format(
                accuracy,precision,recall,f1_score)
        else:
            stats_text = "\n\nAccuracy={:0.3f}".format(accuracy)
    else:
        stats_text = ""
```

```
    # SET FIGURE PARAMETERS ACCORDING TO OTHER ARGUMENTS
    if figsize==None:
        #Get default figure size if not set
        figsize = plt.rcParams.get('figure.figsize')

    if xyticks==False:
        #Do not show categories if xyticks is False
        categories=False


    # MAKE THE HEATMAP VISUALIZATION
    plt.figure(figsize=figsize)

sns.heatmap(cf,annot=box_labels,fmt="",cmap=cmap,cbar=cbar,xticklabels=categories,yticklabels=categori
es)

    if xyplotlabels:
        plt.ylabel('True label')
        plt.xlabel('Predicted label' + stats_text)
    else:
        plt.xlabel(stats_text)

    if title:
        plt.title(title)

labels = ['True Neg','False Pos','False Neg','True Pos']
categories = ['0', '1']
make_confusion_matrix(confusion,
                      group_names=labels,
                      categories=categories,
                      cmap='binary')

"""#### Select with p-value"""

y=df[['personal_loan']]
x=df.drop(['personal_loan', 'id', 'zip_code', 'mortgage', 'ccavg', 'cd_account'], axis=1)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

x_train

Y=y_train
X=x_train
model = sm.Logit(endog=Y, exog=X).fit()
print(model.summary())

pred = model.predict(X)

binary_predictions = round(pred)

from sklearn.metrics._plot.confusion_matrix import confusion_matrix
# https://realpython.com/logistic-regression-python/
confusion = confusion_matrix(Y, binary_predictions)



labels = ['True Neg','False Pos','False Neg','True Pos']
categories = ['0', '1']
make_confusion_matrix(confusion,
                      group_names=labels,
                      categories=categories,
                      cmap='binary')
```

```python
"""### Feature Selection"""
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression

# Create a logistic regression model
model = sm.Logit(endog=Y, exog=X).fit()
p_values = model.pvalues
print(p_values)

ranked_features = p_values.sort_values(ascending=True)
print(ranked_features)

"""#### ROC curve"""
pip install stat

from sklearn.metrics import roc_curve, auc, roc_auc_score

pred2 = model.predict(x_test)
binary_predictions = round(pred2)

fpr, tpr, thresholds = roc_curve(y_true=y_test, y_score=pred2)
auc = roc_auc_score(y_true=y_test, y_score=pred2)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC Curve (AUC = {auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
```