

**ANL252 (Online)**

**Python for Data Analytics**

# **End-of-Course Assignment**

**July 2022 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Chan Teck Fong** | **W2070448** |

**Tutorial Group: ­­­­­­­­­­­­ TV 09**

**Instructor’s Name: Dr. Munish Kumar**

**Submission Date: 04/09/2022**

**Section A**

**Question 1**

Categorical variables refer to discrete variables which can be used to represent qualitative results by classifying or categorizing observations into certain groups (Eric, 2021). Meanwhile, numeric variables are measurements with value. Below is the list of categorical and numerical variables from the dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| **S/n** | **Variables** | **Type of variables** | |
| **Categorical** | **Numeric** |
| 1 | ID | ✓ |  |
| 2 | Limit |  | ✓ |
| 3 | Balance |  | ✓ |
| 4 | Income |  | ✓ |
| 5 | Rating | ✓ |  |
| 6 | Gender | ✓ |  |
| 7 | Education | ✓ |  |
| 8 | Marital | ✓ |  |
| 9 | Age |  | ✓ |
| 10 | S(n) – S1, S2, S3, S4, S5 | ✓ |  |
| 11 | B(n) – B1, B2, B3, B4, B5 |  | ✓ |
| 12 | R(n) – R1, R2, R3, R4, R5 |  | ✓ |

**Question 2**

Four data pre-processing tasks conducted:

i. Data cleaning

Data cleaning is required to perform to remove incomplete or missing data. There are total of 51 null objects detected from dataset, 13 from education column, 38 from marital column. The percentage of null objects from dataset (51/18769, 0.3%) is insignificant, hence, elimination method is choosing as correction for missing data. See below for Python coding for data cleaning.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("C:\\Users\\Candice\\Desktop\\Python practice\\ECA\_data.csv")

df.shape

i. Data cleaning

df.isnull().sum()

# check the null objects from dataset

df1 = df.dropna()

# remove rows which have null objects

df1.isnull().sum() # to verify whether the null objects have been removed

ii. Handling extremes/ outlier data

Outliers will impact the statistical estimation. Hence, it is required to work on it. After completed remove null objects from task (i), we need to generate statistical summary to review the data performance. For Age variable, 5 minimum values of -1 and 5 maximum values of 199 are extreme outliers in the dataset. The percentage of outliers from dataset (10/18720, 0.05%) is insignificant, hence, elimination method is choosing as correction for outlier’s data. See below for Python coding for handling outlier data.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("C:\\Users\\Candice\\Desktop\\Python practice\\ECA\_data.csv")

df.shape

i. Handling outlier data

df1.describe()

# Find the statistical summary from dataset

# From the statistical summary above, age -1 and age 199 are the outliers.

df1['AGE'].value\_counts()[199]

# count the total number of occurrence of age "199" in the dataset

df1['AGE'].value\_counts()[-1]

# count the total number of occurrence of age "-1" in the dataset

# Completely remove the 10 outliers’ data points consider 'AGE' variable had a minimum of -1 and a maximum value of 199.

index = df1[(df1['AGE'] >= 199) | (df1['AGE'] <= -1)].index # creates an index for all the data points where age takes these 2 values

df1.drop(index, inplace=True) # remove these index rows from data

df1['AGE'].describe() # print summary statistics for the variable to ensure it has been removed

iii. Data Discretisation

Discretisation or binning method is chosen without sacrificing the quality of dataset results. Age group is selected to bin into 8 sub-age categories. There are 'Age <20', 'Age 20-29', 'Age 30-39', 'Age 40-49', 'Age 50-59', 'Age 60-69', 'Age 70-79', 'Age > 80'. See below for Python coding for data discretisation.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("C:\\Users\\Candice\\Desktop\\Python practice\\ECA\_data.csv")

df.shape

iii. Data discretisation

df1['AGE GROUP'] = pd.cut(df1['AGE'], bins = (0,20,30,40,50,60,70,80,100),right=False, labels =['Age <20','Age 20-29','Age 30-39','Age 40-49','Age 50-59','Age 60-69','Age 70-79','Age > 80'] )

display(df1)

iv. Data transformation

Variable with a large range (such as Income) outweigh variables with a small range (such as Age) will have impact on the results. Hence, normalisation is another data transformation method to scale down a variable. The value of a normalised variable can only within 0 to 1. See below for Python coding for data transformation.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("C:\\Users\\Candice\\Desktop\\Python practice\\ECA\_data.csv")

df.shape

iv. Data transformation

df1.describe()

# Find the statistical summary from dataset

from sklearn import preprocessing

x\_array = np.array(df1['INCOME'])

normalized\_arr = preprocessing.normalize([x\_array])

print(normalized\_arr)

**Question 3**

The given dataset was from the credit facility to analyse customer’s demographic, balance amount owed, payment history etc. The dimension of dataset comprises of 24 variables in columns and 18769 rows, type of variable generally divided into quantitative/ numeric data and categorical/ qualitative data (Bevans, 2022). From the given dataset, there are some missing objects and extremes/outliers. Hence, data cleaning & reduction process are required prior further analysis of relationship of variables. The data cleaning process, relationship or correlational of variables such as Balance, Income, Rating, Gender, Education, Age from dataset to be done by using Python coding.

i. Relationship of customer’s gender by age group

The given gender from dataset revealed that gender 0 representing male, 1 representing female.

As customer’s age ranges from 20 to 80 from dataset, it is recommended to bin into few sub-categories. Data discretisation is involved to bin age into 8 bins. There are 'Age <20', 'Age 20-29', 'Age 30-39', 'Age 40-49', 'Age 50-59', 'Age 60-69', 'Age 70-79', 'Age > 80'.

Text

Description automatically generated

Table 1: Summarize Table (Gender vs Age Group)

Chart, bar chart

Description automatically generated

Figure 1: Customer’s Gender by Age Group

Refer to Figure 1, the main customer come from the group of Age 30-39, followed by Age 20-29, Age 40-49, Age 50-59, Age 60-69, Age 70-79 and lastly Age >80. Table 1 revealed that 7043 (37.64%) customers come from the Age 30-39, 5924 (31.66%) customers from Age 20-29, and 4079 (21.80%) from Age 40-49. Another observation made was male-female ratio. Total of 11566 (61.82%) customers are from Female, and 7144 (38.18%) from male. Female customers be the dominant in credit facility.

In the adult development stages, 30’s are mostly mature adults, setting down and raising a family (Sullens, 2022). And 20’s are young adults who are having and completing their studies and begin their career. Some young adults will get married and starting a family. These could be the reasons customers from the Age group 30-39 and 20-29 become the top 2 sub-variable from Figure 1.

ii. Relationship of customer’s average income by age group

A bar chart is plotting to reflect the relationship of customer’s average current income according to the age group.

Text

Description automatically generated

Table 2: Summarize Table (Average Income vs Age Group)

Chart, bar chart

Description automatically generated

Figure 2: Customer’s Average Income by Age Group

Cross reference Table 2 and Figure 2, it revealed that the average current income ranges from 295117.64 to 82953.00. Customer from Age 70-79 draw the highest average current income at 295,117.64, followed by Age 30-39 at 208920.82, Age 60-69 at 202302.01. It could be some passive incomes generated for the Age 70-79, middle adulthood Age 30-39 more committed to their career and late adulthood Age 60-69 at stable period, getting more hobbies and explore other interest and reinvest to generate more incomes (Ayres, 2022).

iii. Customer education level

Education variable was the highest education attained by customer. It was categorized as 0 representing others, 1 representing postgraduate, 2 representing tertiary, 3 representing high school. A pie chart is plotting to reveal the ratio of education level of customer.

Chart, pie chart

Description automatically generated

Figure 3: Customer’s Education

From Figure 3, it shown that the highest education level attained by customer is 2-Tertiary at the number of 8,865 (47.38%), followed by 1-Postgraduate at 6,404 (34.23%), 3-High school at 3,107 (16.61%) and lastly 0-Others at 334 (1.79%). Tertiary and Postgraduate education customers could have different attitudes towards financial planning. Often prefer stable in their living condition, set bigger purchases in life such as buying cars, houses, pursuing their interests and societal norms etc.

iv. Relationship of customer rating by gender

There are 2 categorical variables analysis for the relationship – Rating and Gender. Customer rating with 0 representing as Good, 1 representing as Bad. Whereas Gender 0 representing Male, 1 representing Female.

Graphical user interface, application

Description automatically generated

Table 3: Summarize Table (Customer rating by gender)

Chart, bar chart

Description automatically generated

Figure 4: Customer’s Rating by Gender

Figure 4 revealed that 14,599 (78.03%) of customer are at the good rating, 4,111 (21.97%) at bad rating. Another observation made was the male-female ratio. Generally, there is a gender gap exists in the realm of customer’s rating. 9,192 (62.96%) female customer at Good rating whereas 5,407 (37.04%) at Bad rating. It could be female handle better credit than male (Hoffman, 2013).

v. Relationship of average current credit balance by customer’s education

There are 4 sub-categories from customer’s Education variable. It was categorized as 0 representing others, 1 representing postgraduate, 2 representing tertiary, 3 representing high school. A bar chart is plotting to reflect the relationship of customer’s current average credit balance and their education level.

Chart

Description automatically generated with medium confidence

Table 4: Summarize Table (Balance by Education)

Chart, bar chart

Description automatically generated

Figure 5: Customer’s Average Credit Balance by Education

From Figure 5, it revealed that customer possessed 0-Others education level has the highest credit balance (12,756.65), followed by 2-Tertiary (9,596.87), 1-Postgraduate (8,663.8) and lastly 3-High School (8,415.06). It could be High School education level customers’ more committed to settle their credit commitment rather than Postgraduate & Tertiary level who might fully maximize their income.

Below please find the Python coding for the 5 charts in question 3:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv("C:\\Users\\Candice\\Desktop\\Python practice\\ECA\_data.csv")

df.shape

# find the size of dataframe (rows, columns)

df.isnull().sum()

# check the null objects from dataset

df1 = df.dropna()

# remove rows which have null objects

df1.describe()

# Find the statistical summary from dataset

df1['AGE'].value\_counts()[199]

# count the total number of occurrence of age "199" in the dataset

df1['AGE'].value\_counts()[-1]

# count the total number of occurrence of age "-1" in the dataset

# Remove the 10 extremes/ outliers' data points for 'AGE' variable which at minimum of -1 and a maximum value of 199.

index = df1[(df1['AGE'] >= 199) | (df1['AGE'] <= -1)].index # creates an index for all the data points where age takes these 2 values

df1.drop(index, inplace=True) # remove these index rows from data

df1['AGE'].describe() # print summary statistics for the variable to ensure it has been removed

# Discretising a numeric variable into 8 bins

df1['AGE\_GROUP'] = pd.cut(df1['AGE'], bins = (0,30,40,50,60,70,80,100),right=False, labels =['Age 20-29','Age 30-39','Age 40-49','Age 50-59','Age 60-69','Age 70-79','Age > 80'] )

display(df1)

df1.head(2)

df1.columns.values

# list all the variables after data cleaning

df1.index.values # list the index after data cleaning

print(type(df1.columns))

print(type(df1.index))

df1.columns.tolist()

df1.index.tolist()

print (type(df1.columns.tolist()))

print (type(df1.index.tolist()))

i. Stack Bar Chart : Customer's Gender by Age Group

df2 = df1.groupby(['AGE\_GROUP', 'GENDER'])['GENDER'].count().unstack().fillna(0).astype(int)

df1.shape

# find the size of dataframe (rows, columns)

print(df2)

## To plot "Customer's Gender by Age"

df2.plot(kind='bar', stacked=True)

plt.title("Customer's Gender by Age Group")

plt.xlabel("Age Group")

plt.ylabel("No. of Customer")

plt.figure(figsize=(80,60))

plt.rc('xtick', labelsize= 7.5)

plt.rc('ytick', labelsize= 8)

ii. Bar Chart : Average Customer Income by Age Group

df3 = df1.groupby(['AGE\_GROUP'])['INCOME'].sum(numeric\_only=False)

search\_product\_df1 = df1.groupby(['AGE\_GROUP'])

def findavg(itemname):

selected\_items = search\_product\_df1.get\_group(itemname)

AGE\_GROUP\_count=selected\_items['INCOME'].count()

AGE\_GROUP\_total\_INC=selected\_items['INCOME'].sum()

AGE\_GROUP\_avg=round(AGE\_GROUP\_total\_INC/AGE\_GROUP\_count,2)

return AGE\_GROUP\_avg

data = {'Age 20-29':(findavg('Age 20-29')),\ 'Age 30-39':(findavg('Age 30-39')),'Age 40-49':(findavg('Age 40-49')),'Age 50-59':(findavg('Age 50-59')),\ 'Age 60-69':(findavg('Age 60-69')), 'Age 70-79':(findavg('Age 70-79')), 'Age > 80':(findavg('Age > 80'))}

AGE\_GROUP = list(data.keys())

values = list(data.values())

print(pd.DataFrame(data={'AGE\_GROUP':AGE\_GROUP, 'Average Income': values}))

# Plot average of customer's income by age group

plt.bar(AGE\_GROUP, values, color = 'blue', width = 0.5)

plt.xlabel("Age Group")

plt.ylabel("Average INCOME")

plt.title("Average of INCOME by AGE GROUP")

plt.figure(figsize=(100,60))

plt.rc('xtick', labelsize= 8)

plt.rc('ytick', labelsize= 8)

iii. Pie chart : Customer's Education

df1.EDUCATION.count()

df4 =df1['EDUCATION'].value\_counts(sort=False)

df4

# Plot pie chart for customer's education level

values = df4

labels = df1['EDUCATION'].unique().tolist()

plt.pie(df4, labels=values, startangle =90)

plt.legend(labels, loc = 3)

plt.show()

iv. Stack bar chart: Customer Rating by Gender

df5 = df1.groupby(['RATING', 'GENDER'])['GENDER'].count().unstack().fillna(0)

print(df5)

# Plot customer's rating by gender

#Gender: 0:Male, 1:Female

#Rating: 0:Good, 1:Bad

df5.plot(kind='bar', stacked=True)

plt.title("Customer's Rating by Gender")

plt.xlabel("Customer\'s Rating")

plt.ylabel("No. of Customer")

plt.xticks(rotation=0, ha='center')

v. Bar Chart: Average Balance by Customer's Education

df6 = df1.groupby(['EDUCATION'])['BALANCE'].sum(numeric\_only=False)

search\_product\_df1 = df1.groupby(['EDUCATION'])

# Find the average of balance per Education

def findavg(itemname):

selected\_items = search\_product\_df1.get\_group(itemname)

EDUCATION\_count=selected\_items['BALANCE'].count()

EDUCATION\_total\_BAL=selected\_items['BALANCE'].sum()

EDUCATION\_avg=round(EDUCATION\_total\_BAL/EDUCATION\_count,2)

return EDUCATION\_avg

EDUCATION = list(data.keys())

# Assign average value to dictionaries

data = {'0':(findavg(0)),'1':(findavg(1)),'2':(findavg(2)),'3':(findavg(3))}

print(data)

values = list(data.values())

print(pd.DataFrame(data={'Education Level': EDUCATION, 'Average Income': values}))

# Plot bar char for the average balance by customer's education level

plt.bar(EDUCATION, values, color = 'blue', width = 0.5)

plt.xlabel('Customer\'s Education')

plt.ylabel('Average Balance')

plt.title('Average Balance by Customer\'s Education')

**Question 4**

Linear regression uses to study the relationship or correlation of variables by fitting a straight line into the dataset. From the dataset given, there are 24 variables or columns. There are

'ID', 'LIMIT', 'BALANCE', 'INCOME', 'RATING', 'GENDER', 'EDUCATION', 'MARITAL', 'AGE', 'S1', 'S2', 'S3', 'S4', 'S5', 'B1', 'B2', 'B3', 'B4', 'B5', 'R1', 'R2', 'R3', 'R4', 'R5'.

The 24 variables are in numeric variable to meet linear regression criteria. Use print(df.info()) to check the types. R3 type is object, hence, need to convert to numeric type to perform linear regression. Next, we will need to perform data cleaning. We need to identify and remove blank items & extremes/ outliers. Method of eliminating is recommended as the total affected items (69 items, 0.37%) is less than 0.5% from the dataset. Variable ID is categorical, the number do not have any relationship to anything, hence, it is recommended to drop.

We will build linear regression model using Scikit-Learn, with the assumption of one variable has linear relationship with other variables. For this study, X will be the features of the dataset and y will be the target predict the variable, B1. We need to create 2 arrays for the feature and target.

X = ‘LIMIT’, 'BALANCE', 'INCOME', 'RATING', 'GENDER', 'EDUCATION', 'MARITAL', 'AGE', 'S1', 'S2', 'S3', 'S4', 'S5', 'B2', 'B3', 'B4', 'B5', 'R1', 'R2', 'R3', 'R4', 'R5'

y = B1 (predicted value)

We need to create training and testing dataset. Import sklearn.model\_selection import train\_test\_split, partition training and testing dataset at 80%, 20% ratio. Using training dataset to create fitted model. And use testing data to see the data accuracy. Metrics R-squared and RMSE are used to quantify how well the model fit to given dataset. The model has 2 attributes:

i. Coefficients which are for an array coefficient of the model

ii. Intercept which is the y-intercept of linear model

The predicted variable B1 will incur based on ‘LIMIT’, 'BALANCE', 'INCOME', 'RATING', 'GENDER', 'EDUCATION', 'MARITAL', 'AGE', 'S1', 'S2', 'S3', 'S4', 'S5', 'B2', 'B3', 'B4', 'B5', 'R1', 'R2', 'R3', 'R4', 'R5'.

Below please find the Python coding for linear regression modelling to predict the variable, B1.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv("C:\\Users\\Candice\\Desktop\\Python practice\\ECA\_data.csv")

df.shape # find the size of dataframe (rows, columns)

print(df.info()) # to check the data type

df['R3'] = pd.to\_numeric(df['R3'],errors = 'coerce') # convert variable type

print(df.info()) # print info to check again after convert R3 type

Data Cleaning process : remove blank, extremes/ outliers

df.isnull().sum() # check the null objects from dataset

df1 = df.dropna() # remove rows which have null objects

df1['AGE'].value\_counts()[199]

# count the total number of occurrence of age "199" and "-1" in the dataset

df1['AGE'].value\_counts()[-1]

# count the total number of occurrence of age "199" and "-1" in the dataset

# remove the 10 extremes/ outliers' data points consider 'AGE' variable had a minimum of -1 and a maximum value of 199.

index = df1[(df1['AGE'] >= 199) | (df1['AGE'] <= -1)].index

df1.drop(index, inplace=True) # remove these index rows from data

df1['AGE'].describe() # print summary statistics for the variable to ensure it has been removed

df1.columns.values # list all the variables after data cleaning

df1.index.values # list the index after data cleaning

print(type(df1.columns))

print(type(df1.index))

df1.columns.tolist()

df1.index.tolist()

print (type(df1.columns.tolist()))

print (type(df1.index.tolist()))

df1.drop(['ID'], axis=1, inplace=True) # VARIBALE is NO MEANING, as it an identity, no impact to predict variable

df1.head(2) # removed ID

from sklearn.linear\_model import LinearRegression

from sklearn import datasets, linear\_model

from sklearn.metrics import mean\_squared\_error, r2\_score

model = LinearRegression()

# Creating new variables

X = df1[['LIMIT', 'BALANCE', 'INCOME', 'RATING', 'GENDER', 'EDUCATION', 'MARITAL', 'AGE', 'S1', 'S2', 'S3', 'S4', 'S5', 'B1', 'B2', 'B3', 'B4', 'B5', 'R1', 'R2', 'R3', 'R4', 'R5']]

y = df1['B1']

# Splitting the data into training and testing (80%,20% ratio)

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, shuffle=True, train\_size=0.2)

# Creating a new model and fitting it

multi\_model = LinearRegression()

multi\_model.fit(X\_train, y\_train)c

# Prediction new values

predictions = multi\_model.predict(X\_test)

r2 = r2\_score(y\_test, predictions)

rmse = mean\_squared\_error(y\_test, predictions, squared=False)

print('The r2 is: ', r2)

print('The rmse is: ', rmse)

# Printing coefficients and intercept

print(multi\_model.coef\_)

print(multi\_model.intercept\_)

# Writing a function to predict limit

coefficients = multi\_model.coef\_

intercept = multi\_model.intercept\_

def calculate\_B1(LIMIT, BALANCE, INCOME, RATING, GENDER, EDUCATION, MARITAL, AGE, S1, S2, S3, S4, S5, B2, B3, B4, B5, R1, R2, R3, R4, R5):

return ((LIMIT \* coefficients[0]) + (BALANCE \* coefficients[1]) + (INCOME \* coefficients[2]) + (RATING \* coefficients[3]))\

+(GENDER \* coefficients[4]) + (EDUCATION \* coefficients[5]) +(MARITAL \* coefficients[6]) +(AGE \* coefficients[7])\

+(S1 \* coefficients[8]) +(S2 \* coefficients[9]) +(S3 \* coefficients[10]) + (S4 \* coefficients[11]) + (S5 \* coefficients[12])\

+(B2 \* coefficients[13]) +(B3 \* coefficients[14]) + (B4 \* coefficients[15]) + (B5 \* coefficients[16])\

+(R1 \* coefficients[17]) +(R2 \* coefficients[18]) +(R3 \* coefficients[19]) + (R4 \* coefficients[20]) + (R5 \* coefficients[21])\

+ intercept

**Question 5**

Generic equation of linear regression is :

Yi = α + 𝛽1𝑋i + ⋯ + εi  where Yi is dependent variable; α is intercept; 𝛽 is slope coefficient; X is independent variable; ε is random error term; n= 1, 2, 3 …n

Linear equation from question 4 is :

y, B1 = -1349.0994972916887 + (2.18734063e-02 x LIMIT)

+ (1.99460949e+00 x BALANCE) + (-1.66792751e-02 x INCOME)

+ (-1.76935453e+02 x RATING) + (1.35117397e+03 x GENDER)

+ (-3.43680274e+02 x EDUCATION) + (1.26025211e+03x MARITAL)

+ (1.77294684e+01 x AGE) + (-3.64682635e+02 x S1)

+ (-1.07074957e+02x S2) + (-1.46166713e+02 x S3) + (1.69704798e+02 x S4)

+ (-2.96253031e+02 x S5) + (6.55555619e-01 x B2) + (3.39335608e-02 x B3)

+ (-9.43495225e-02 x B4) + (8.57804953e-02 x B5) + (-5.51837149e-01 x R1)

+ (8.89583614e-04 x R2) + (4.48335260e-0 x R3) + (-8.05382128e-03 x R4)

+ (6.09038696e-02 x R5)

We can turn into a predictive function to return a predicted B1 value.

Let’s say we have a customer with total Limit of 10,000, current credit Balance at 5,000, current Income at 8,000, Rating is 0, Gender is 1, Education is 3, Marital status is 2, Age is 45, S1 is -1, S2 is 0, S3 is 0, S4 is -1, S5 is 0, B2 is 400, B3 is 400, B4 is 400, B5 is 400, R1 is 390, R2 is 390, R3 is 450, R4 is 450, R5 is 600. The predicted B1 billable amount for the month (n=1) is **13119.24** (rounded to 2 decimal places)

Python coding for linear regression modelling to predict the variable,

B1 = 13119.238642650293 or 13119.24 (rounded to 2 decimal places). See below below:

… Continue Q4 Python coding.

# Writing a function to predict limit

coefficients = multi\_model.coef\_

intercept = multi\_model.intercept\_

def calculate\_B1(LIMIT, BALANCE, INCOME, RATING, GENDER, EDUCATION, MARITAL, AGE, S1, S2, S3, S4, S5, B2, B3, B4, B5, R1, R2, R3, R4, R5):

return ((LIMIT \* coefficients[0]) + (BALANCE \* coefficients[1]) + (INCOME \* coefficients[2]) + (RATING \* coefficients[3]))\

+(GENDER \* coefficients[4]) + (EDUCATION \* coefficients[5]) +(MARITAL \* coefficients[6]) +(AGE \* coefficients[7])\

+(S1 \* coefficients[8]) +(S2 \* coefficients[9]) +(S3 \* coefficients[10]) + (S4 \* coefficients[11]) + (S5 \* coefficients[12])\

+(B2 \* coefficients[13]) +(B3 \* coefficients[14]) + (B4 \* coefficients[15]) + (B5 \* coefficients[16])\

+(R1 \* coefficients[17]) +(R2 \* coefficients[18]) +(R3 \* coefficients[19]) + (R4 \* coefficients[20]) + (R5 \* coefficients[21])\

+ intercept

# Predicted B1 value

print (calculate\_B1(10000, 5000, 8000, 0, 1, 3, 2, 45, -1, 0, 0, 0, -1, 400, 400, 400, 400, 390, 390, 450, 450, 600))

**Section B**

**Question 6**

Organization of code.

A well-structured Python coding is required for all the questions. The coding has been pasted at the end of question after explanation. Each question & file name are per below:

Question 2 – ECA\_Q2.ipynb

Question 3 - ECA\_Q3.ipynb

Question 4 - ECA\_Q4.ipynb

Question 5 - ECA\_Q4.ipynb

**References**

Bevans, R. (2022, July 21). *Types of Variables in Research & Statistics.* https://www.scribbr.com/methodology/types-of-variables/

Sullens, C. (2022, July 5). *Adult Developmental Stages: Adults in Their 30’s.* https://owlcation.com/social-sciences/Adult-Developmental-Stages-Adults-in-Their-30s

Ayres, A. (2022, June 12). *What are the Stages of Adult Developments?* https://study.com/learn/lesson/stages-of-adult-development-levinson-vaillant-neugartens-theories.html

Hoffman, C.V. (2013, May 29). *Study: Women Handle credit better than men.* https://www.cbsnews.com/news/study-women-handle-credit-better-than-men/

Datagy. (2022, January 5). *Linear Regression in Scikit-Learn (sklearn): An Introduction* https://datagy.io/python-sklearn-linear-regression/