**Rule Based Classification For Potential Income of New Customers**

**Business Problem:**

A game company wants to create new level-based customer definitions (personas) by using some features of its customers. They want to create segments according to these new customer definitions and estimate how much the company will earn from the new customers according to these segments.

The Persona.csv dataset contains the prices of the products sold by an international game company and some demographic information about the users who buy these products. The data set consists of records created in each sales transaction. This means that a user with certain demographic characteristics may have made more than one purchase.

PRICE – Customer's spending amount  
SOURCE – The type of device the customer is connecting to  
SEX – Gender of the client  
COUNTRY – Country of the customer  
AGE – Age of the customer

**1. Importing the Libraries and Uploading Dataset**

In [1]:

**import** pandas **as** pd

**import** numpy **as** np

df **=** pd**.**read\_csv("persona.csv")

df**.**head()

Out[1]:

|  | **PRICE** | **SOURCE** | **SEX** | **COUNTRY** | **AGE** |
| --- | --- | --- | --- | --- | --- |
| **0** | 39 | android | male | bra | 17 |
| **1** | 39 | android | male | bra | 17 |
| **2** | 49 | android | male | bra | 17 |
| **3** | 29 | android | male | tur | 17 |
| **4** | 49 | android | male | tur | 17 |

**2. General Information About Dataset**

In [2]:

print("############################### Shape ##################################")

print(df**.**shape)

print("############################### Types ##################################")

print(df**.**dtypes)

print("############################### Head ##################################")

print(df**.**head())

print("############################### Tail ##################################")

print(df**.**tail())

print("############################### NA ##################################")

print(df**.**isnull()**.**sum())

print("############################### Quantiles ##################################")

print(df**.**describe([0, 0.05, 0.50, 0.95, 0.99, 1])**.**T)

############################### Shape ##################################

(5000, 5)

############################### Types ##################################

PRICE int64

SOURCE object

SEX object

COUNTRY object

AGE int64

dtype: object

############################### Head ##################################

PRICE SOURCE SEX COUNTRY AGE

0 39 android male bra 17

1 39 android male bra 17

2 49 android male bra 17

3 29 android male tur 17

4 49 android male tur 17

############################### Tail ##################################

PRICE SOURCE SEX COUNTRY AGE

4995 29 android female bra 31

4996 29 android female bra 31

4997 29 android female bra 31

4998 39 android female bra 31

4999 29 android female bra 31

############################### NA ##################################

PRICE 0

SOURCE 0

SEX 0

COUNTRY 0

AGE 0

dtype: int64

############################### Quantiles ##################################

count mean std min 0% 5% 50% 95% 99% 100% \

PRICE 5000.0 34.1320 12.464897 9.0 9.0 19.0 39.0 49.0 59.0 59.0

AGE 5000.0 23.5814 8.995908 15.0 15.0 15.0 21.0 43.0 53.0 66.0

max

PRICE 59.0

AGE 66.0

**3. Let's Find Answers for Belows Questions**

**3.1 How many unique SOURCE are there? What are their frequencies?**

In [3]:

df["SOURCE"]**.**nunique() *# unique source*

Out[3]:

2

In [4]:

df["SOURCE"]**.**value\_counts() *# frequencies*

Out[4]:

android 2974

ios 2026

Name: SOURCE, dtype: int64

**3.2 How many unique PRICEs are there?**

In [5]:

df["PRICE"]**.**nunique()

Out[5]:

6

**3.3 How many sales were made from which PRICE?**

In [8]:

df**.**groupby("PRICE")["PRICE"]**.**count()

Out[8]:

PRICE

9 200

19 992

29 1305

39 1260

49 1031

59 212

Name: PRICE, dtype: int64

In [29]:

*# or*

df["PRICE"]**.**value\_counts()

Out[29]:

29 1305

39 1260

49 1031

19 992

59 212

9 200

Name: PRICE, dtype: int64

**3.4 How many sales from which country?**

In [30]:

df**.**groupby("COUNTRY")["COUNTRY"]**.**count()

Out[30]:

COUNTRY

bra 1496

can 230

deu 455

fra 303

tur 451

usa 2065

Name: COUNTRY, dtype: int64

In [31]:

*# or*

df["COUNTRY"]**.**value\_counts()

Out[31]:

usa 2065

bra 1496

deu 455

tur 451

fra 303

can 230

Name: COUNTRY, dtype: int64

**3.5 How much was earned in total from sales by country?**

In [32]:

df**.**groupby("COUNTRY")["PRICE"]**.**sum()

Out[32]:

COUNTRY

bra 51354

can 7730

deu 15485

fra 10177

tur 15689

usa 70225

Name: PRICE, dtype: int64

In [41]:

*# or let's use agg()*

df**.**groupby("COUNTRY")**.**agg({"PRICE":"sum"})

Out[41]:

|  | **PRICE** |
| --- | --- |
| **COUNTRY** |  |
| **bra** | 51354 |
| **can** | 7730 |
| **deu** | 15485 |
| **fra** | 10177 |
| **tur** | 15689 |
| **usa** | 70225 |

**3.6 What are the sales amount according to SOURCE types?**

In [42]:

df**.**groupby("SOURCE")["PRICE"]**.**sum()

Out[42]:

SOURCE

android 101636

ios 69024

Name: PRICE, dtype: int64

In [44]:

*# or let's use agg()*

df**.**groupby("SOURCE")**.**agg({"PRICE":"sum"})

Out[44]:

|  | **PRICE** |
| --- | --- |
| **SOURCE** |  |
| **android** | 101636 |
| **ios** | 69024 |

**3.7 What are the PRICE averages by country?**

In [45]:

df**.**groupby("COUNTRY")["PRICE"]**.**mean()

Out[45]:

COUNTRY

bra 34.327540

can 33.608696

deu 34.032967

fra 33.587459

tur 34.787140

usa 34.007264

Name: PRICE, dtype: float64

In [47]:

*# or let's use agg()*

df**.**groupby("COUNTRY")**.**agg({"PRICE":"mean"})

Out[47]:

|  | **PRICE** |
| --- | --- |
| **COUNTRY** |  |
| **bra** | 34.327540 |
| **can** | 33.608696 |
| **deu** | 34.032967 |
| **fra** | 33.587459 |
| **tur** | 34.787140 |
| **usa** | 34.007264 |

**3.8 What are the PRICE averages by SOURCEs?**

In [48]:

df**.**groupby("SOURCE")["PRICE"]**.**mean()

Out[48]:

SOURCE

android 34.174849

ios 34.069102

Name: PRICE, dtype: float64

In [49]:

*# or let's use agg()*

df**.**groupby("SOURCE")**.**agg({"PRICE":"mean"})

Out[49]:

|  | **PRICE** |
| --- | --- |
| **SOURCE** |  |
| **android** | 34.174849 |
| **ios** | 34.069102 |

**3.9 What are the PRICE averages in the COUNTRY - SOURCE?**

In [9]:

df**.**groupby(["COUNTRY", "SOURCE"])["PRICE"]**.**mean()

Out[9]:

COUNTRY SOURCE

bra android 34.387029

ios 34.222222

can android 33.330709

ios 33.951456

deu android 33.869888

ios 34.268817

fra android 34.312500

ios 32.776224

tur android 36.229437

ios 33.272727

usa android 33.760357

ios 34.371703

Name: PRICE, dtype: float64

In [10]:

*# or let's use agg()*

df**.**groupby(["COUNTRY", "SOURCE"])**.**agg({"PRICE":"mean"})

Out[10]:

|  |  | **PRICE** |
| --- | --- | --- |
| **COUNTRY** | **SOURCE** |  |
| **bra** | **android** | 34.387029 |
| **ios** | 34.222222 |
| **can** | **android** | 33.330709 |
| **ios** | 33.951456 |
| **deu** | **android** | 33.869888 |
| **ios** | 34.268817 |
| **fra** | **android** | 34.312500 |
| **ios** | 32.776224 |
| **tur** | **android** | 36.229437 |
| **ios** | 33.272727 |
| **usa** | **android** | 33.760357 |
| **ios** | 34.371703 |

**4. What are the average earnings by COUNTRY - SOURCE - SEX - AGE?**

To be able to create new level based customer definitions, we must calculate the average price paid by customers by COUNTRY - SOURCE - SEX - AGE first.

In [21]:

agg\_df **=** df**.**groupby(["COUNTRY", "SOURCE", "SEX", "AGE"], as\_index**=False**)["PRICE"] \

**.**mean()**.**sort\_values("PRICE", ascending**=False**, ignore\_index**=True**)

agg\_df**.**head()

Out[21]:

|  | **COUNTRY** | **SOURCE** | **SEX** | **AGE** | **PRICE** |
| --- | --- | --- | --- | --- | --- |
| **0** | bra | android | male | 46 | 59.0 |
| **1** | usa | android | male | 36 | 59.0 |
| **2** | fra | android | female | 24 | 59.0 |
| **3** | usa | ios | male | 32 | 54.0 |
| **4** | deu | android | female | 36 | 49.0 |

**5. Now convert the Age variable to a categorical variable and add it to agg\_df**

Since we are creating level-based customer definitions, we must change the type of age variable to a categorical variable. In order to do that, we must create intervals for the age variable.

In [23]:

agg\_df['AGE\_CAT'] **=** pd**.**cut(agg\_df["AGE"],bins**=**[0,18,23,30,40,70],labels**=**['0\_18','19\_23','24\_30','31\_40','41\_70'])

agg\_df

Out[23]:

|  | **COUNTRY** | **SOURCE** | **SEX** | **AGE** | **PRICE** | **AGE\_CAT** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | bra | android | male | 46 | 59.0 | 41\_70 |
| **1** | usa | android | male | 36 | 59.0 | 31\_40 |
| **2** | fra | android | female | 24 | 59.0 | 24\_30 |
| **3** | usa | ios | male | 32 | 54.0 | 31\_40 |
| **4** | deu | android | female | 36 | 49.0 | 31\_40 |
| **...** | ... | ... | ... | ... | ... | ... |
| **343** | usa | ios | female | 38 | 19.0 | 31\_40 |
| **344** | usa | ios | female | 30 | 19.0 | 24\_30 |
| **345** | can | android | female | 27 | 19.0 | 24\_30 |
| **346** | fra | android | male | 18 | 19.0 | 0\_18 |
| **347** | deu | android | male | 26 | 9.0 | 24\_30 |

348 rows × 6 columns

**6. Define new level-based customers (personas) and add them as variables to the dataset**

To create our level-based customer definition variable we will concatenate our variables

In [27]:

col\_list **=** ['COUNTRY','SOURCE','SEX','AGE\_CAT']

agg\_df['customers\_level\_based'] **=** agg\_df[col\_list]**.**apply('\_'**.**join, axis**=**1)**.**str**.**upper()

agg\_df

Out[27]:

|  | **COUNTRY** | **SOURCE** | **SEX** | **AGE** | **PRICE** | **AGE\_CAT** | **customers\_level\_based** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | bra | android | male | 46 | 59.0 | 41\_70 | BRA\_ANDROID\_MALE\_41\_70 |
| **1** | usa | android | male | 36 | 59.0 | 31\_40 | USA\_ANDROID\_MALE\_31\_40 |
| **2** | fra | android | female | 24 | 59.0 | 24\_30 | FRA\_ANDROID\_FEMALE\_24\_30 |
| **3** | usa | ios | male | 32 | 54.0 | 31\_40 | USA\_IOS\_MALE\_31\_40 |
| **4** | deu | android | female | 36 | 49.0 | 31\_40 | DEU\_ANDROID\_FEMALE\_31\_40 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **343** | usa | ios | female | 38 | 19.0 | 31\_40 | USA\_IOS\_FEMALE\_31\_40 |
| **344** | usa | ios | female | 30 | 19.0 | 24\_30 | USA\_IOS\_FEMALE\_24\_30 |
| **345** | can | android | female | 27 | 19.0 | 24\_30 | CAN\_ANDROID\_FEMALE\_24\_30 |
| **346** | fra | android | male | 18 | 19.0 | 0\_18 | FRA\_ANDROID\_MALE\_0\_18 |
| **347** | deu | android | male | 26 | 9.0 | 24\_30 | DEU\_ANDROID\_MALE\_24\_30 |

348 rows × 7 columns

Attention! After creating customers\_level\_based values with list comprehension, these values need to be unique. For example, it could be more than one of the following: USA\_ANDROID\_MALE\_0\_18. It is necessary to take them to groupby and get the price averages.

In [28]:

*# let's check*

agg\_df["customers\_level\_based"]**.**value\_counts()

Out[28]:

BRA\_ANDROID\_MALE\_24\_30 7

USA\_ANDROID\_MALE\_41\_70 7

USA\_IOS\_FEMALE\_24\_30 7

BRA\_ANDROID\_FEMALE\_24\_30 7

USA\_ANDROID\_MALE\_24\_30 7

..

TUR\_ANDROID\_MALE\_41\_70 1

CAN\_ANDROID\_MALE\_19\_23 1

TUR\_IOS\_MALE\_31\_40 1

TUR\_IOS\_MALE\_24\_30 1

CAN\_ANDROID\_FEMALE\_24\_30 1

Name: customers\_level\_based, Length: 109, dtype: int64

In [29]:

*# Then we will make it unique customers\_level\_based values with groupby and get the price averages.*

agg\_df2 **=** agg\_df**.**groupby("customers\_level\_based", as\_index **=** **False**)**.**agg({"PRICE": "mean"})

agg\_df2

Out[29]:

|  | **customers\_level\_based** | **PRICE** |
| --- | --- | --- |
| **0** | BRA\_ANDROID\_FEMALE\_0\_18 | 35.645303 |
| **1** | BRA\_ANDROID\_FEMALE\_19\_23 | 34.077340 |
| **2** | BRA\_ANDROID\_FEMALE\_24\_30 | 33.863946 |
| **3** | BRA\_ANDROID\_FEMALE\_31\_40 | 34.898326 |
| **4** | BRA\_ANDROID\_FEMALE\_41\_70 | 36.737179 |
| **...** | ... | ... |
| **104** | USA\_IOS\_MALE\_0\_18 | 33.983495 |
| **105** | USA\_IOS\_MALE\_19\_23 | 34.901872 |
| **106** | USA\_IOS\_MALE\_24\_30 | 34.838143 |
| **107** | USA\_IOS\_MALE\_31\_40 | 36.206324 |
| **108** | USA\_IOS\_MALE\_41\_70 | 35.750000 |

109 rows × 2 columns

**7. Create a new variable called SEGMENT.**

We are doing this by dividing the average price into four groups called A, B, C, and D

In [30]:

agg\_df2["SEGMENT"] **=** pd**.**qcut(agg\_df2["PRICE"], 4, labels**=**["D", "C", "B", "A"])

agg\_df2

Out[30]:

|  | **customers\_level\_based** | **PRICE** | **SEGMENT** |
| --- | --- | --- | --- |
| **0** | BRA\_ANDROID\_FEMALE\_0\_18 | 35.645303 | B |
| **1** | BRA\_ANDROID\_FEMALE\_19\_23 | 34.077340 | C |
| **2** | BRA\_ANDROID\_FEMALE\_24\_30 | 33.863946 | C |
| **3** | BRA\_ANDROID\_FEMALE\_31\_40 | 34.898326 | B |
| **4** | BRA\_ANDROID\_FEMALE\_41\_70 | 36.737179 | A |
| **...** | ... | ... | ... |
| **104** | USA\_IOS\_MALE\_0\_18 | 33.983495 | C |
| **105** | USA\_IOS\_MALE\_19\_23 | 34.901872 | B |
| **106** | USA\_IOS\_MALE\_24\_30 | 34.838143 | B |
| **107** | USA\_IOS\_MALE\_31\_40 | 36.206324 | A |
| **108** | USA\_IOS\_MALE\_41\_70 | 35.750000 | B |

109 rows × 3 columns

Let's look at some statistics of the SEGMENTs we created:

In [37]:

agg\_df2**.**groupby("SEGMENT")**.**agg({"PRICE": ['mean','min','max','std','sum','count']})**.**sort\_values("SEGMENT",ascending **=** **False**)

Out[37]:

|  | **PRICE** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
|  | **mean** | **min** | **max** | **std** | **sum** | **count** |
| **SEGMENT** |  |  |  |  |  |  |
| **A** | 38.691234 | 36.060606 | 45.428571 | 2.581762 | 1044.663328 | 27 |
| **B** | 34.999645 | 34.103727 | 36.000000 | 0.636502 | 944.990411 | 27 |
| **C** | 33.509674 | 32.500000 | 34.077340 | 0.492587 | 904.761209 | 27 |
| **D** | 29.206780 | 19.000000 | 32.333333 | 3.638037 | 817.789833 | 28 |

**8. Let's classify new customers and estimate how much revenue we will earn from them.**

In [38]:

new\_user\_list **=** ["TUR\_ANDROID\_FEMALE\_31\_40",

"FRA\_IOS\_FEMALE\_31\_40",

"BRA\_IOS\_MALE\_41\_70",

"DEU\_ANDROID\_MALE\_19\_23",

"USA\_ANDROID\_FEMALE\_24\_30",

"CAN\_IOS\_MALE\_41\_70"]

**for** user **in** new\_user\_list:

print(agg\_df2[agg\_df2["customers\_level\_based"] **==** user])

print("\n----------------------------------------------\n")

customers\_level\_based PRICE SEGMENT

72 TUR\_ANDROID\_FEMALE\_31\_40 41.833333 A

----------------------------------------------

customers\_level\_based PRICE SEGMENT

63 FRA\_IOS\_FEMALE\_31\_40 32.818182 C

----------------------------------------------

customers\_level\_based PRICE SEGMENT

19 BRA\_IOS\_MALE\_41\_70 31.083694 D

----------------------------------------------

customers\_level\_based PRICE SEGMENT

40 DEU\_ANDROID\_MALE\_19\_23 36.070707 A

----------------------------------------------

customers\_level\_based PRICE SEGMENT

91 USA\_ANDROID\_FEMALE\_24\_30 31.269981 D

----------------------------------------------

customers\_level\_based PRICE SEGMENT

33 CAN\_IOS\_MALE\_41\_70 31.0 D

----------------------------------------------