



**Data Glacier**

Your Deep Learning Partner

# ***FINAL PROJECT***

*Virtual Internship(30-May 2022)*

## **CUSTOMER SEGMENTATION**

# Team Members

- Ray Ng

- rayng2018@gmail.com
- Canada
- University of British Columbia
- Specialization: Data Science

- Rita Uzoka

- rita.uzoka@yahoo.com
- United Kingdom
- Sheffield Hallam University
- Specialization: Data Science

- Fatemeh Bagheri

- f.bagheri13@gmail.com
- France

# Background –G2M(cab industry) case study

- XYZ bank wants to roll out Christmas offers to their customers. But the bank does not want to roll out the same offer to all customers, instead they want to roll out personalized offers to sets of customers. If they manually start understanding the category of customer, then this will not be efficient and, they will not be able to uncover the hidden pattern in the data ( pattern which groups certain kinds of customer in one category). Bank approached ABC analytics company to solve their problem. Bank also shared information with ABC analytics that they don't want more than 5 groups as this will be inefficient for their campaign.
- Objective : ABC analytics should try to understand certain patterns in customer behaviour, which include relations between province, sex, seniority, etc. and household income. ABC must find at most five groups in which customers share common behaviours.

The analysis has been divided into four parts:

1. Description of the Datasets
2. Exploratory Data Analysis
3. Tools and Techniques
4. Implementation
5. Results
6. Conclusion

# Agenda



PROJECT  
DESCRIPTION



EDA(EXPLORATORY  
DATA ANALYSIS)



TOOLS AND  
TECHNIQUES



IMPLEMENTATION



RESULTS



CONCLUSION

# Description of the Datasets

- 48 Features
- Timeframe of the data: 1995-01-01 to 2015-12-31
- Total data points :1000000

<b>Total number of observations</b>	1000000
<b>Total number of files</b>	1
<b>Total number of features</b>	48
<b>Base format of the file</b>	.csv
<b>Size of the data</b>	154 MB

# Description of the Datasets

		fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	fecha_alta	ind_nuevo	antigueda	indrel	ult_fec_cli_1t	indrel_1mes	tiprel_1mes	indresi	indext	conyuemp
2	0	28/01/2015	1375586	N	ES	H	35	12/01/2015	0	6	1		1	A	S	N	
3	1	28/01/2015	1050611	N	ES	V	23	10/08/2012	0	35	1		1	I	S	S	
4	2	28/01/2015	1050612	N	ES	V	23	10/08/2012	0	35	1		1	I	S	N	
5	3	28/01/2015	1050613	N	ES	H	22	10/08/2012	0	35	1		1	I	S	N	
6	4	28/01/2015	1050614	N	ES	V	23	10/08/2012	0	35	1		1	A	S	N	
7	5	28/01/2015	1050615	N	ES	H	23	10/08/2012	0	35	1		1	I	S	N	
8	6	28/01/2015	1050616	N	ES	H	23	10/08/2012	0	35	1		1	I	S	N	
9	7	28/01/2015	1050617	N	ES	H	23	10/08/2012	0	35	1		1	A	S	N	
10	8	28/01/2015	1050619	N	ES	H	24	10/08/2012	0	35	1		1	I	S	N	
11	9	28/01/2015	1050620	N	ES	H	23	10/08/2012	0	35	1		1	I	S	N	
12	10	28/01/2015	1050621	N	ES	V	23	10/08/2012	0	35	1		1	I	S	N	
13	11	28/01/2015	1050622	N	ES	H	23	10/08/2012	0	35	1		1	I	S	N	
14	12	28/01/2015	1050623	N	ES	H	23	10/08/2012	0	35	1		1	A	S	N	
15	13	28/01/2015	1050624	N	ES	H	65	10/08/2012	0	35	1		1	A	S	N	
16	14	28/01/2015	1050625	N	ES	V	23	10/08/2012	0	35	1		1	A	S	N	

## Description of the dataset: What kind of data does each column hold in the raw data?

Unnamed: 0: [ 0 1 2 ... 999997 999998 999999]

fecha\_dato: ['2015-01-28' '2015-02-28']

ncodpers: [1375586 1050611 1050612 ... 1149999 1150908 1183305]

ind\_empleado: ['N' nan 'A' 'B' 'F' 'S']

pais\_residencia: ['ES' nan 'CA' 'CH' 'CL' 'IE' 'AT' 'NL' 'FR' 'GB' 'DE' 'DO' 'BE' 'AR' 'VE'  
'US' 'MX' 'BR' 'IT' 'EC' 'PE' 'CO' 'HN' 'FI' 'SE' 'AL' 'PT' 'MZ' 'CN']

## Description of the dataset: How many NA (Missing data) are in each column?

Unnamed: 0: 0

fecha\_dato: 0

ncodpers: 0

ind\_empleado: 10782

pais\_residencia: 10782

sexo: 10786

age: 10782

fecha\_alta: 10782

ind\_nuevo: 10782

antiguedad: 10782

indrel: 10782

ult\_fec\_cli\_1t: 998899

indrel\_1mes: 10782

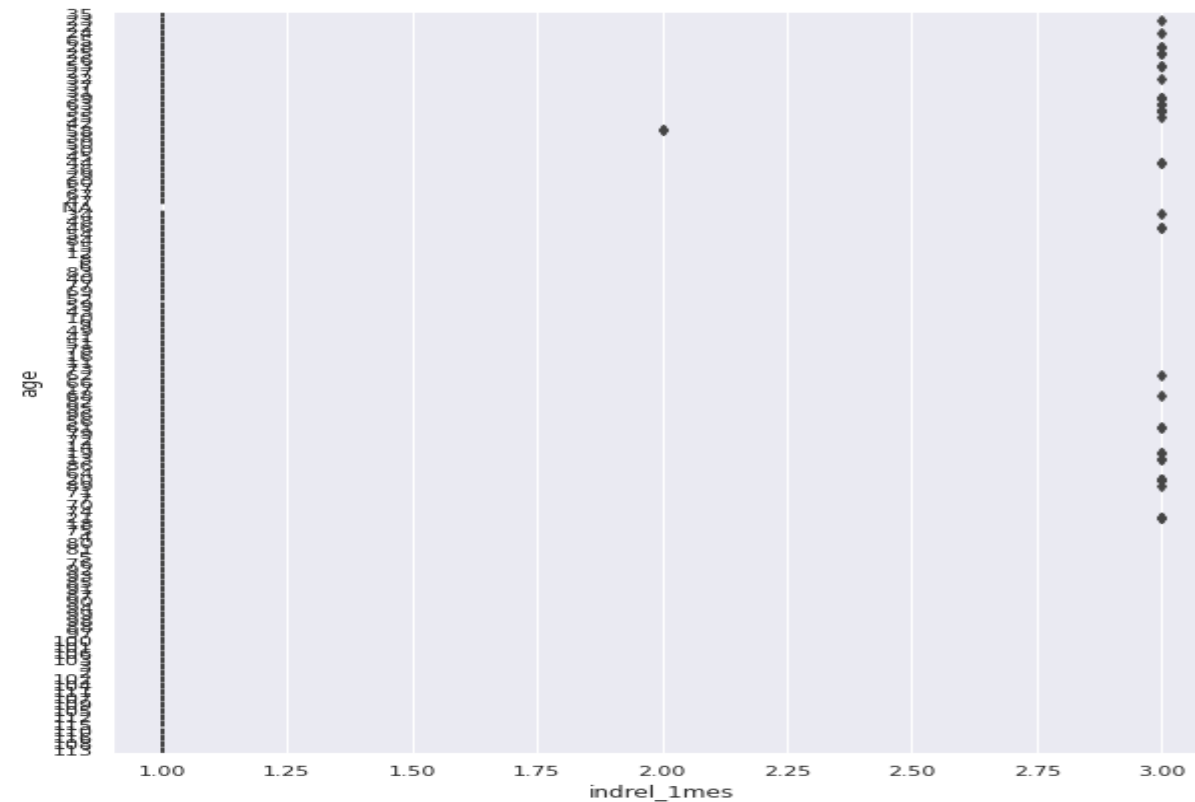
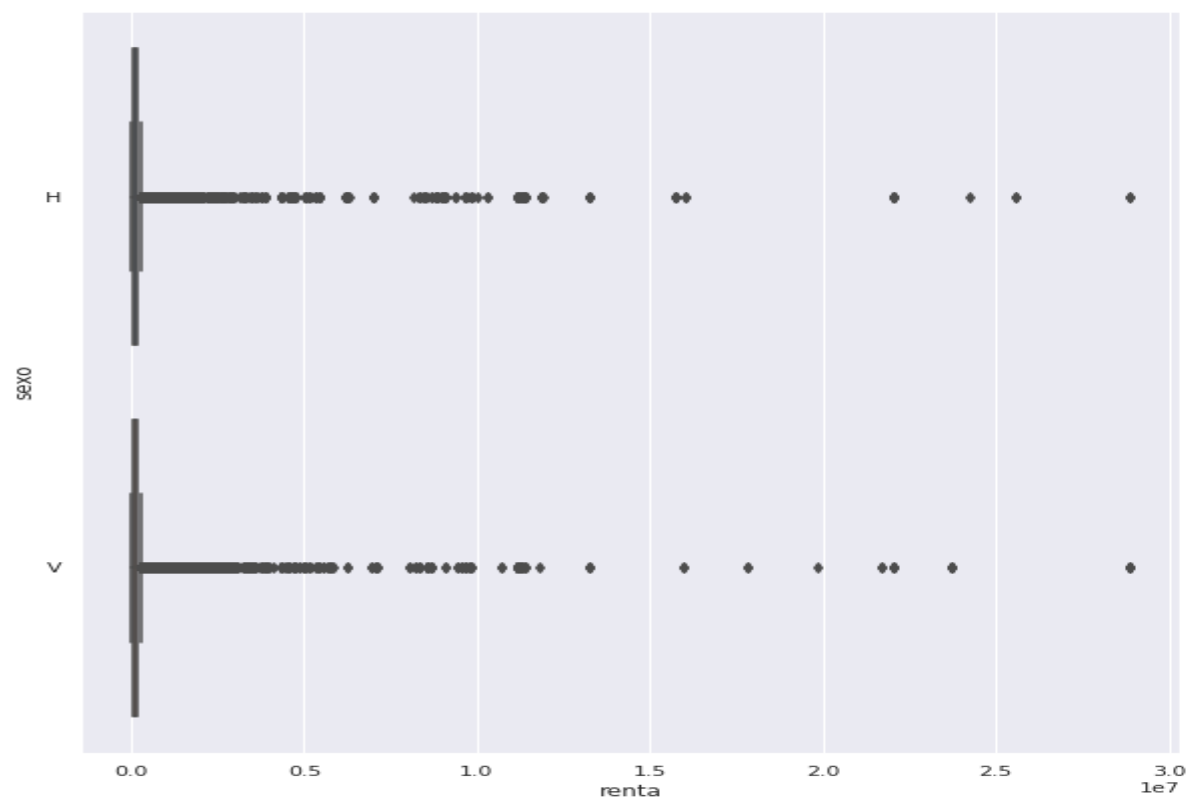


## Description of the dataset: Correlation of selected variables

- **Multivariate Correlation:** We try to find out if there is a linear relationship between the variables in the masterdata since we have more than one variable to relate with.

	Unnamed: 0	ncodpers	ind_nuevo	indrel	indrel_1mes	tipodom	cod_prov	ind_actividad_cliente	renta	ind_ahor_fin_ult1	...	ind_hi
Unnamed: 0	1.000000	-0.447119	0.008312	-0.007374	-0.000538	NaN	0.035614	0.116265	0.044528	0.005893	...	
ncodpers	-0.447119	1.000000	0.002898	0.011554	0.001253	NaN	-0.040761	-0.187022	-0.088417	-0.013469	...	
ind_nuevo	0.008312	0.002898	1.000000	0.026681	0.268051	NaN	-0.000279	0.008165	-0.000986	-0.000296	...	
indrel	-0.007374	0.011554	0.026681	1.000000	0.004462	NaN	0.001728	-0.030518	-0.000745	-0.000447	...	
indrel_1mes	-0.000538	0.001253	0.268051	0.004462	1.000000	NaN	-0.000779	-0.000859	0.000615	-0.000088	...	
tipodom	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	
cod_prov	0.035614	-0.040761	-0.000279	0.001728	-0.000779	NaN	1.000000	0.025574	-0.013720	0.001998	...	
ind_actividad_cliente	0.116265	-0.187022	0.008165	-0.030518	-0.000859	NaN	0.025574	1.000000	0.036270	0.004421	...	
renta	0.044528	-0.088417	-0.000986	-0.000745	0.000615	NaN	-0.013720	0.036270	1.000000	0.002655	...	
ind_ahor_fin_ult1	0.005893	-0.013469	-0.000296	-0.000447	-0.000088	NaN	0.001998	0.004421	0.002655	1.000000	...	
ind_aval_fin_ult1	0.002577	-0.005766	-0.000139	-0.000210	-0.000041	NaN	0.000567	0.005510	0.002259	-0.000083	...	
ind_cco_fin_ult1	-0.148813	0.249102	-0.006167	-0.001371	-0.007860	NaN	-0.020276	-0.068090	-0.023454	-0.003762	...	
ind_cder_fin_ult1	0.009754	-0.016762	-0.000540	-0.000815	-0.000160	NaN	0.000875	0.016487	0.002011	-0.000324	...	

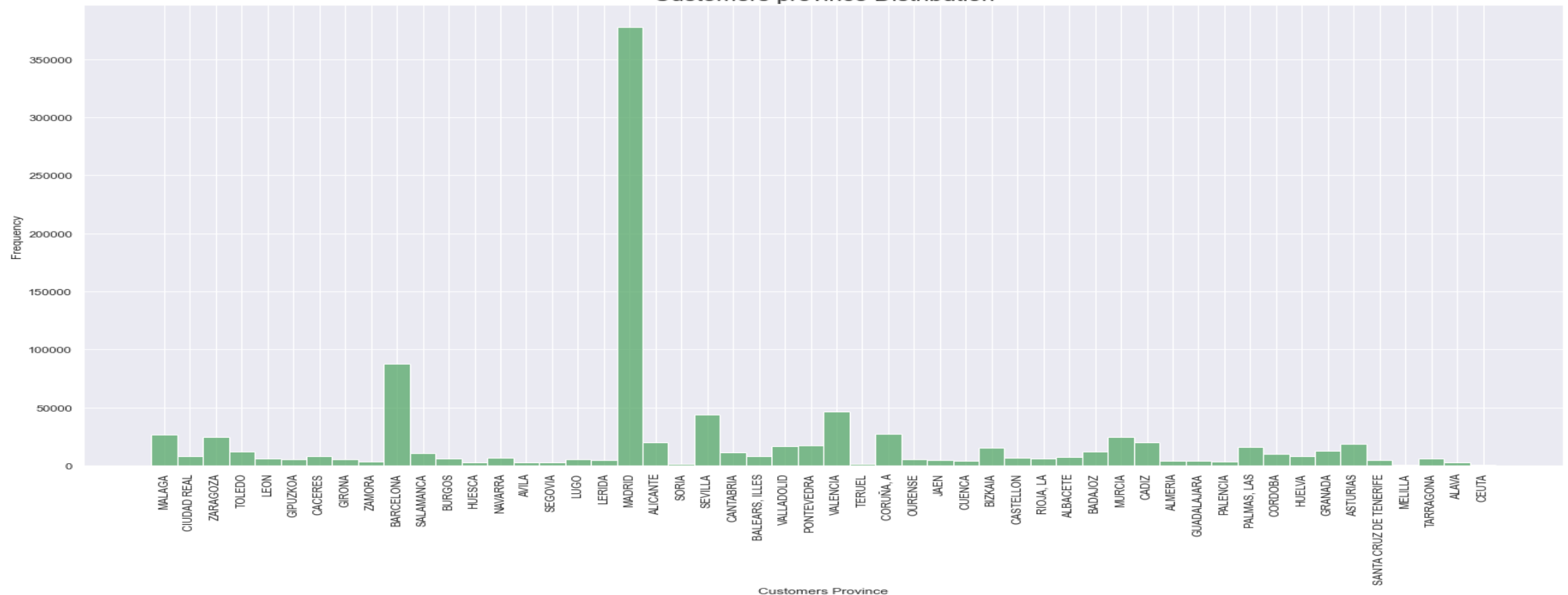
## Description of the dataset: Outliers in numerical data



## EDA: Share of customers by province

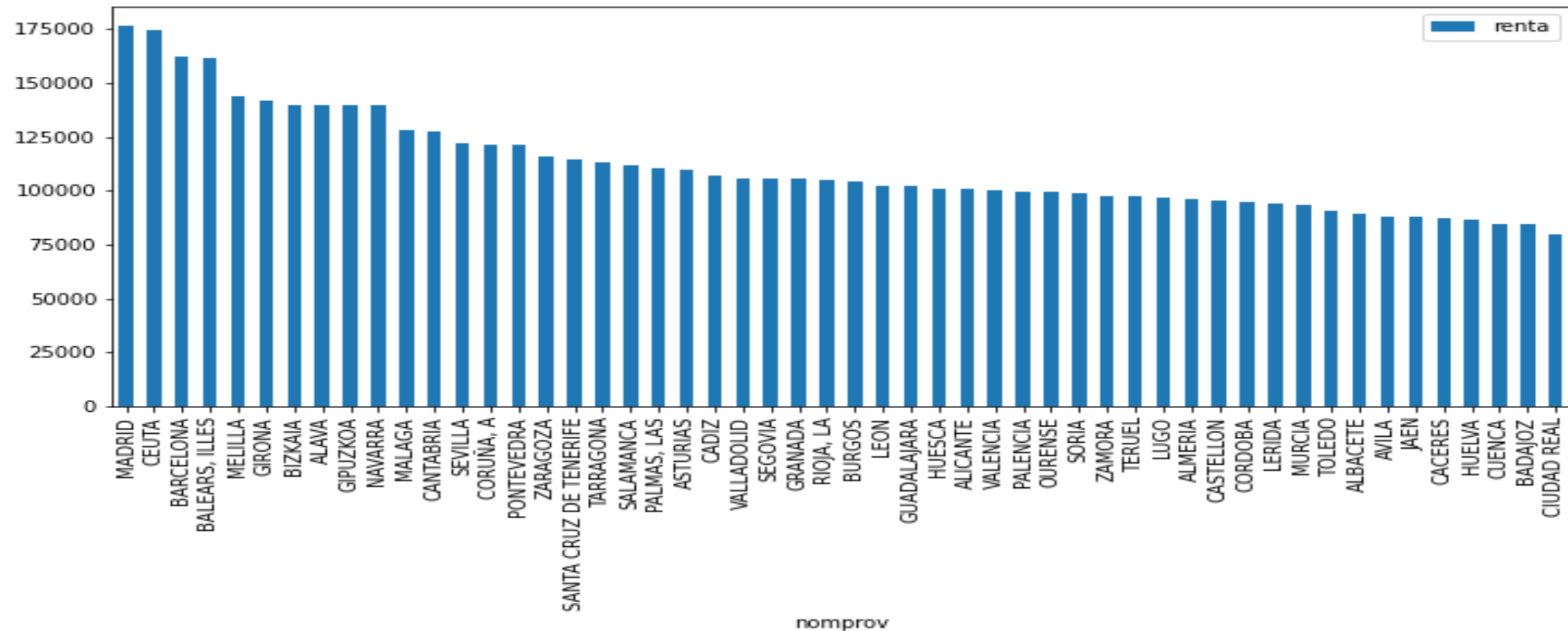
- The graph below shows that most of the customers are located in Madrid.

Customers province Distribution



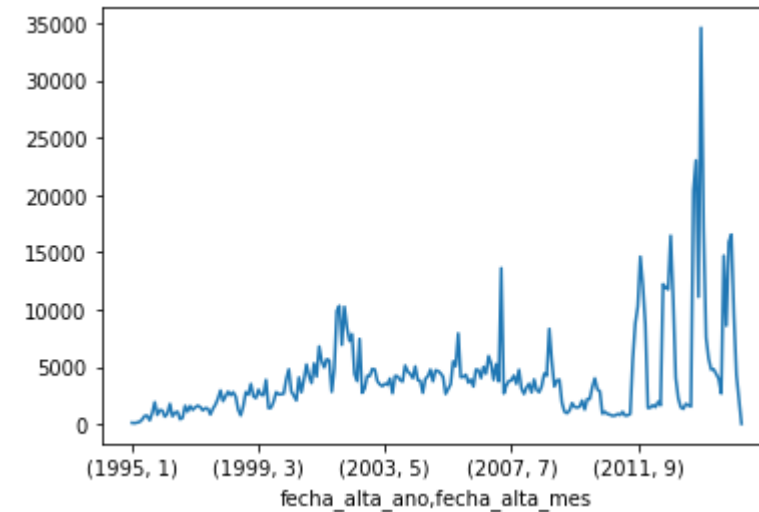
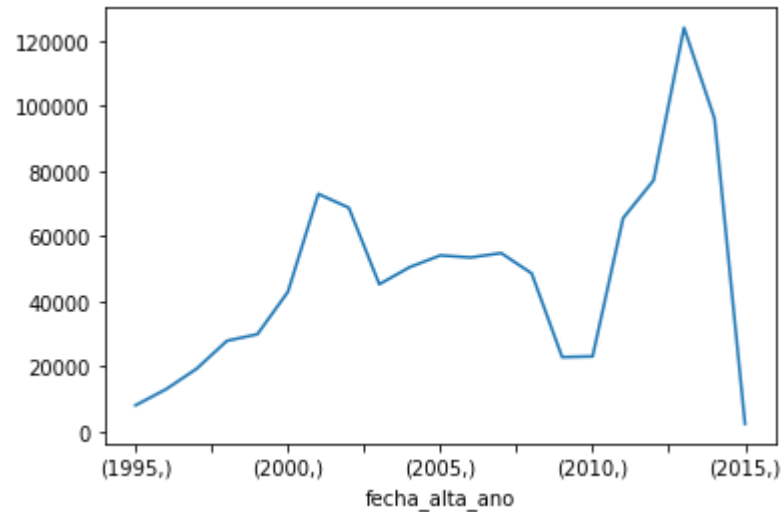
## EDA: Average household income by province

- The graph below shows that Madrid has the highest household income.



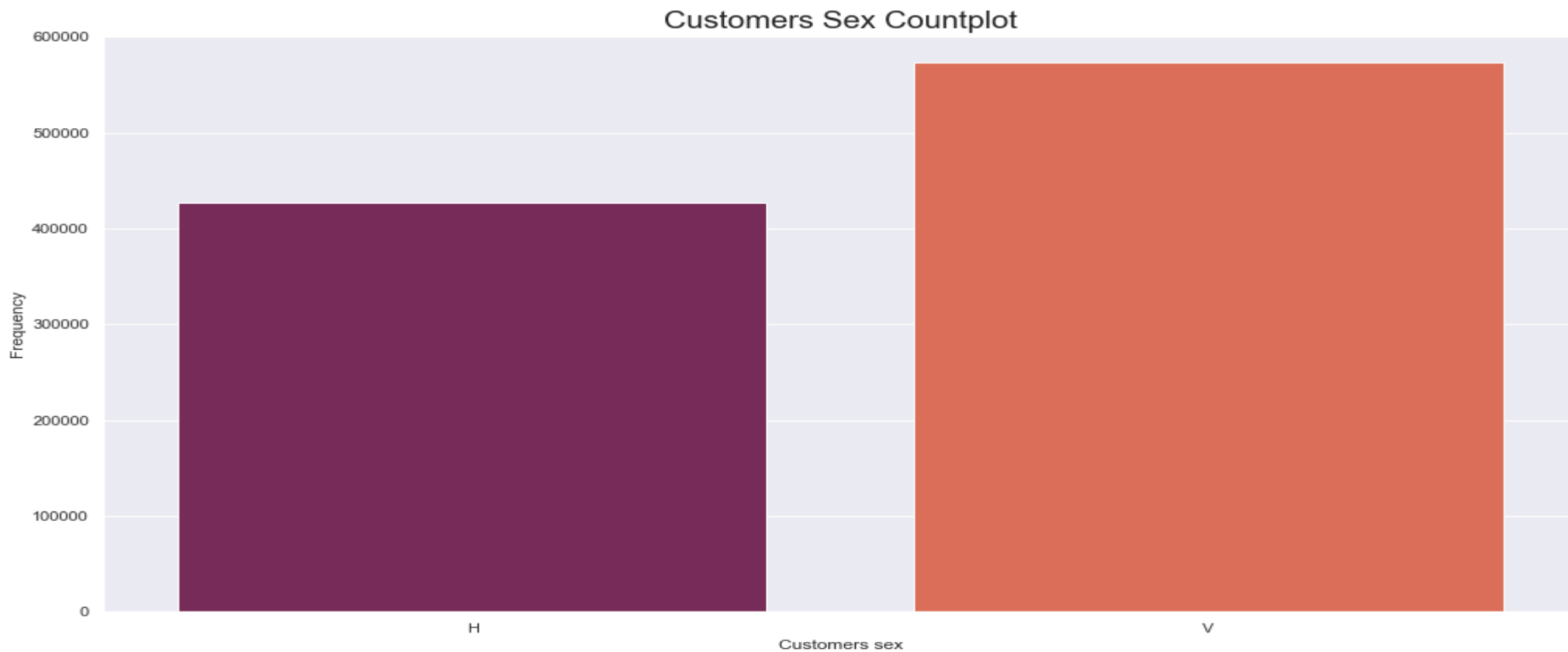
## EDA: Monthly and Annual Trends in Customer Registration

Number of customer registrations by year and by month. Note that the period after 2011 has more registrations compared to before.



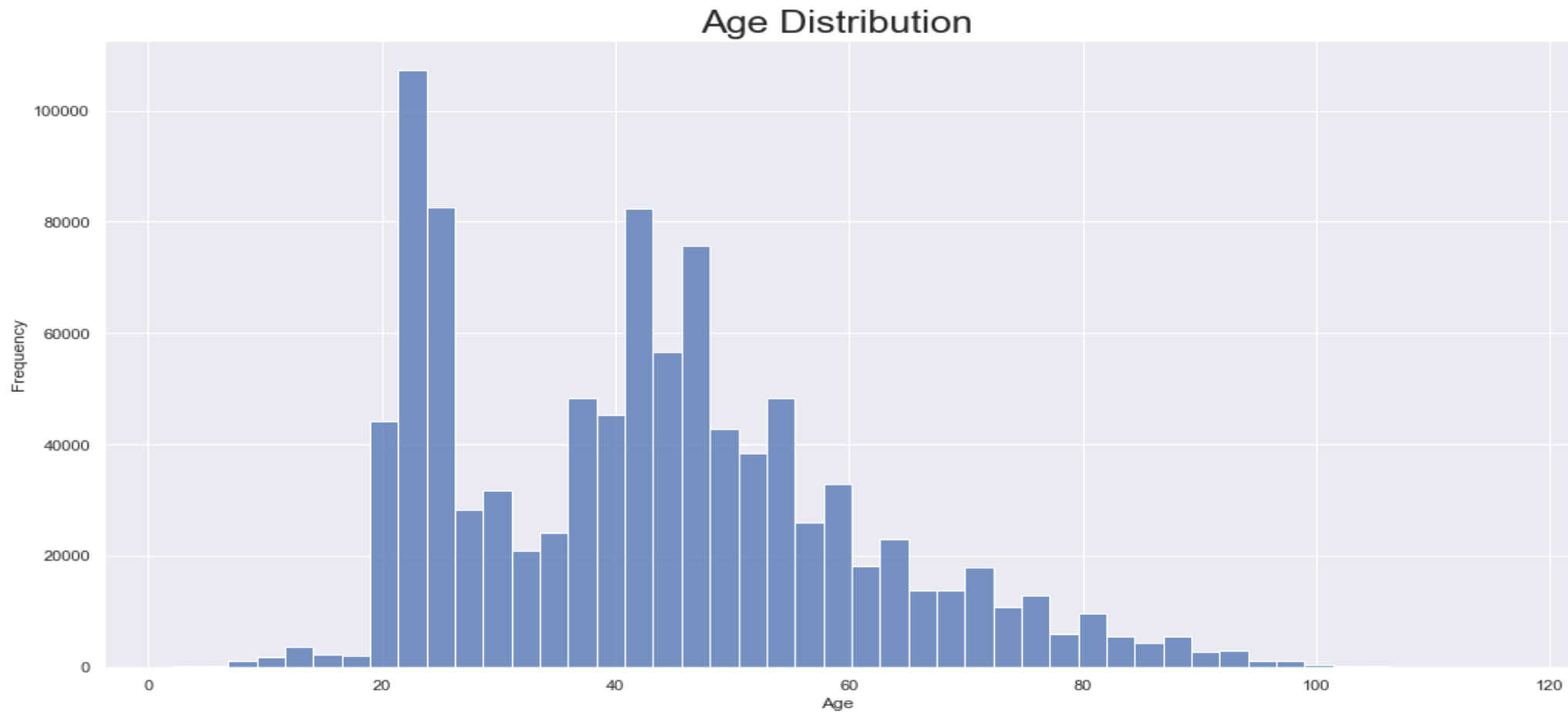
## EDA:Customers Sex Category Distribution

Most of the customers belong to sex category V



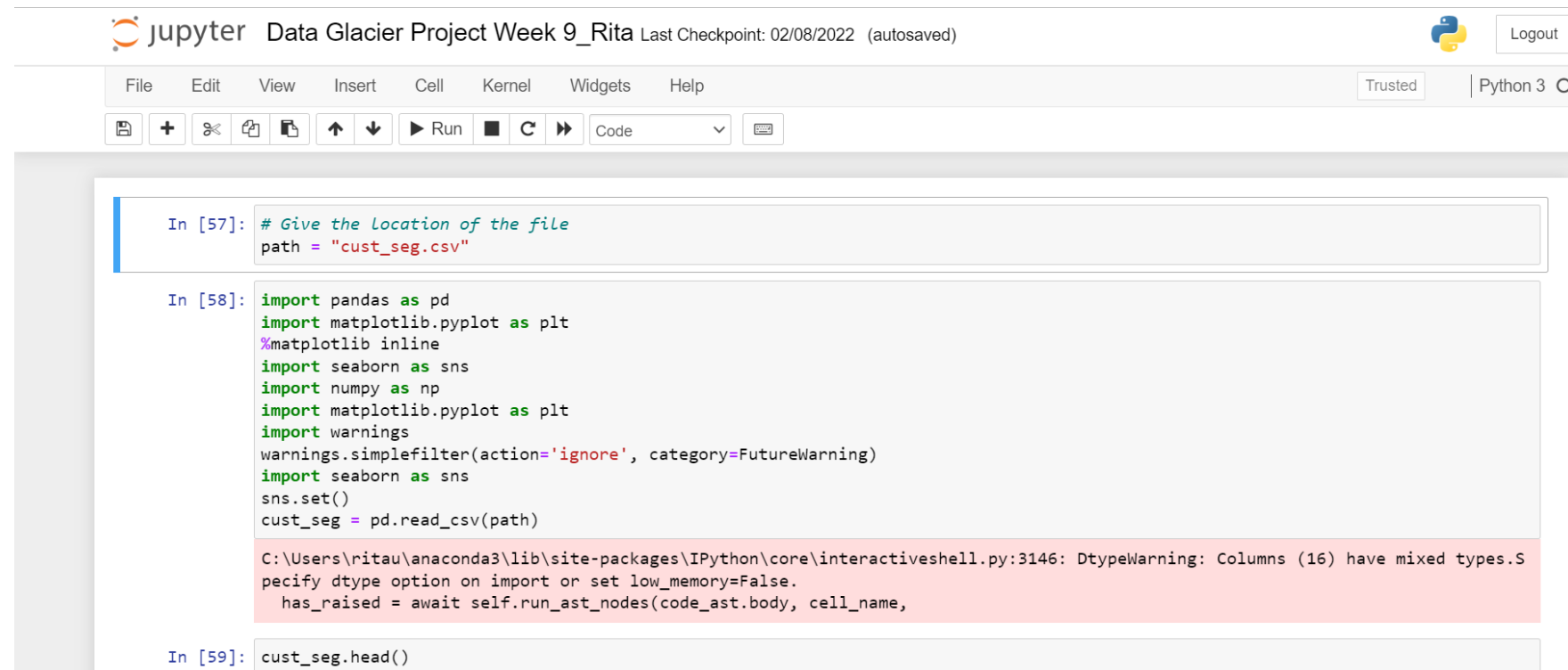
## EDA: Customers Age Distribution

Most of the customers are between the age of 22-45



# Tools and Techniques (Jupyter Notebook)

- ☐ An integrated development environment (IDE) for Python
- ☐ It comes with a console editor for executing the codes
- ☐ Having a lot of open-source libraries to:
  - ☐ Read, write
  - ☐ Integrate
  - ☐ Validate, clean the data
  - ☐ Perform arithmetic calculations
  - ☐ Perform Machine Learning .
  - ☐ Perform DQL(Data Query Language)



The screenshot displays the Jupyter Notebook interface. The top bar shows the Jupyter logo, the title "Data Glacier Project Week 9\_Rita", and the last checkpoint information "Last Checkpoint: 02/08/2022 (autosaved)". A "Logout" button is visible on the right. Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A toolbar contains icons for saving, adding, deleting, and running code cells. The main area shows three code cells. The first cell (In [57]) sets a file path. The second cell (In [58]) imports various libraries (pandas, matplotlib, seaborn, numpy) and reads a CSV file. A warning message is displayed below the second cell: "C:\Users\ritau\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (16) have mixed types. Specify dtype option on import or set low\_memory=False." The third cell (In [59]) shows the first few rows of the loaded data.

```
In [57]: # Give the Location of the file
path = "cust_seg.csv"

In [58]: import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
import seaborn as sns
sns.set()
cust_seg = pd.read_csv(path)

C:\Users\ritau\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (16) have mixed types. Specify dtype option on import or set low_memory=False.
has_raised = await self.run_ast_nodes(code_ast.body, cell_name,

In [59]: cust_seg.head()
```

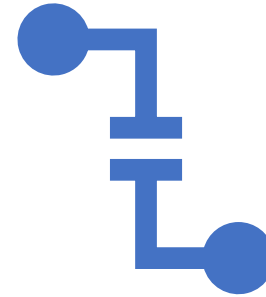


# Tools and techniques (Clustering: a machine learning model)

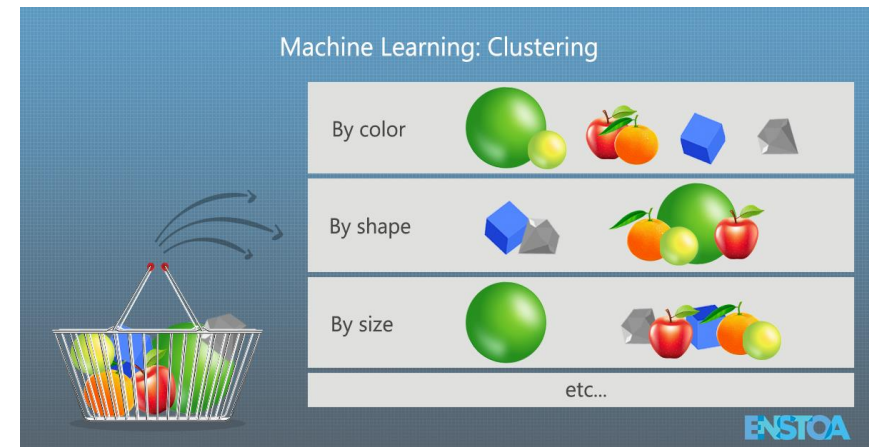


The data provided has some unnamed columns which made us conclude to use an unsupervised algorithm.

K-means clustering is one of the most widely used unsupervised machine learning models that form clusters of data based on the similarities between data instances.

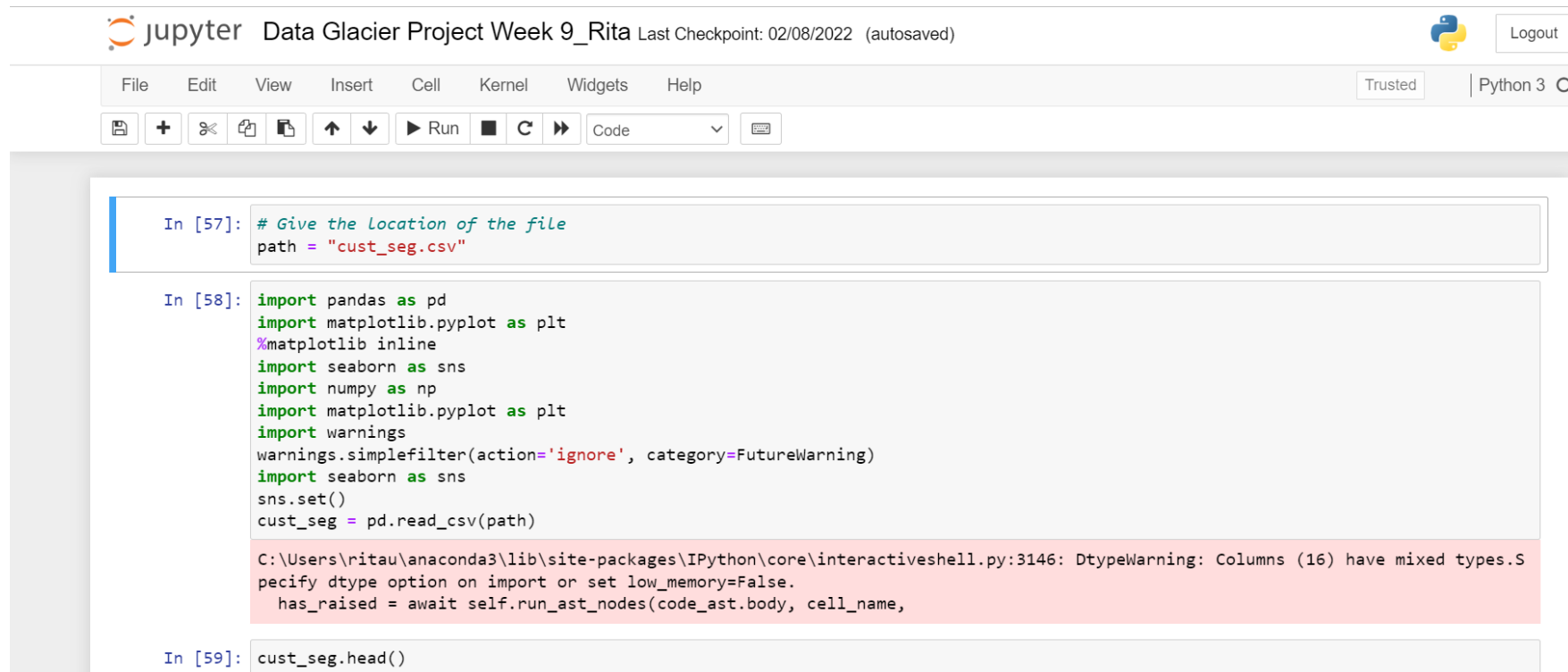


K-means clustering is an unsupervised clustering algorithm and that it belongs to the non-hierarchical class of clustering algorithms.



# Implementation: Extraction

The customer segmentation data was extracted from the source system into the virtual machine local system and read into Jupyter Notebook.



The screenshot displays a Jupyter Notebook interface. The top bar shows the Jupyter logo, the notebook name "Data Glacier Project Week 9\_Rita", and the last checkpoint information "Last Checkpoint: 02/08/2022 (autosaved)". On the right, there is a "Logout" button and a Python 3 kernel indicator. Below the top bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A toolbar with various icons for file operations and execution is located below the menu bar. The main area contains three code cells. The first cell, labeled "In [57]:", contains a comment and a variable assignment: `# Give the Location of the file` and `path = "cust_seg.csv"`. The second cell, labeled "In [58]:", contains a series of import statements for pandas, matplotlib, seaborn, and numpy, followed by a warning filter and a call to `pd.read_csv(path)` to load the data into a DataFrame named `cust_seg`. Below this code, a warning message is displayed: "C:\Users\ritau\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (16) have mixed types. Specify dtype option on import or set low\_memory=False." The third cell, labeled "In [59]:", contains the command `cust_seg.head()` to view the first few rows of the data.

```
In [57]: # Give the Location of the file
path = "cust_seg.csv"

In [58]: import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
import seaborn as sns
sns.set()
cust_seg = pd.read_csv(path)

C:\Users\ritau\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (16) have mixed types. Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,

In [59]: cust_seg.head()
```

# Implementation: Transformation

The Python programming language was used for data transformation, cleaning and data quality checks.

- Steps in our data transformation stages

- ☐ Data Aggregation
- ☐ Data Quality
- ☐ Data Deduplication
- ☐ Data Cleansing
- ☐ Data Filtering

# Implementation: Transformation

- We found that the issue with the original data set were:
- Data Incompleteness - various columns, such as the ind\_empleado (Employee index), pais\_residencia (Customer's Country residence), sexo (Customer's sex) etc., have empty values (NA) in certain records.
- Mitigation: Impute the missing values based on distribution in the training dataset. Depending on the context of the attribute, fill in the missing values with an appropriate value. If the variable is categorical, use the mode. If the variable is quantitative, use the median if discrete, and the mean if continuous.
- Recommendation: Make input for such fields required, especially easily obtainable ones such as customer sex. If information cannot be obtained, then do not include. Alternatively, may consider removing some attributes which are not so important.

# Implementation: Transformation

- Inconsistent data types
- Mitigation: Make data transformations to ensure consistent data types in each column. Examples: Convert selected records in strings to numeric, and remove non-numeric characters from the string.
- Recommendation: Ensure that fact tables in the given database have constraints on data types. Having different data types for a given field makes it difficult to interpret results at the later stage.
- Inconsistent values for the same attribute (e.g. -999999 as a value for seniority in months).
- Mitigation: The seniority in months cant have a negative number. Use regular positive numbers to replace extended values into positive numbers or zero to ensure consistency across seniority in months. Or replace -999999 with NA and perform the missing data transformations as above using the median
- Recommendation: Enforce a mandatory field where only positive integers are valid for this attribute.

# Implementation: Data Quality

## Replacing With Mean

This is the most common method of imputing missing values of numeric columns. If there are outliers then the mean will not be appropriate. In such cases, outliers need to be treated first.

renta and indrel\_1mes features are numeric columns with outliers, hence replacing with mean will not be appropriate.

## Replacing With Mode

Mode is the most frequently occurring value. It is used in the case of categorical features.

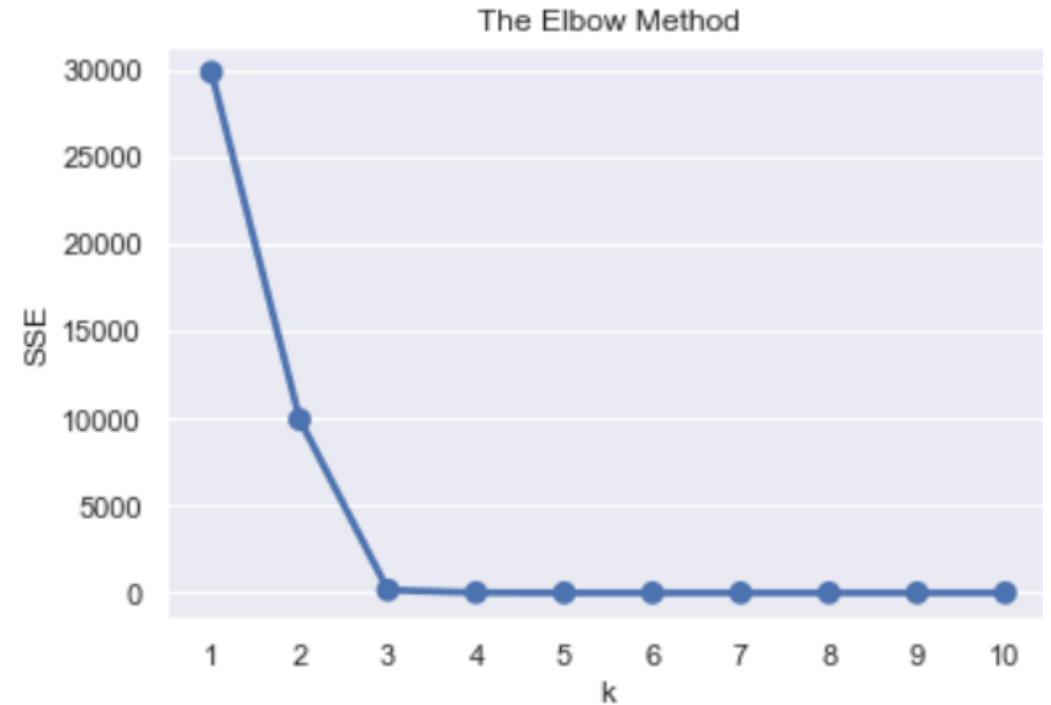
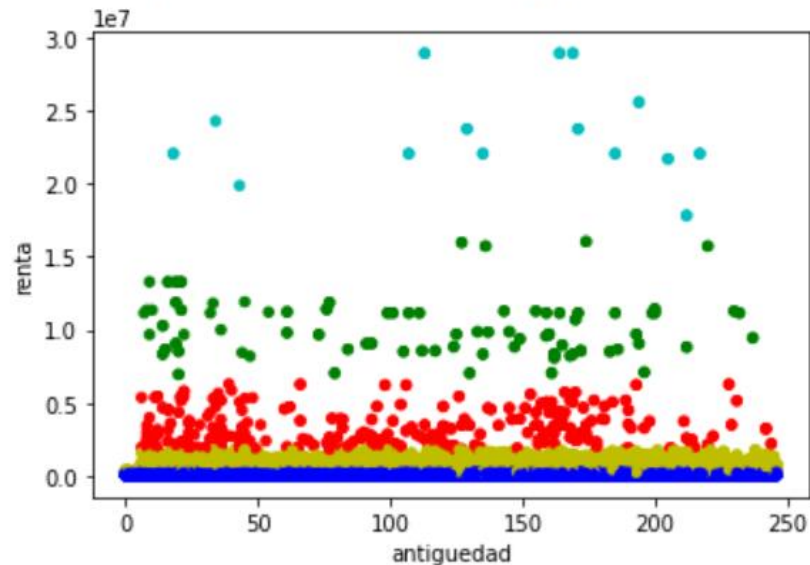
You can use the 'fillna' method for imputing the categorical columns 'sexo', 'ind\_empleado', 'pais\_residencia', 'tiprel\_1mes', 'indresi', 'indext', 'canal\_entrada', 'indfall', 'nomprov', 'conyuemp'.

```
cust_seg['sexo'] = cust_seg['sexo'].fillna(cust_seg['sexo'].mode()[0])
cust_seg['ind_empleado'] = cust_seg['ind_empleado'].fillna(cust_seg['ind_empleado'].mode()[0])
cust_seg['pais_residencia'] = cust_seg['pais_residencia'].fillna(cust_seg['pais_residencia'].mode()[0])
cust_seg['tiprel_1mes'] = cust_seg['tiprel_1mes'].fillna(cust_seg['tiprel_1mes'].mode()[0])
cust_seg['indresi'] = cust_seg['indresi'].fillna(cust_seg['indresi'].mode()[0])
cust_seg['indext'] = cust_seg['indext'].fillna(cust_seg['indext'].mode()[0])
cust_seg['canal_entrada'] = cust_seg['canal_entrada'].fillna(cust_seg['canal_entrada'].mode()[0])
cust_seg['indfall'] = cust_seg['indfall'].fillna(cust_seg['indfall'].mode()[0])
cust_seg['nomprov'] = cust_seg['nomprov'].fillna(cust_seg['nomprov'].mode()[0])
cust_seg['conyuemp'] = cust_seg['conyuemp'].fillna(cust_seg['conyuemp'].mode()[0])
```

# Implementation: K-Means Clustering

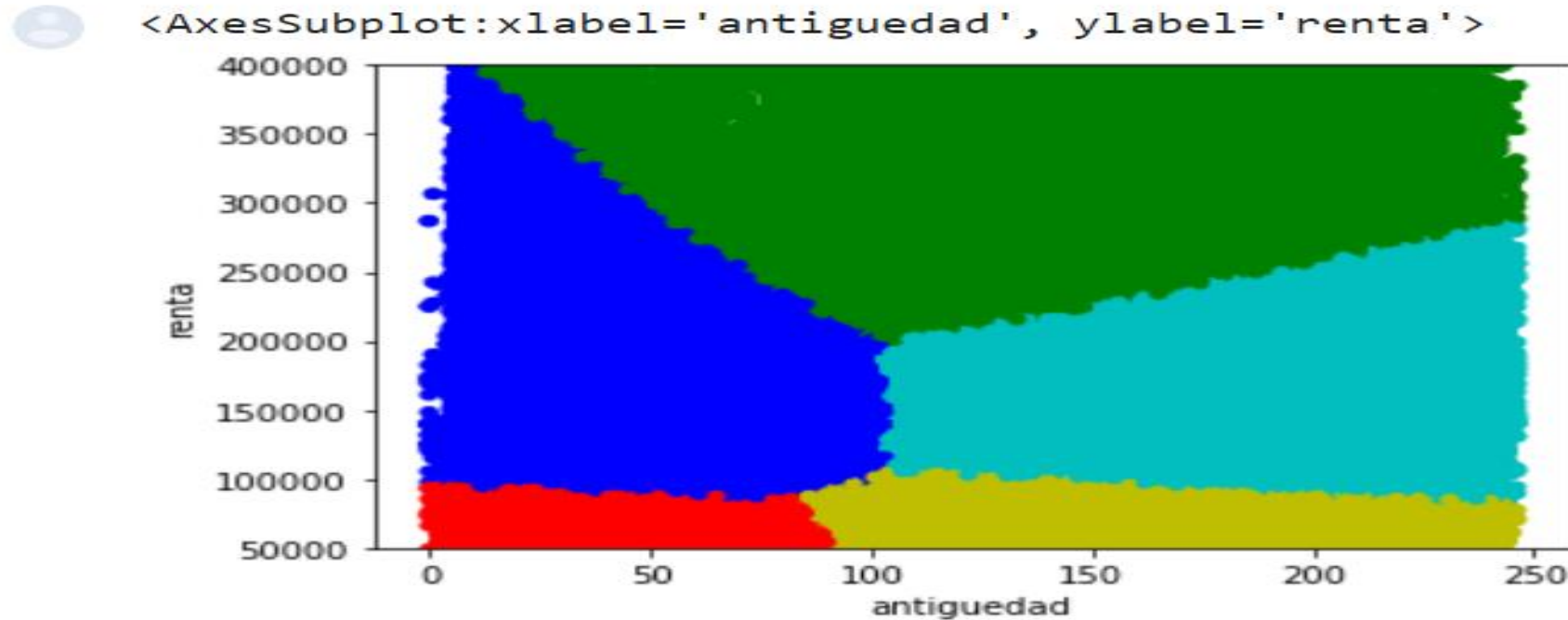
## STEPS:

- Gather the data(transformed data)
- Manage skewness and scale each variable,
- Explore the data,
- Cluster the data,
- Interpret the result



A clear bend can be seen at the 2nd cluster. Cool!

# Results: Finding similarities between income and Seniority.

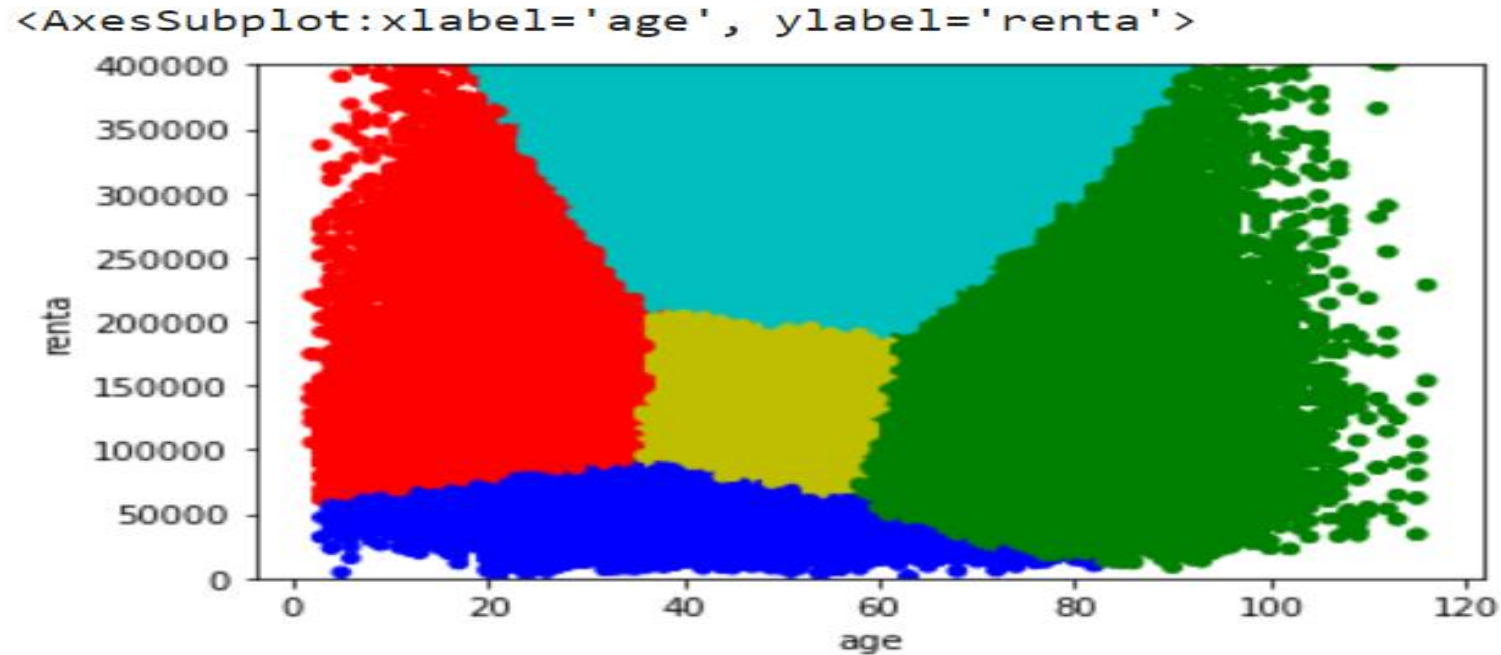


Based on this clustering results, the five groups which we could split the customer segment are:

- Customers with lower income (under 100,000) and less seniority (under 96 months)\*Customers with lower income (under 100,000) and more seniority (over 96 months)
- Customers with medium income (between 100,000 and 250,000) and less seniority (under 96 months)
- Customers with medium income (between 100,000 and 250,000) and more seniority (over 96 months)
- Customers with higher income (over \$250,000)



# Results: Finding similarities between income and Age.



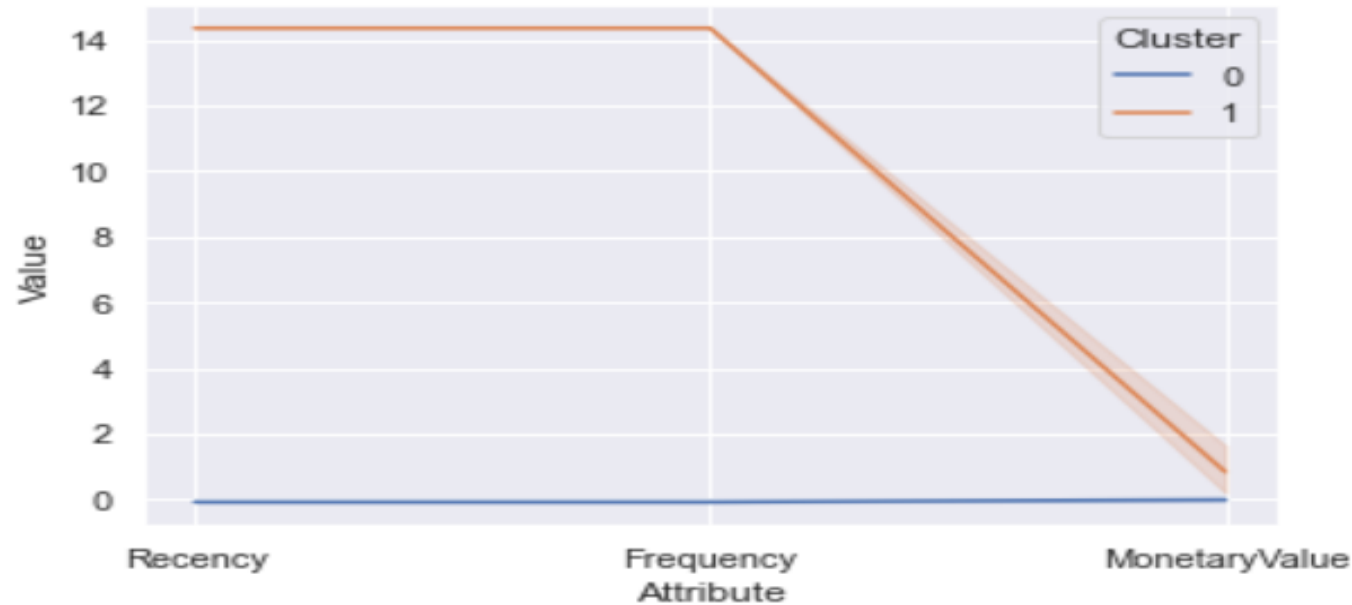
Based on this clustering, we could also segment the customers by the following groups:

- High-income (above 250,000) customers \* Old age (above 60) customers with household income not exceeding 250,000
- Low income (under \$70,000) customers with age not exceeding 60
- Other middle aged (35-60) customers
- Other young (under 35) customers

# Results: Finding similarities in Active customers.

Out[117]:

<AxesSubplot:xlabel='Attribute', ylabel='Value'>



By using this plot, we know how each segment differs. It describes more than we use the summarized table.

We infer that cluster 0 is they are new customers, less active, made no deposits. Therefore, it could be the cluster of a new customers.

Finally, the cluster 1 is old customers, more active, but less deposits. Therefore, it could be the cluster of Repeat customers.

# Conclusions

We created customer segmentations based on:

1. Their income and seniority
2. Their income and age
3. How active they are as a customer(Monetary wise as well)

The above segmentation can help shape on personalized gifts to increase customer relationship, cross selling, customer referrals etc.

# Thank You



**Data Glacier**

Your Deep Learning Partner