MDS6212 Fintech Theory and Practice Assignment 01

Youcong Lu

Content

[Week 1 Assignment Report 3](#_Toc83067452)

[1.Data Overlook 3](#_Toc83067453)

[2.Data Cleaning 4](#_Toc83067454)

[3.Logit regression on single variables 5](#_Toc83067455)

[4.Logit regression on multiple variables 7](#_Toc83067456)

[5.Appendix 11](#_Toc83067457)

# Week 1 Assignment Report

## 1.Data Overlook

There are total 5000 records with 47 columns. Based on past research on likelihood of loan delinquency and loan approval, we do some statistic summary on the key variables on the sample space, including age, gender, loan amount, interest rate, credit scores.

Based on the result, we can define these key variables into three types:

(1) Personal properties (such as age and gender):

We found that the average age of the sample is about 27.5 which means it is a rather “young sample”. At the same time, after we transform gender into numeric value (0: Male, 1 : Female), the statistics indicates that the proportion of males in the sample is much higher. To be more specific, gentlemen are six times more than lady in the sample space. Based on the result it occurs to me that gender is likely to be a useless result in the sample.

(2) Properties of the loan (such as: instalments and rates):

As for these two variables, data in instalments is fairly clean, and we can easily get the statistics of instalments. However, there are three records (0.06%) missing in rates. Since the number of missing records is limited, we will simply fill the value with mean value of the sample.

(3) Credit score (“creditlevelsbuyer”, “tencentscore”,”gaodescore”)

There are three types of credit score, including “creditlevelsbuyer”, “tencentscore” and “gaodescore”. Among three types of credit score, the first type is most dirty, with about 20% missing data. Besides, it ranges from 0 to 1830, leading to a high variance of sample. Therefore, we need to fill the null value in this column.

(4) Digital footprint (such as “highcontact”)

The column “highcontact” stands for if the borrower has frequent contact records. I create a dummy variable for this column. There are total 5000 observations, including similarly same number of two types of sample.

(5) Loan status (“default”, “deal”)

There are 5000 observations in deal column while only 2205 in column default. These two variables are originally labels so I create dummy variable to represent the original variables.

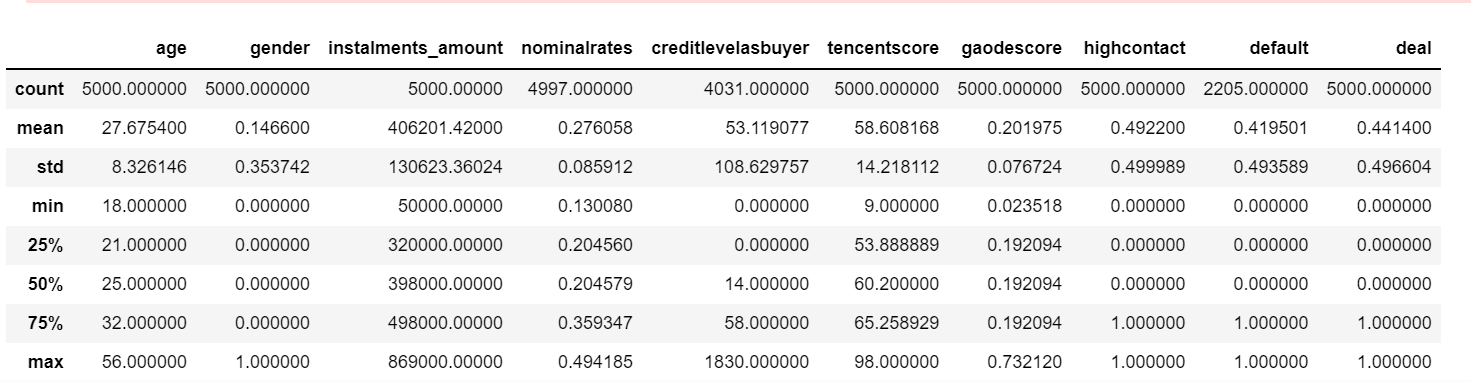


Table 1-1 Feature summary

## 2.Data Cleaning

1. Creditlevelasbuyer

I firstly conduct the correlation test on it and get the result in Figure (2), then I select highly related variables involvingTENCENT score, GAODE score, HUABEI balance and HUABEI amount and conduct a linear regression on the data.

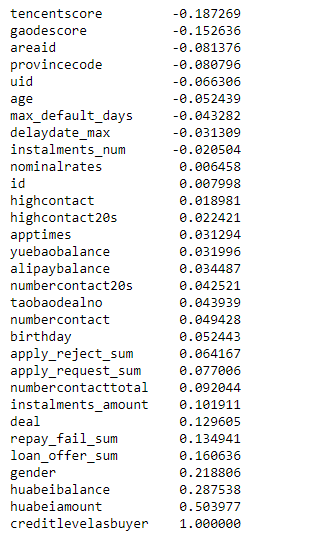
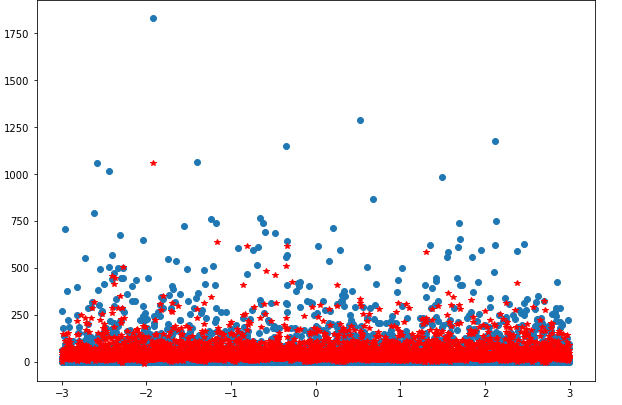


Table 2-1 Correlation test on credit score

Nevertheless, the regression result is fairly pleasant compared to using average value. Although due to its high variance the regression result can only cover part of special values, it is till has better performance than average number.



Graph 2-1 Regression result on Credit score

1. Default

Default variable will work as dependent variable later, since its data type is Boolean, we create dummy variables for default.

## 3.Logit regression on single variables

1. Default vs Credit score.

There are above half of the Default records are missing. Because it is the dependent variable, we just simply drop the missing records.

We run logit regression on three credit scores and default variable.Then we get the result:

|  |  |  |  |
| --- | --- | --- | --- |
|  | creditlevelsbuyer | tencent score | gaode score |
| coef | -0.0004 | 0.0092 | 1.9983 |
| p-value | 0.357 | 0.001 | 0.001 |

Table 3-1 Default likelihood regression

From the result we can come to a conclusion that, “creditlevelsbuyer” is useless since its p-value is up to 0.357. At the same time the “tencentscore” and “gaodescore” have a low p-value. And their coefficient is more than 0. It indicates that when someone has a higher score he is more likely to default.

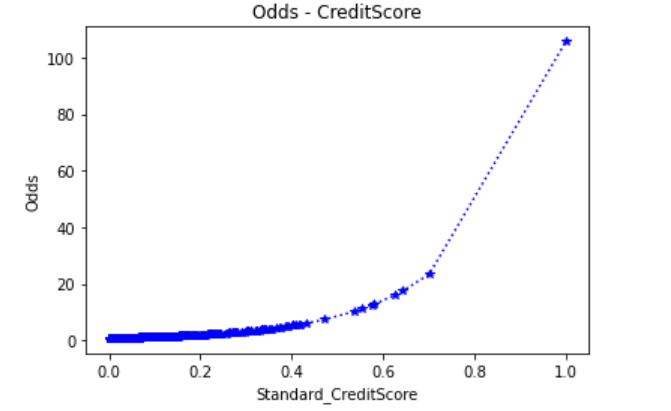
1. Deal vs Credit score.

We conduct logit regression on three types of credit score and deal variable. Different from the result on default, “creditlevelsbuyer” now has a low p-value, with a coefficient at 0.0030.

|  |  |  |  |
| --- | --- | --- | --- |
|  | creditlevelsbuyer | tencent score | gaode score |
| coef | 0.0030 | -0.0317 | -2.9408 |
| p-value | 0.000 | 0.000 | 0.000 |

Table 3-2 Deal likelihood regression (credit score)

To illustrate the result better, I illustrate odds and with standardized credit score. From the line, we learn that when the individual’s credit score is higher, his application for loan is more likely to be approved. Further, compared its influence on the likelihood of default, it obviously influenced the likelihood of approval much stronger, ranging from odds at 0 to 100.



Graph 3-1 Odds -standard credit score

(3) Deal vs frequent contact.

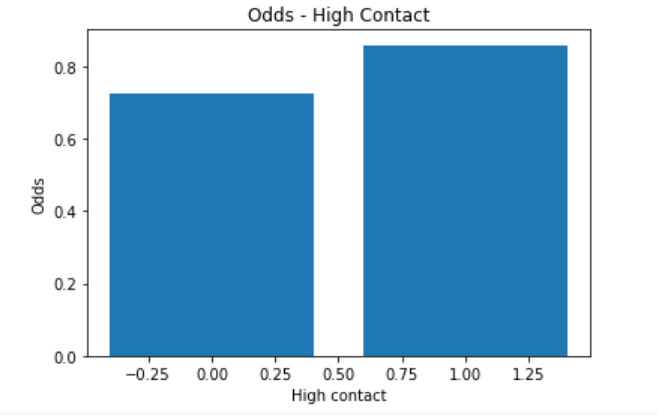
There are two types of values (True/False) in the contact variable and true value stands for the object has a frequent contact. Based on that, we only need to create one dummy variable for contact variable (0: False, 1: True).

After we run the logit regression on these two variables, we got the coefficient 0.1679, and a p-value at 0.003. Therefore, when an individual has a high frequency of contact, the likelihood of approval of his loan would be higher.

|  |  |
| --- | --- |
|  | High-contact |
| COEF | 0.1679 |
| P-VALUE | 0.003 |

Table 3-3 Deal likelihood regression (High-contact)

According to the line, we can learn that when some has a frequent contact his loan is slightly more likely to be approved. Nevertheless, two value of odds is less than 1 which means their loans is more likely to be rejected. In other words, contact frequency only has small connection with the likelihood of loan approval.



Graph 3-2 Odds -High contact

## 4.Logit regression on multiple variables

The two logit regression models on single variable both end up at a pretty low score. Therefore, I then involve more variables to enhance the model.

|  |  |
| --- | --- |
| model | score |
| deal~credit score | 0.5794 |
| deal~high-contact | 0.5586 |

Table 4-1 Model score on single feature

Since there are many variables are available in the data sample, I simply run a correlation test on Deal in the sample. Then I pick variables which has high absolute value of correlation. To be more specific I pick variables rank high in absolute value of correlation and reaches at least 0.1, including: “apptimes”,”huabeiamount”,”huabeibalance”,”nominalrates”,”credit\_score”,”tencentscore”,”gaodescore”.

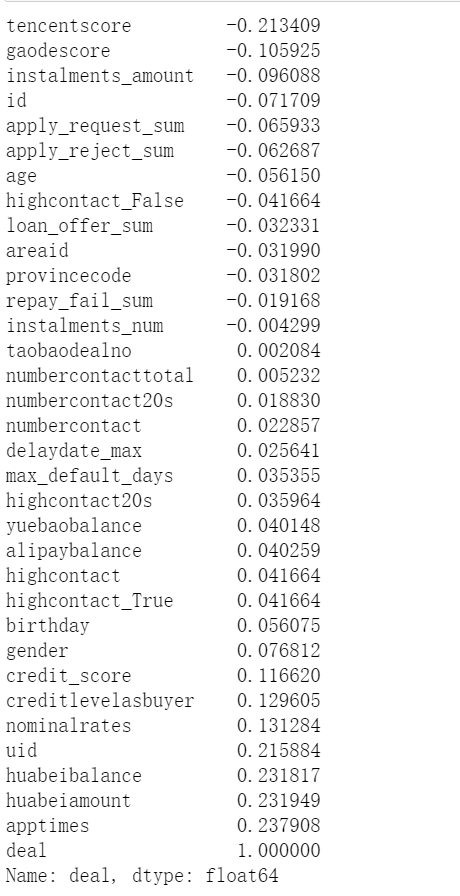


Table 4-2 Correlation test on Deal

Next, I run the logit regression on the sample, the model has much better performance, rating 0.7016. The coefficients are shown in the table. According to the figure, the p-value of credit score is significantly big, so I replace it with another variable to improve the model.

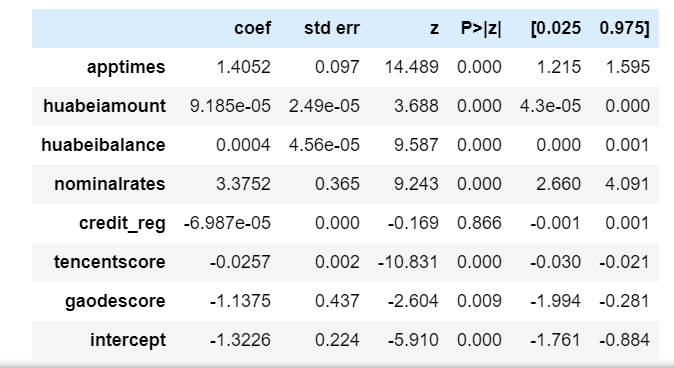


Table 4-3 Deal likelihood regression on multiple features

To pick new variable in variable left, I run a correlation test on the most related variable “apptimes”. I then use the variable which is least relevant to “apptimes” to involve new data angle into the regression. Therefore, I replace “credit score” the useless variable with “gender” the least relevant variable to “apptimes”

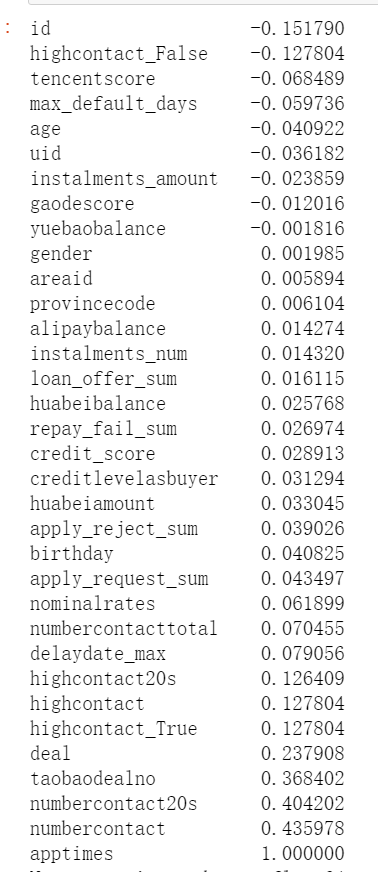


Table 4-4 Correlation test on app-times

After I rerun the logit regression with new features. To control the number of features, I replace the useless feature (“credit score”) with new feature (“Gender”). And the model improves with score reaching 0.714.

|  |  |
| --- | --- |
| Model | Score |
| model ~ (credit-score```) | 0.7016 |
| model ~(gender```) | 0.714 |
| difference | 0.0124 |

Table 4-5 Model scores comparison

The new coefficient result:



Table 4-6 Deal likelihood regression on replaced features

Now we found all variables included in the model are grouped with a low p-value. And the model peaks the score at 0.714.

## 5.Appendix

# Content

\* [1. Data Summary](#1)

\* [2. Logit Regression for one variable](#2)

\* [2.1 Default vs Credit score](#2.1)

\* [2.2 Deal vs Credit score](#2.2)

\* [2.3 Deal vs Contact](#2.3)

\* [2.4 logistic regression with multiple variables](#2.4)

import pandas as pd

import numpy as np

import math

import matplotlib.pyplot as plt

from sklearn.linear\_model import LogisticRegression

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures

from sklearn.pipeline import Pipeline

import statsmodels.api as sm

import statsmodels.formula.api as smf

<a id="1"></a>

## 1.Data Summary

data = pd.read\_csv(r"C:\Users\15161\Jupyter\_git\Fin-Tech\Assignment1\Input\MDS6212 Week 1 Data.csv",encoding="gbk")

data.head()

data.shape

data.info()

## Convert gender to 0/1

data["gender"][data["gender"]==False] = 0

data["gender"][data["gender"]==True] = 1

data["gender"].value\_counts()

data.describe()

data = pd.get\_dummies(data,columns=["highcontact"])

## Convert default to 0/1

data["default"][data["default"]==False] = 0

data["default"][data["default"]==True] = 1

data['highcontact'] = data["highcontact\_True"]

data["default"] = data["default"].astype("float64")

data[["age","gender","instalments\_amount","nominalrates","creditlevelasbuyer","tencentscore","gaodescore","highcontact","default","deal"]].describe()

data.corr()['creditlevelasbuyer'].sort\_values()

data[["gaodescore","tencentscore","huabeiamount","huabeibalance","gender"]].describe()

clf = LinearRegression()

Y = data["creditlevelasbuyer"][data["creditlevelasbuyer"].notnull()]

X = data[["gaodescore","tencentscore","huabeiamount","huabeibalance"]][data["creditlevelasbuyer"].notnull()]

clf.fit(X,Y)

clf.score(X,Y)

x = np.random.uniform(-3,3,size=len(Y))

y\_pre = clf.predict(X)

plt.figure(figsize=[10, 7])

plt.scatter(x,Y)

plt.plot(x,y\_pre,"r\*")

plt.show()

data["credit\_reg"] =data["creditlevelasbuyer"]

X\_P = data[["gaodescore","tencentscore","huabeiamount","huabeibalance"]][data["creditlevelasbuyer"].isnull()]

data["credit\_reg"][data["creditlevelasbuyer"].isnull()] = clf.predict(X\_P)

## Using poly regression to fill na value in credit score

# model = Pipeline(

# [

# ('poly',PolynomialFeatures(degree=3)),

# ('linear',LinearRegression(fit\_intercept=False))

# ]

# )

# model =model.fit(X,Y)

# model.score(X,Y)

# x = np.random.uniform(-3,3,size=len(Y))

# y\_pre = model.predict(X)

# plt.figure(figsize=[10, 7])

# plt.scatter(x,Y)

# plt.plot(x,y\_pre,"r:\*")

# plt.show()

<a id="2"></a>

## 2Logit Regression

<a id="2.1"> </a>

### 2.1 Default vs Credit score

## drop null value records in default since it is the target variables

data\_default = data[data['default'].notnull()]

data\_default.shape

## Fill the void with mean

data\_default['creditlevelasbuyer'][data\_default['creditlevelasbuyer'].isnull()] = data\_default['creditlevelasbuyer'].mean()

## Fill the void with other related variables

# data\_default.corr()['creditlevelasbuyer']

## Defin X and Y

Y = data\_default['default']

X = np.array(data\_default['creditlevelasbuyer']).reshape(-1,1)

Y = Y.astype('int')

lg = LogisticRegression()

lg.fit(X,Y)

lg.score(X,Y)

lg.coef\_

X\_sort = data\_default['creditlevelasbuyer'].sort\_values()

X\_sort = np.array(X\_sort).reshape(-1,1)

P\_1 = lg.predict\_proba(X\_sort)

P\_1

odds =[]

for i in range(len(P\_1)):

odds.append(P\_1[i][1]/P\_1[i][0])

odds

plt.plot(X\_sort,odds)

plt.ylabel("Odds")

plt.xlabel("CreditScore")

plt.title("Odds(Default) - CreditScore")

plt.show()

data\_default['intercept'] = 1.0

# model = sm.GLM.from\_formula("default ~ creditlevelasbuyer", family = sm.families.Binomial(), data = data\_default)

model = sm.Logit(data\_default['default'].astype('int'),data\_default[['creditlevelasbuyer','intercept']])

result = model.fit()

result.summary()

model = sm.Logit(data\_default['default'].astype('int'),data\_default[['gaodescore','intercept']])

result = model.fit()

result.summary()

model = sm.Logit(data\_default['default'].astype('int'),data\_default[['tencentscore','intercept']])

result = model.fit()

result.summary()

model = sm.Logit(data\_default['default'].astype('int'),data\_default[['gaodescore','intercept']])

result = model.fit()

result.summary()

<a id="2.2"> </a>

### 2.2 deal vs Credit score

## Fill na

data['credit\_score'] = data['creditlevelasbuyer']

data['credit\_score'][data['credit\_score'].isnull()] = data["credit\_score"].mean()

Y2 = data["deal"]

X2 = np.array(data["credit\_reg"]).reshape(-1,1)

data['intercept'] = 1.0

model = sm.Logit(data['deal'].astype('int'),data[['credit\_reg','intercept']])

result = model.fit()

result.summary()

model = sm.Logit(data['deal'].astype('int'),data[['tencentscore','intercept']])

result = model.fit()

result.summary()

model = sm.Logit(data['deal'].astype('int'),data[['gaodescore','intercept']])

result = model.fit()

result.summary()

lg2 = LogisticRegression()

lg2.fit(X2,Y2)

lg2.score(X2,Y2)

lg2.coef\_

X2\_sort = data["credit\_reg"].sort\_values()

## standardlize

X2\_plot = (X2\_sort-X2\_sort.min())/(X2\_sort.max()-X2\_sort.min())

X2\_sort = np.array(X2\_sort).reshape(-1,1)

P\_2 = lg2.predict\_proba(X2\_sort)

P\_2

odds =[]

for i in range(len(P\_2)):

odds.append(P\_2[i][1]/P\_2[i][0])

odds

plt.plot(X2\_plot,odds,"b:\*")

plt.ylabel("Odds")

plt.xlabel("Standard\_CreditScore")

plt.title("Odds - CreditScore")

plt.show()

<a id="2.3"> </a>

### 2.3 Deal vs Contact

## See if there is null value in column contact

data["highcontact"].value\_counts(dropna=False)

X3 = np.array(data["highcontact"]).reshape(-1,1)

Y3 = data["deal"]

model = sm.Logit(data['deal'].astype('int'),data[['highcontact','intercept']])

result = model.fit()

result.summary()

lg3 = LogisticRegression()

lg3.fit(X3,Y3)

lg3.score(X3,Y3)

lg3.coef\_

P\_3 = lg3.predict\_proba(X3)

P\_3

odds =[]

for i in range(len(P\_3)):

odds.append(P\_3[i][1]/P\_3[i][0])

odds

pd.Series(odds).describe()

X\_contact = [0,1]

Y\_Odds = [odds[0],odds[1]]

## not useful,only value < 1 means with only high contact, no matter what its value is it will predict it as not a fail deal

plt.bar(X\_contact, Y\_Odds)

plt.ylabel("Odds")

plt.xlabel("High contact")

plt.title("Odds - High Contact")

plt.show()

<a id="2.4"> </a>

## 2.4 Logist regression with multiple variables

data.corr()["deal"].sort\_values()

## take variables: tencent score,gaodescore,apptime,huabeiamount,huabeibalance,nominalrates,credit\_score

data["nominalrates"][data["nominalrates"].isnull()] = data["nominalrates"].mean()

Y = data["deal"]

X = data[["apptimes","huabeiamount","huabeibalance","nominalrates","credit\_reg","tencentscore","gaodescore"]]

model = sm.Logit(data['deal'].astype('int'),data[["apptimes","huabeiamount","huabeibalance","nominalrates","credit\_reg","tencentscore","gaodescore",'intercept']])

result = model.fit()

result.summary()

lg4 = LogisticRegression()

lg4.fit(X,Y)

lg4.score(X,Y)

lg4.coef\_[0]

index = ["apptimes","huabeiamount","huabeibalance","nominalrates","credit\_score","tencentscore","gaodescore"]

model = sm.Logit(data['deal'].astype('int'),data[["apptimes","huabeiamount","huabeibalance","nominalrates","gender","tencentscore","gaodescore",'intercept']])

result = model.fit()

result.summary()

Y = data["deal"]

X = data[["apptimes","huabeibalance","huabeibalance","nominalrates","tencentscore","gaodescore","gender"]]

lg = LogisticRegression()

lg.fit(X,Y)

lg.score(X,Y)