

Level Based Persona

Simple Customer Segmentation Project

Which segment does a new future customer belong to?

S. Emre Kiyak
Data Science Researcher

Outline



- The purpose of the Project
- What is the concept of Persona?
- How to target personas?
- Dataset and Exploration
- Findings and Key Points
- References

The Purpose of the Project

- Consider, investigate the concept of Persona.
- Information obtained from the dataset is aimed to use in order to make new customer definitions based on level by separating them into certain levels or categories and considering each breakout point as a persona.
- Accordingly, when a new customer shows up, determining which segment the new future customer may belong to.
- Investigating the impact of the size of the sample with respect to the Central Limit Theorem to acquire accurate and consistent results.



What is the concept of Persona?

- Persona is a Latin origin concept widely used in Marketing, CRM and analytics.
- Persona guides the **customer experience design** and utilizes making the right channel investment. It develops the ability of analysts and marketer's ability to tailor communications effectively.
- Proper persona strategy   the crucial milestone of **Customer engagement & better conversions**
- When defining a persona, the main purpose is to **get to know them as much as possible considering within a certain groups and characteristics.**
- Therefore, identifying the needs and aiming at offering appropriate products or services to personas in line with their characteristics.



How to target personas?

- Analysing the personas' interests, purchase behaviour and channel engagement by proper segmentation
- Generating tailored and initiated strategies for targeted cross-selling, upselling, and repeated possible orders for each specific persona category.
- Utilizing market research, customer insights and web - social media analytics



Dataset and Exploration

- There are two different data tables that contain the customers' characteristics and transaction information.
- While the **users.csv** table represents the characteristics of the customers, the **purchases.csv** table contains the purchasing information of the customers.
- Each user has a unique customer number (uid). The process of combining both tables (merge) can be done with the (uid) number.

users.csv

uid : Unique customer number

reg_date : Registration date

device : The type of product used by the customer. (Android, iOS)

gender : The Gender of the customer

country: Country of the customer

age : The age of the customer

purchases.csv

uid : Unique customer number

date : The date the customer made a purchase

price : The amount that Customer spent



Number of Observations : 10k

Unique Customers: 1322

Number of Variables : 8

Reading the Data

```
In[5]: users = pd.read_csv("Datasets/users.csv")
...: users.head()
...:
```

Out[5]:

| | uid | reg_date | device | gender | country | age |
|---|----------|----------------------|--------|--------|---------|-----|
| 0 | 54030035 | 2017-06-29T00:00:00Z | and | M | USA | 19 |
| 1 | 72574201 | 2018-03-05T00:00:00Z | iOS | F | TUR | 22 |
| 2 | 64187558 | 2016-02-07T00:00:00Z | iOS | M | USA | 16 |
| 3 | 92513925 | 2017-05-25T00:00:00Z | and | M | BRA | 41 |
| 4 | 99231338 | 2017-03-26T00:00:00Z | iOS | M | FRA | 59 |

```
In[6]: users.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 10000 entries, 0 to 9999
```

```
Data columns (total 6 columns):
```

| # | Column | Non-Null Count | Dtype |
|---|----------|----------------|--------|
| 0 | uid | 10000 non-null | int64 |
| 1 | reg_date | 10000 non-null | object |
| 2 | device | 10000 non-null | object |
| 3 | gender | 10000 non-null | object |
| 4 | country | 10000 non-null | object |
| 5 | age | 10000 non-null | int64 |

```
dtypes: int64(2), object(4)
```

```
memory usage: 468.9+ KB
```

```
In[7]: purchases = pd.read_csv("Datasets/purchases.csv")
...: purchases.head()
...:
```

Out[7]:

| | date | uid | price |
|---|------------|----------|-------|
| 0 | 2017-07-10 | 41195147 | 499 |
| 1 | 2017-07-15 | 41195147 | 499 |
| 2 | 2017-11-12 | 41195147 | 599 |
| 3 | 2017-09-26 | 91591874 | 299 |
| 4 | 2017-12-01 | 91591874 | 599 |

```
In[8]: purchases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 9006 entries, 0 to 9005
```

```
Data columns (total 3 columns):
```

| # | Column | Non-Null Count | Dtype |
|---|--------|----------------|--------|
| 0 | date | 9006 non-null | object |
| 1 | uid | 9006 non-null | int64 |
| 2 | price | 9006 non-null | int64 |

```
dtypes: int64(2), object(1)
```

```
memory usage: 211.2+ KB
```

1. Step: Merging the datasets according to the "uid" variable with an inner join.

```
df = purchases.merge(users, how="inner", on="uid")
df.head()
df.shape  # (9006, 8)
```

| | date | uid | price | reg_date | device | gender | country | age |
|---|------------|----------|-------|----------------------|--------|--------|---------|-----|
| 0 | 2017-07-10 | 41195147 | 499 | 2017-06-26T00:00:00Z | and | M | BRA | 17 |
| 1 | 2017-07-15 | 41195147 | 499 | 2017-06-26T00:00:00Z | and | M | BRA | 17 |
| 2 | 2017-11-12 | 41195147 | 599 | 2017-06-26T00:00:00Z | and | M | BRA | 17 |
| 3 | 2017-09-26 | 91591874 | 299 | 2017-01-05T00:00:00Z | and | M | TUR | 17 |
| 4 | 2017-12-01 | 91591874 | 599 | 2017-01-05T00:00:00Z | and | M | TUR | 17 |

2. Step: What are the total earnings in the breakdown of “country”, “device”, “gender”, “age”?

```
df.groupby(["country", "device", "gender", "age"]).agg({"price": "sum"})
agg_df = df.groupby(["country", "device", "gender", "age"]).agg({"price": "sum"}).sort_values("price", ascending=False)
agg_df.head()
agg_df.reset_index(inplace=True)
agg_df
```

| | country | device | gender | age | price |
|-----|---------|--------|--------|-----|-------|
| 0 | USA | and | M | 15 | 61550 |
| 1 | BRA | and | M | 19 | 45392 |
| 2 | DEU | iOS | F | 16 | 41602 |
| 3 | USA | and | F | 17 | 40004 |
| 4 | USA | and | M | 23 | 39802 |
| .. | ... | ... | ... | ... | ... |
| 445 | BRA | iOS | F | 34 | 199 |
| 446 | CAN | and | F | 27 | 199 |
| 447 | USA | and | F | 60 | 199 |
| 448 | BRA | iOS | M | 47 | 199 |
| 449 | DEU | and | M | 26 | 99 |

[450 rows x 5 columns]

3. Step: Converting the age variable to a categorical variable and adding to the dataset as a new variable.

```
agg_df["age"].dtype # int
agg_df["age"].value_counts()

# num to cat!
bins = [0, 19, 24, 31, 41, agg_df["age"].max()]
labels = ["0_18", "19_23", "24_30", "31_40", "41_" + str(agg_df["age"].max())]
agg_df["age_cat"] = pd.cut(agg_df["age"], bins=bins, labels=labels)
agg_df["age_cat"]
agg_df.head()
```

| | country | device | gender | age | price | age_cat |
|---|---------|--------|--------|-----|-------|---------|
| 0 | USA | and | M | 15 | 61550 | 0_18 |
| 1 | BRA | and | M | 19 | 45392 | 0_18 |
| 2 | DEU | iOS | F | 16 | 41602 | 0_18 |
| 3 | USA | and | F | 17 | 40004 | 0_18 |
| 4 | USA | and | M | 23 | 39802 | 19_23 |

4. Step: Considering the categorical breakdowns as customer groups and defining new level-based customers by combining these groups.

```
agg_df["customers_level_based"] = [col[0] + "_" + col[1].upper() + "_" + col[2] + "_" + col[-1] for col in agg_df.values]

# Alternative way:
for index, column in agg_df.iterrows():
    agg_df.loc[index, "customers_level_based"] = column["country"].upper() + "_" + column["device"].upper() + "_" + column["gender"].upper() + "_" + column["age_cat"].upper()

agg_df[["customers_level_based", "price"]]
```

```
In[4]: agg_df[["customers_level_based", "price"]]
Out[4]:
```

| | customers_level_based | price |
|----|-----------------------|-------|
| 0 | USA_AND_M_0_18 | 61550 |
| 1 | BRA_AND_M_0_18 | 45392 |
| 2 | DEU_IOS_F_0_18 | 41602 |
| 3 | USA_AND_F_0_18 | 40004 |
| 4 | USA_AND_M_19_23 | 39802 |
| .. | ... | ... |



The variable "customers_level_based" is now our new customer definition.

For example "USA_AND_M_0_18". The USA-ANDROID-MALE-0-18 class is a single customer representing one class of customers for us.

5. Step: Segmenting the new customers according to price

```
agg_df["segment"] = pd.qcut(agg_df["price"], 4, labels=["D", "C", "B", "A"])
agg_df[["customers_level_based", "price", "segment"]].head()
```

| | customers_level_based | price | segment |
|---|-----------------------|-------|---------|
| 0 | USA_AND_M_0_18 | 61550 | A |
| 1 | BRA_AND_M_0_18 | 45392 | A |
| 2 | DEU_IOS_F_0_18 | 41602 | A |
| 3 | USA_AND_F_0_18 | 40004 | A |
| 4 | USA_AND_M_19_23 | 39802 | A |

```
agg_df.groupby("segment").agg({"price": "mean"})
```

| segment | price |
|---------|--------------|
| D | 1335.096491 |
| C | 3675.504505 |
| B | 7447.812500 |
| A | 20080.150442 |

Final Question:

What segment is a 42-year-old Turkish woman who uses IOS device in? Express the segment (group) of this person according to the final analysis?

```
new_user = "TUR_IOS_F_41_75"
agg_df[agg_df["customers_level_based"] == new_user]
```

| | country | device | gender | age | price | age_cat | customers_level_based | segment |
|-----|---------|--------|--------|-----|-------|---------|-----------------------|---------|
| 377 | TUR | iOS | F | 51 | 1596 | 41_75 | TUR_IOS_F_41_75 | D |

Finding and Key Points

- It has been identified in which segment to evaluate when a new customer registered in the system.
- As a result of the analysis, a female new user between the ages of 41-75 who uses an IOS device from Turkey, belongs to segment D.
- Let's discuss the finding and its relationship with Central Limit Theorem;
 - The central limit theorem states that the arithmetic mean of a large number of independent and uniformly distributed random variables represent approximately normal distribution.
 - The principle underlying Central Limit Theorem is that a properly selected large sample is likely to resemble the population from which it is selected. The mean of the sample is distributed as a Normal Distribution for any population roughly around the mean of the population.
 - The segment turned out to be D while the number of observations is 10k. A large sample was taken, simple segmentation and rule-based classification processes have been applied and the new user was classified in D segment by its characteristics.
 - It is observed that the large samples are likely to result in a consistent mean and standard deviation according to the Central Limit theorem.
 - Consequently, taking large samples in the conclusions made is of great importance for accurate and consistent results.

References

- An, J., Kwak, H., Jung, S. G., Salminen, J., & Jansen, B. J. (2018). Customer segmentation using online platforms: isolating behavioral and demographic segments for persona creation via aggregated user data. *Social Network Analysis and Mining*, 8(1), 1-19.
- Salminen, J., Jansen, B. J., An, J., Kwak, H., & Jung, S. G. (2018). Are personas done? Evaluating their usefulness in the age of digital analytics. *Persona Studies*, 4(2), 47-65.
- <https://www.veribilimiokulu.com/bootcamp-programlari/veri-bilimci-yetistirme-programi/>
- Salminen, J., Guan, K., Jung, S. G., Chowdhury, S. A., & Jansen, B. J. (2020, April). A literature review of quantitative persona creation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1-14).
- <https://www.data-axle.com/resources/expert-qa/how-build-and-implement-personas-your-crm-strategy/#:~:text=Personas%20are%20characters%20created%20to,well%20as%20drive%20customer%20relevance.>
- Rosenblatt, M. (1956). A central limit theorem and a strong mixing condition. *Proceedings of the National Academy of Sciences of the United States of America*, 42(1), 43.

[illegible]