

**ANL252\_Python for Data Analytics**

**End-of-Course Assessment– July Semester 2022**

**Student PI No: E2210907**

**Full Name: Lynn Tan Ee Leng**

**Submission Date: 4 September 2022**

1.

The categorical data are: RATING, GENDER, EDUCATION, MARITAL, AGE, S(n) and the numerical data are: LIMIT, BALANCE, INCOME, B(n), and R(n).

2.

The first data-processing step is to remove “$” and “,” from the variable R3, so that the data can be used for numerical computations:

#first data processing

df['R3'] = df['R3'].str.replace('$', '')

df['R3'] = df['R3'].str.replace(',', '')

The second data-processing step is to convert variables into categorical type, R3 into integer and ID into string:

#second data processing

to\_be\_converted = ['S1', 'S2', 'S3', 'S4', 'S5', 'GENDER', 'EDUCATION', 'MARITAL', 'RATING']

for element in to\_be\_converted:

df[element] = df[element].astype('category')

df['R3'] = df['R3'].astype(int)

df['ID'] = df['ID'].astype(str)

The third data-processing step is to remove the entire row for which age contains -1 and 199, then reset the column since there is a removal of rows:

#third data processing

df['AGE'] = df['AGE'].astype('str')

df = df[df['AGE'].str.contains('199') == False]

df = df[df['AGE'].str.contains('-1') == False]

df['AGE'] = df['AGE'].astype('int')

df.reset\_index(inplace = True)

df.describe()

The fourth data-processing step is to replace the nulls in marital and education with 0, which is the same thing as ‘others’:

#fourth data processing

df['MARITAL'] = df['MARITAL'].fillna(0)

df['EDUCATION'] = df['EDUCATION'].fillna(0)

df.info()

3.

The first insight was discovered using the command:

#first insight

df.corr()

which I realized that income and limit have a strong correlation. When plotting into the graph, it indeed does have a very strong positive correlation:

import matplotlib.pyplot as plt

df.plot(x = 'INCOME', y = 'LIMIT', kind = 'scatter')

plt.show()

Chart, scatter chart

Description automatically generated

For second insight, I did a groupby on the variable S1:

#second insight

df.groupby('S1').mean() #(2)

which I then realized an interesting thing, that people of high incomes tend to either pay on time or have the longest delay in payment, which is evident from the codes and graph below:

plt.plot(df.groupby('S1').mean().index, df.groupby('S1').mean()['INCOME'])

plt.show()

Chart, line chart

Description automatically generated

For the third insight, a groupby is done on ‘RATING’, which led to discover that age has no correlation with ‘RATING’, however, if someone has a higher income, his or her rating will be higher, along with limit and balance:

#third insight

df.groupby('RATING').mean() #(3)

As for the fourth insight, I decided to do clustering on ‘AGE’ and ‘BALANCE’, using only 3 clusters. The following codes are:

#forth insight: now lets investigate the cluster for age vs balance, but have to remove the age 199

from sklearn.cluster import KMeans

from sklearn.preprocessing import MinMaxScaler

from matplotlib import pyplot as plt

%matplotlib inline

plt.scatter(df['AGE'],df['BALANCE'])

plt.xlabel('Age')

plt.ylabel('Balance')

km = KMeans(n\_clusters=3)

y\_predicted = km.fit\_predict(df[['AGE', 'BALANCE']])

y\_predicted

df['cluster']=y\_predicted

df.head()

df1 = df[df.cluster==0]

df2 = df[df.cluster==1]

df3 = df[df.cluster==2]

plt.scatter(df1.AGE,df1['BALANCE'],color='green')

plt.scatter(df2.AGE,df2['BALANCE'],color='red')

plt.scatter(df3.AGE,df3['BALANCE'],color='black')

plt.scatter(km.cluster\_centers\_[:,0],km.cluster\_centers\_[:,1],color='purple',marker='\*',label='centroid')

plt.xlabel('Age')

plt.ylabel('Balance')

plt.legend()

which produces the graph:

Chart, scatter chart

Description automatically generated

This graph is interesting as the shape is almost like a pyramid. It suggests that people who are middle-aged tend to spend a lot more due to this group having the highest balance. On the other hand, the group that has the least balance tends to be people from all age groups.

As for the fifth insight, I did a groupby on ‘EDUCATION’:

#fifth insight

df.groupby('EDUCATION').mean()

while ignoring ‘0’ for education as I am unsure what is it, I discovered that as one’s education level increases, their income will increase which translates to having a higher limit. However, those with tertiary education spend more compared to those with postgraduate education, despite their lower limit.

4.

Using the following code to find the best correlation to obtain the correlation formula:

#Question 4: Finding the best variable to predict B1

df.corr()

I realized that balance and B1 have the strongest positive correlation.

5.

Using the following codes to train and test the data, and then plot them into the correlation graph:

#Appeared that balance is the best to predict B1, due to the strongest positive correlation

import numpy as np

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

X = np.array(df['BALANCE']).reshape(-1, 1)

y = np.array(df['B1']).reshape(-1, 1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.33)

regr = LinearRegression()

regr.fit(X\_train, y\_train)

y\_pred = regr.predict(X\_test)

plt.scatter(X\_test, y\_test, color ='g')

plt.plot(X\_test, y\_pred, color ='b')

plt.title("Balance against B1")

plt.show()

print("The score correlation score is " + str(regr.score(X\_test, y\_test)))

Chart, scatter chart

Description automatically generated

Knowing that the correlation score is 0.90, affirms the choice of picking balance against B1. The following codes will find out the y-intercept and gradient:

lm = LinearRegression()

lm.fit(X\_train, y\_train)

print("The y intercept is: ", lm.intercept\_)

print("The gradient is: ", lm.coef\_)

Therefore, the equation is:

Y = 5.27x + 2025