



**AI VIET NAM**

@aivietnam.edu.vn

# Data Visualization & EDA

**Hoang-Bach Ngo - TA**  
**Hien Ho - STA**

# Objectives

- ✓ Why we need data visualization and EDA?
- ✓ Using LLMs for exploring data
- ✓ Modern tools for data visualization



PyGWalker

# Outline

1

**Why?**

2

**EDA w/ LLMs**

3

**Visualization  
Tools**

4

**Hands on with  
code**

# Outline

1

**Why?**

2

**EDA w/ LLMs**

3

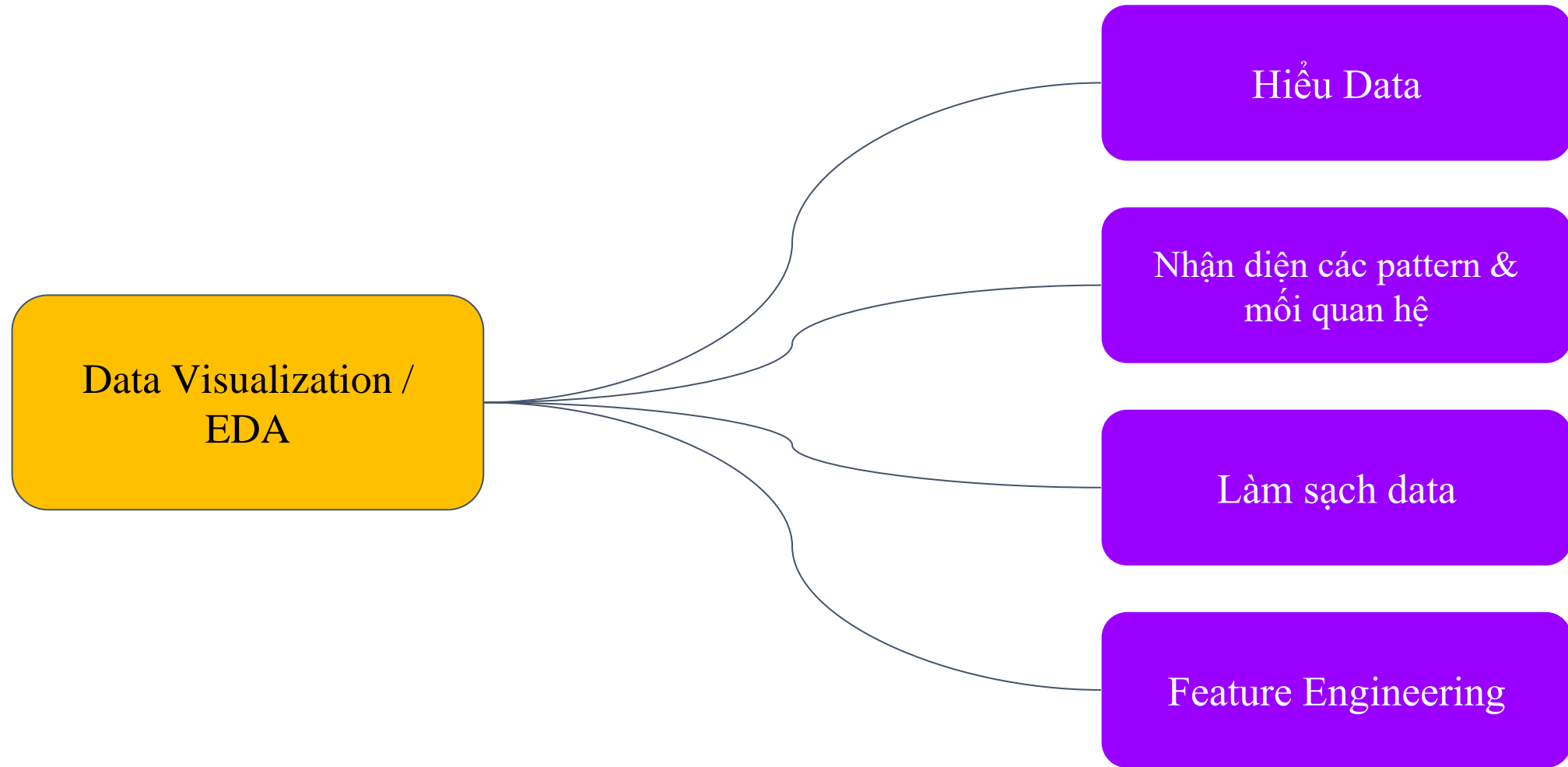
**Visualization Tools**

4

**Hands on with code**



## ❖ Tại sao cần data visualization & EDA?



## ❖ Tại sao cần data visualization & EDA?

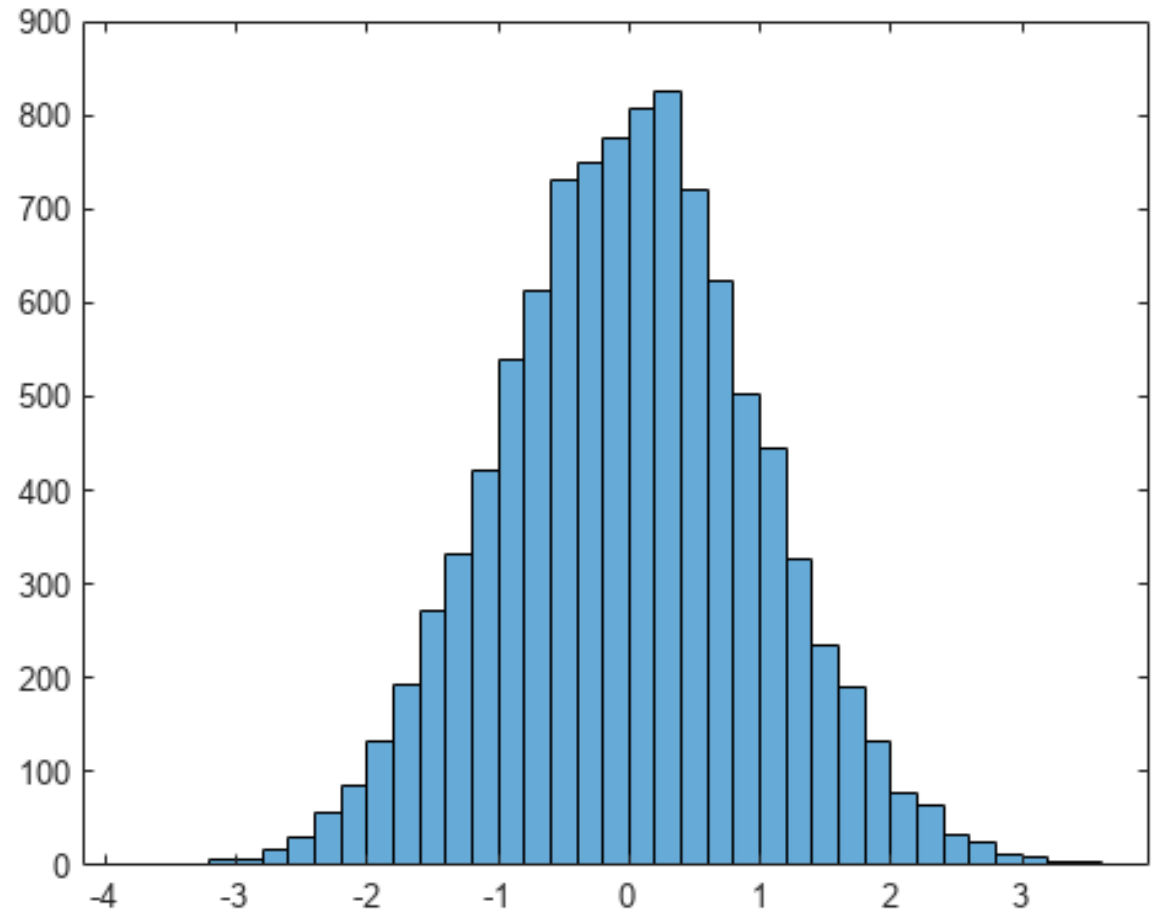
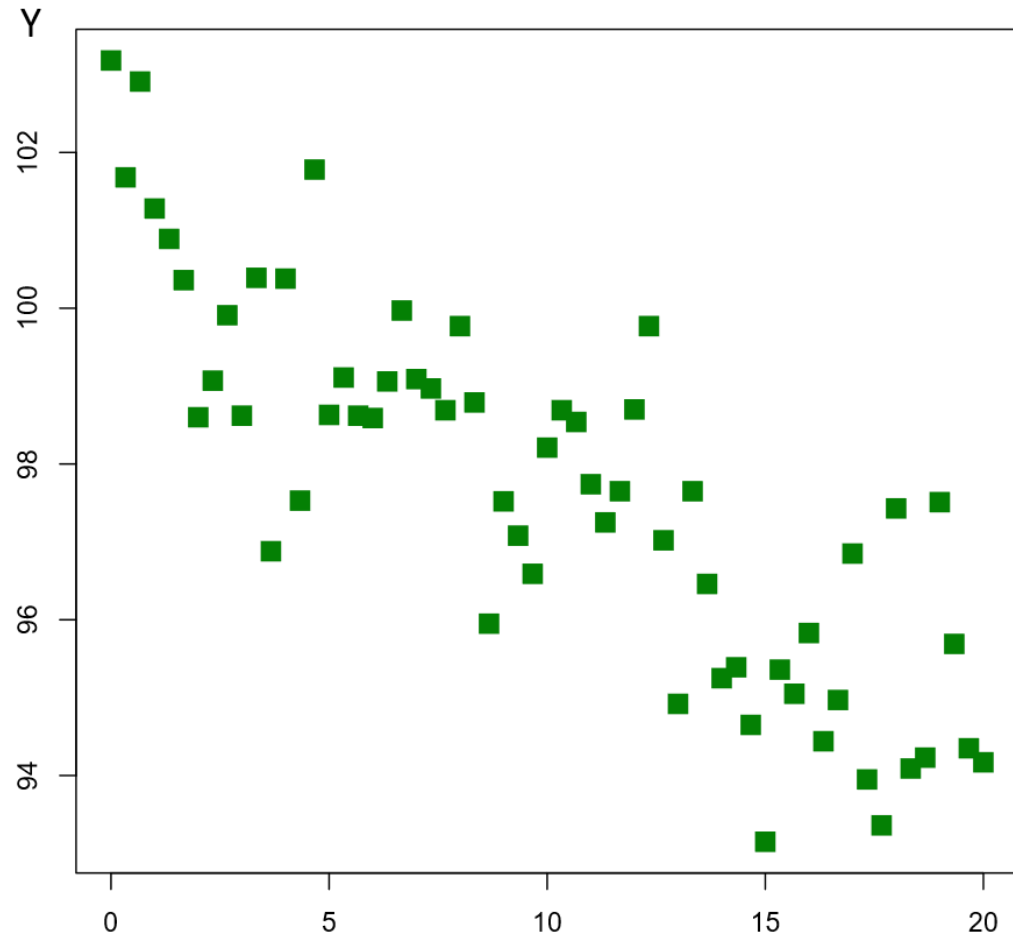
Con người xử lý hình ảnh nhanh hơn văn bản 60.000 lần (Vogel et al., 1986).

90% thông tin truyền đến não là hình ảnh (Potter et al., 2014).

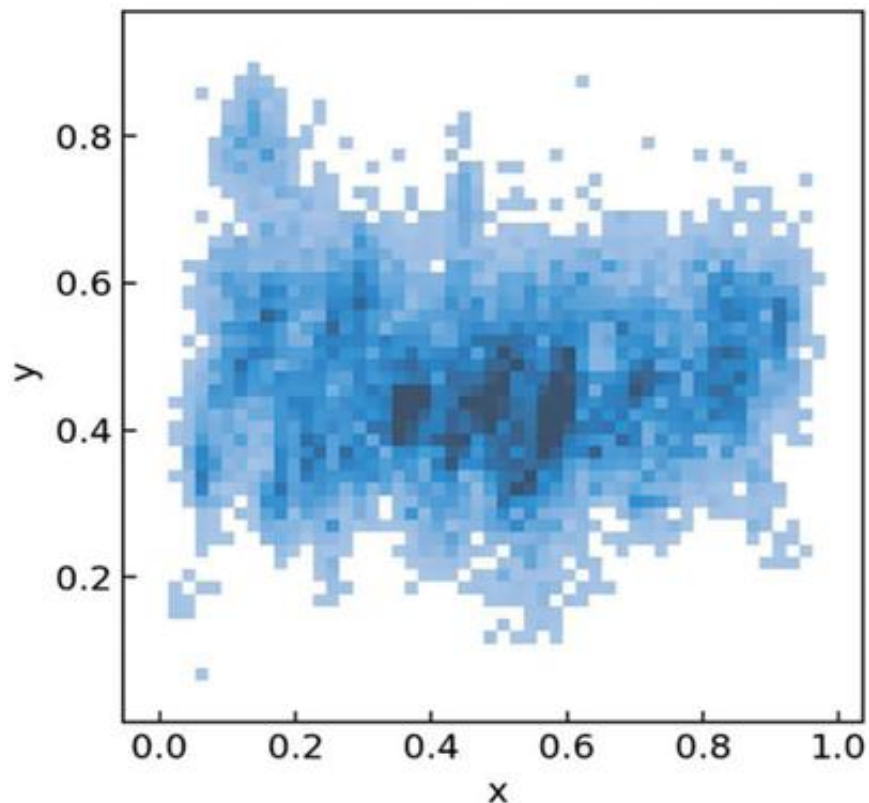
35% não bộ của chúng ta dành cho thị giác (Kelts, 2010).

Não người có thể xử lý toàn bộ hình ảnh mà mắt nhìn thấy chỉ trong 13 mili giây.

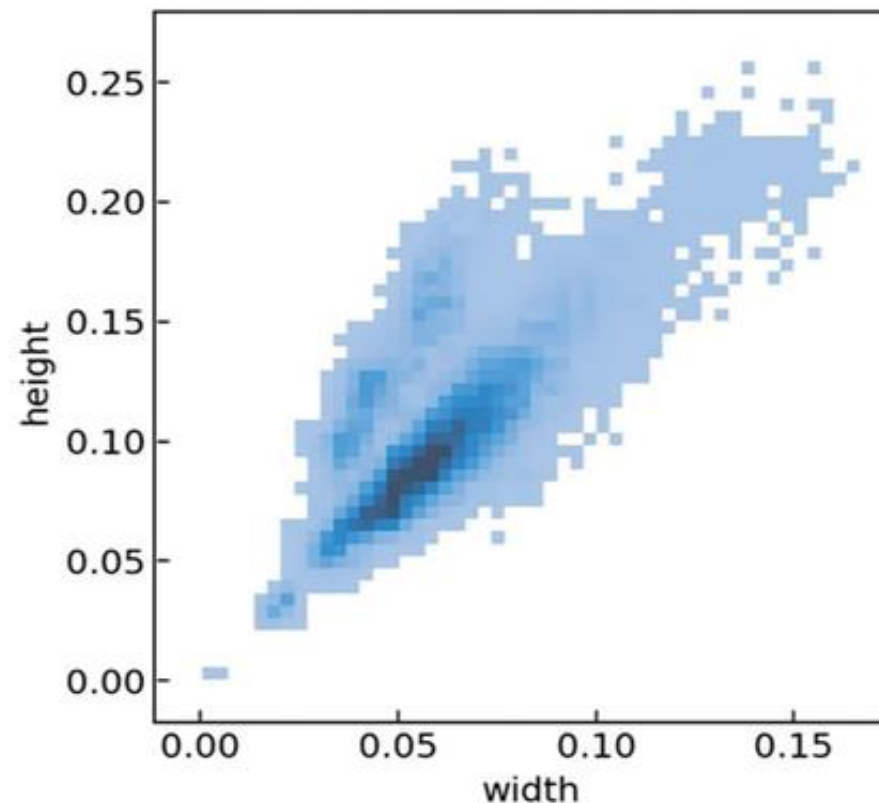
❖ Thấy được xu hướng chung / phân phối của dataset



❖ Thấy được xu hướng chung / phân phối của dataset



(a) Center coordinate distribution

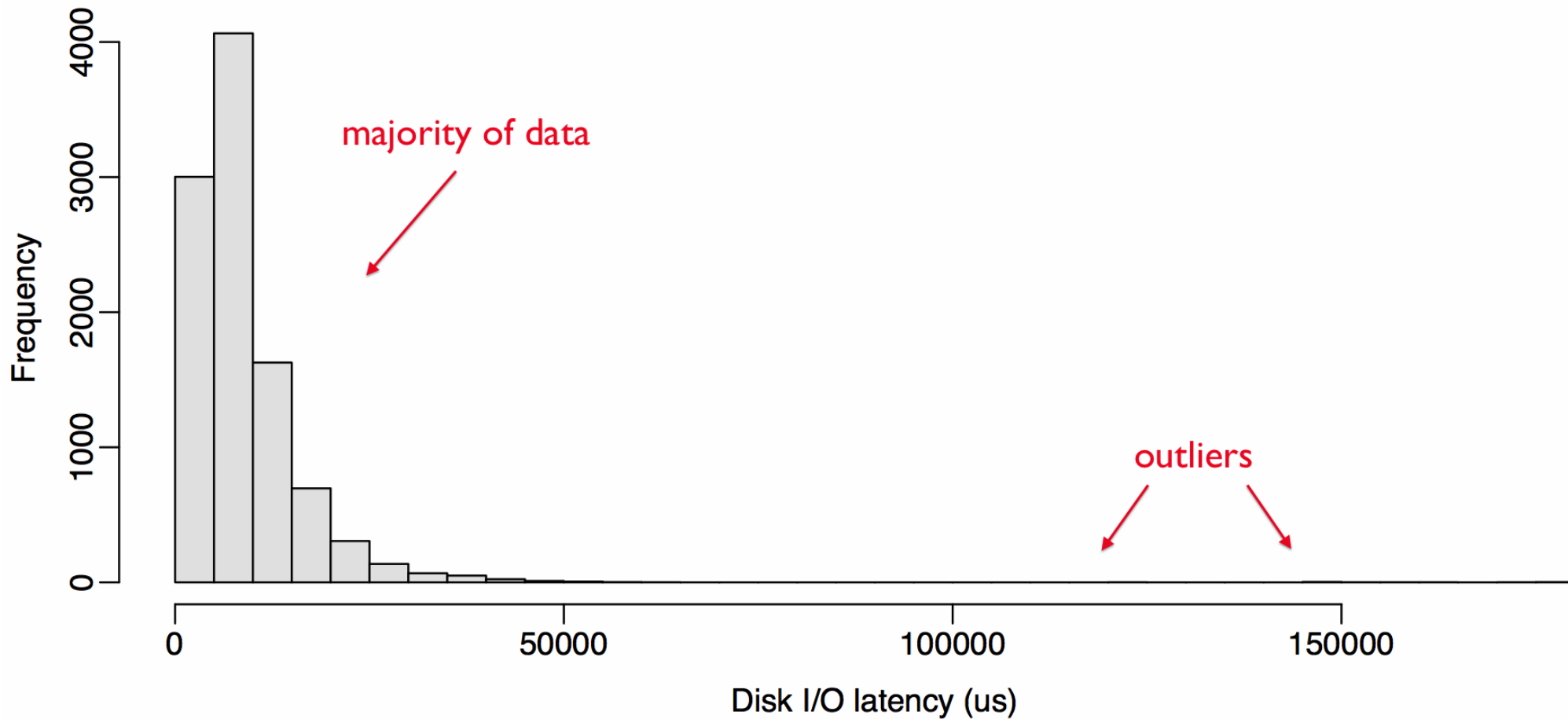


(b) Bounding box size distribution

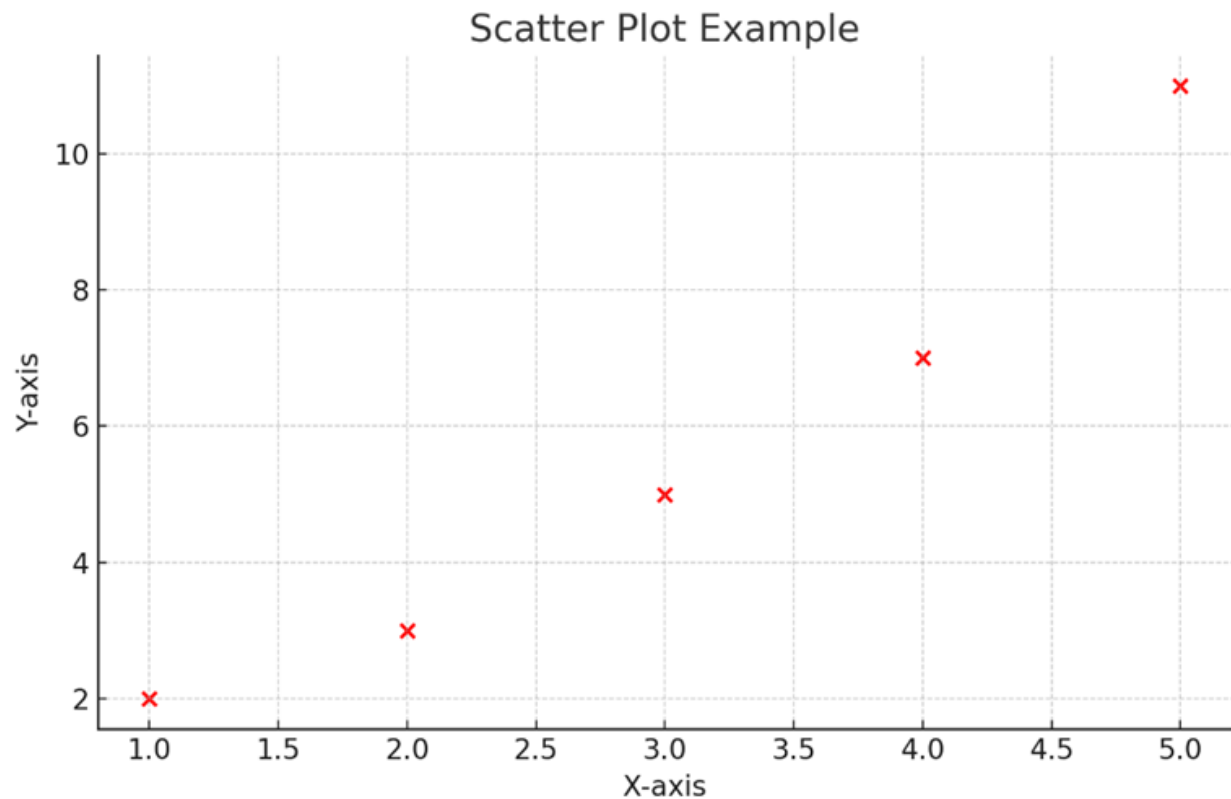
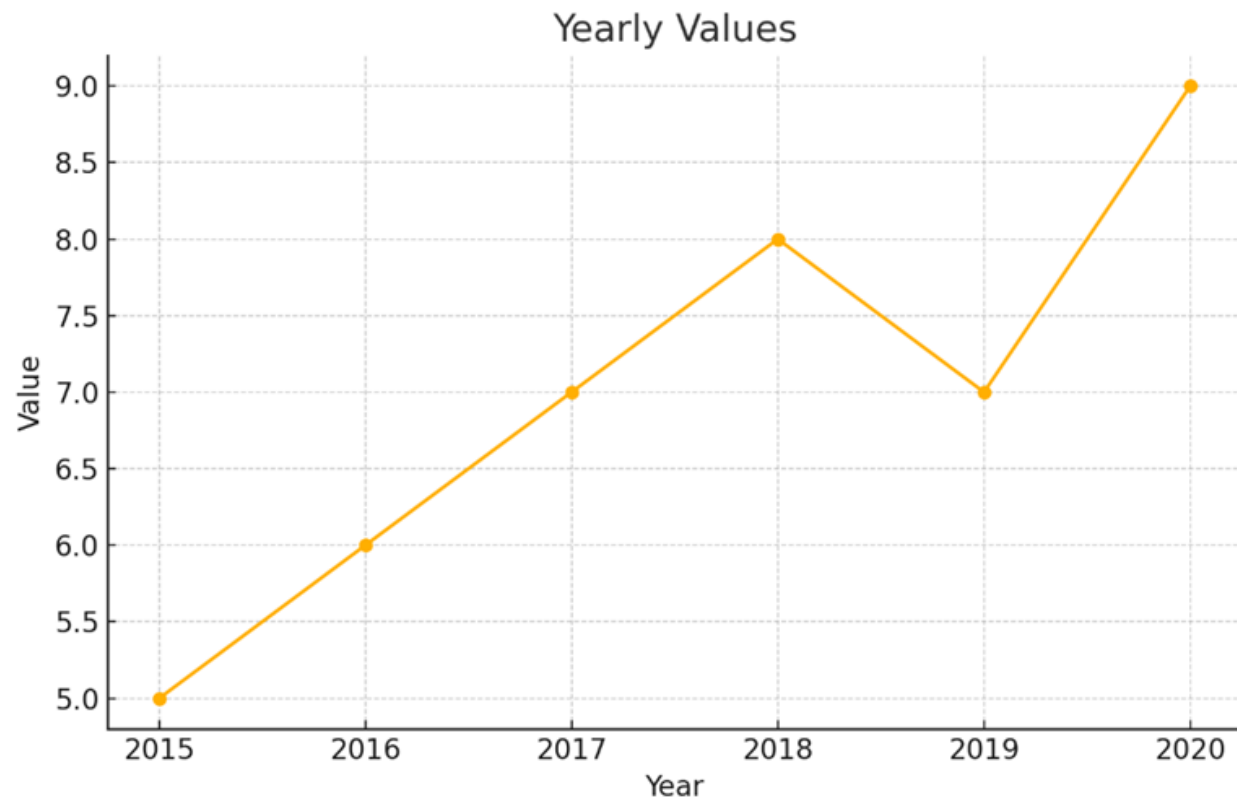


## ❖ Phát hiện outlier trong data

**Latency Distribution**

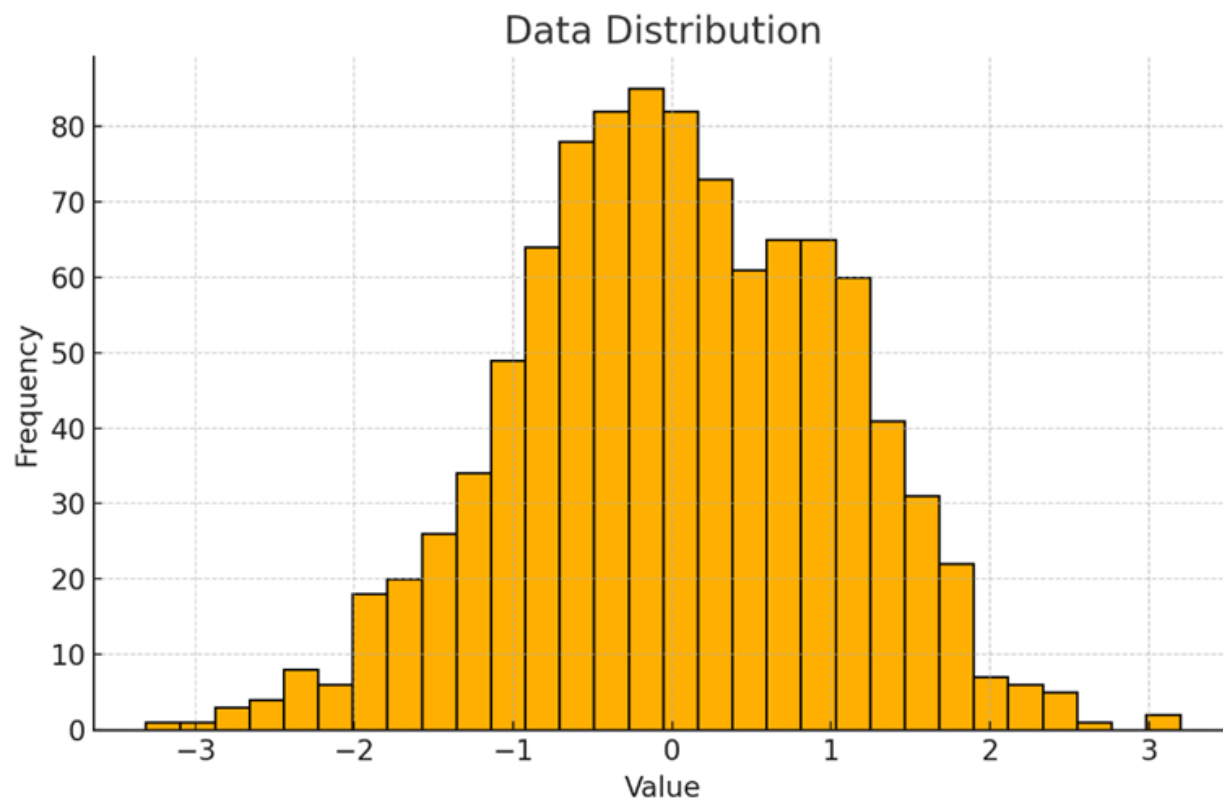


## ❖ Các loại biểu đồ phổ biến



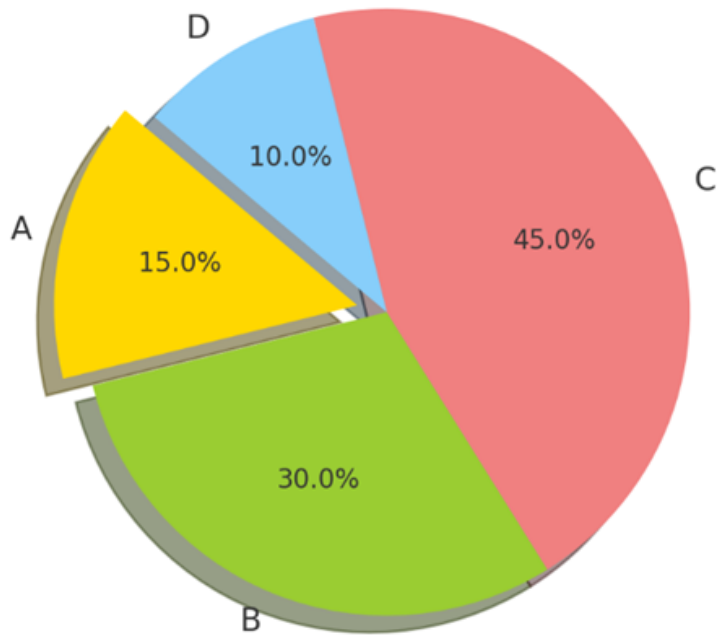


## ❖ Các loại biểu đồ phổ biến

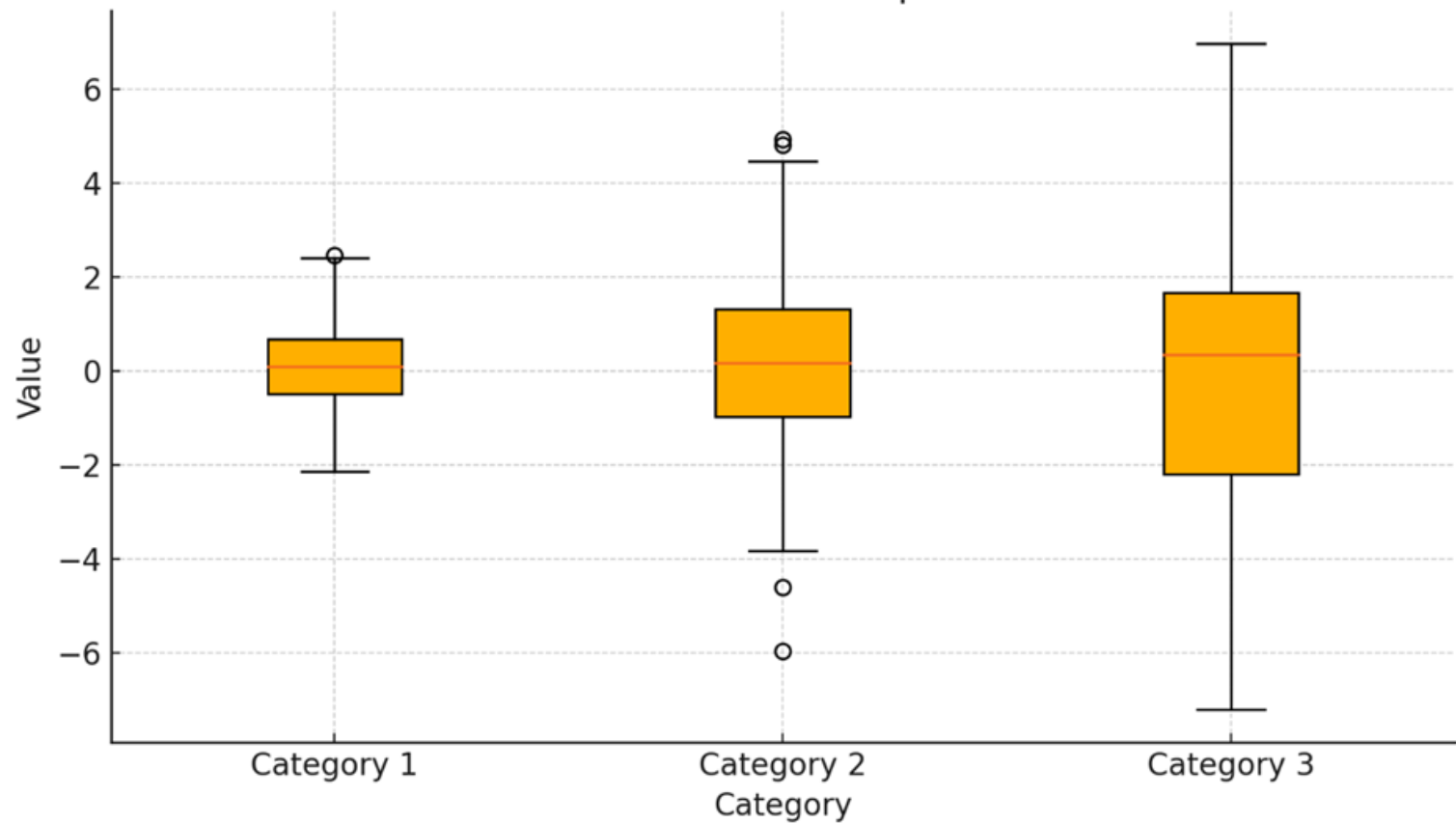


## ❖ Các loại biểu đồ phổ biến

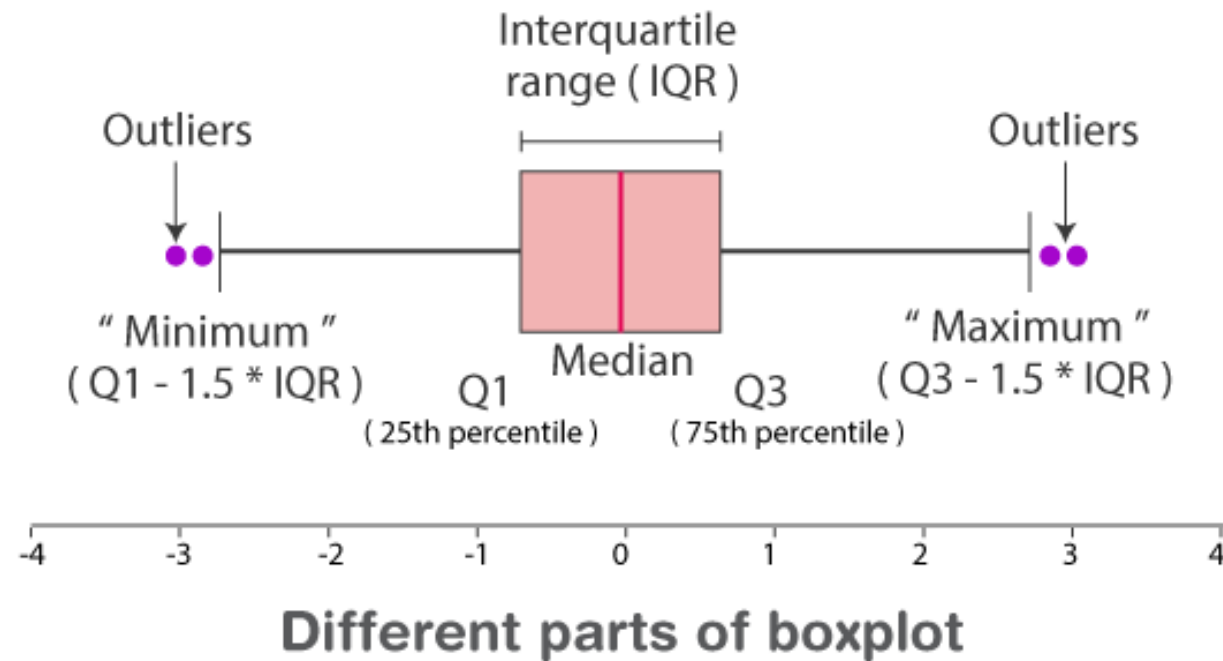
Pie Chart Example



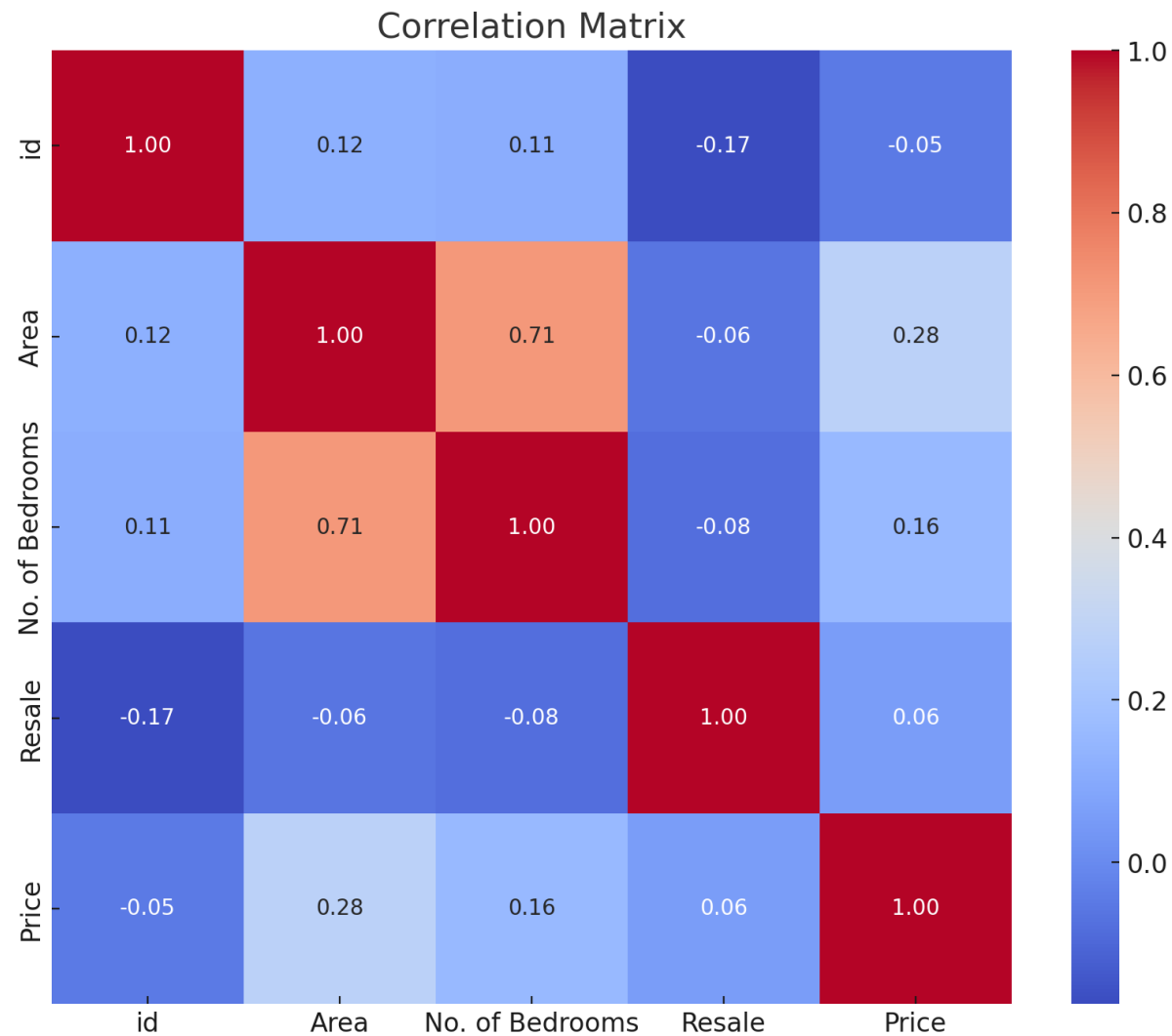
Box Plot Example



## ❖ Các loại biểu đồ phổ biến



## ❖ Các loại biểu đồ phổ biến



# Outline

1

**Why?**

2

**EDA w/ LLMs**

3

**Others Tools**

4

**Hands on with code**



## ❖ EDA với ChatGPT/Gemini



BACH NGO · COMMUNITY PREDICTION COMPETITION · PRIVATE · 10 DAYS TO GO

## [AIO-Competition] Part1 - Housing Price Prediction

Cuộc thi đầu tiên trong series Competition AIO2024

### Submit Prediction



## Overview

## Data

Code

## Models

## Discussion

## Leaderboard

## Rules

## Team

## Submissions

## Settings

## Overview

## Start

7 days ago



**Close**

10 days to go

### Description

## Cuộc thi Dự đoán Giá Nhà - AIO

Chào mừng các bạn đến với cuộc thi "Dự đoán Giá Nhà" được tổ chức nội bộ bởi AIO! Cuộc thi này nhằm mục đích tạo cơ hội cho các học viên trong khóa, thể hiện kỹ năng phân tích dữ liệu và dự đoán thị trường bất động sản. Đây là cơ hội tuyệt vời để các bạn thể hiện sự sáng tạo, kiến thức và khả năng phân tích của mình trong lĩnh vực đầy thử thách này.

### Competition Host

Bach Ngo



## Prizes & Awards

## Kudos

Does not award Points or Medals

## Participation

75 Entrants

## 22 Participants

## 17 Teams

89 Submissions

## Tags

## Housing

Beginner

Tabular

## Regression





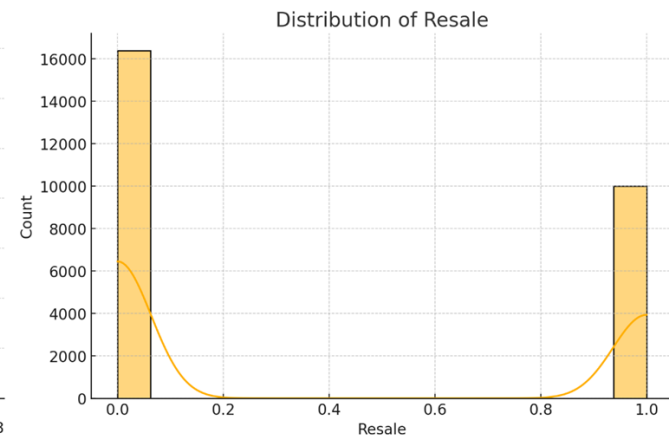
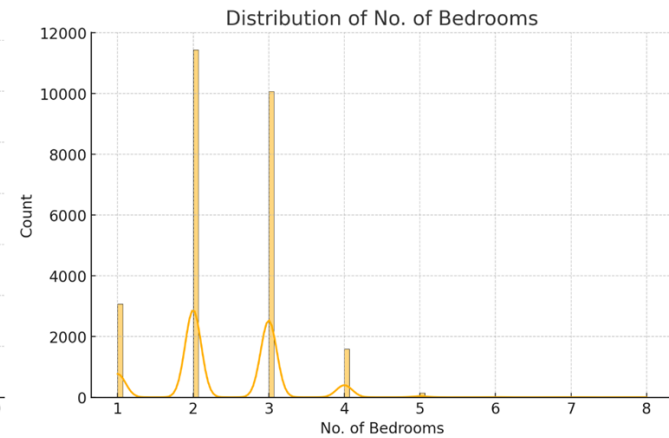
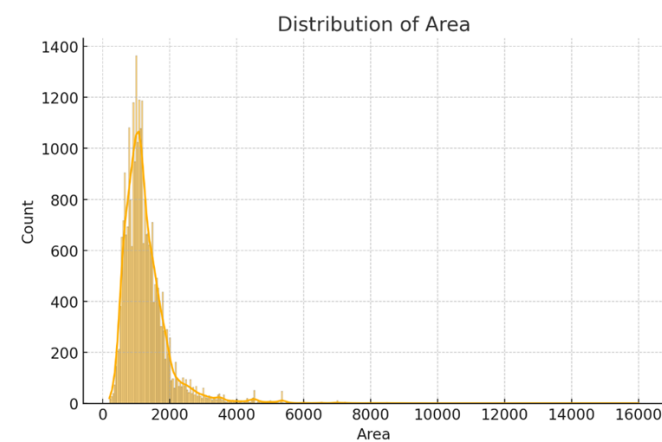
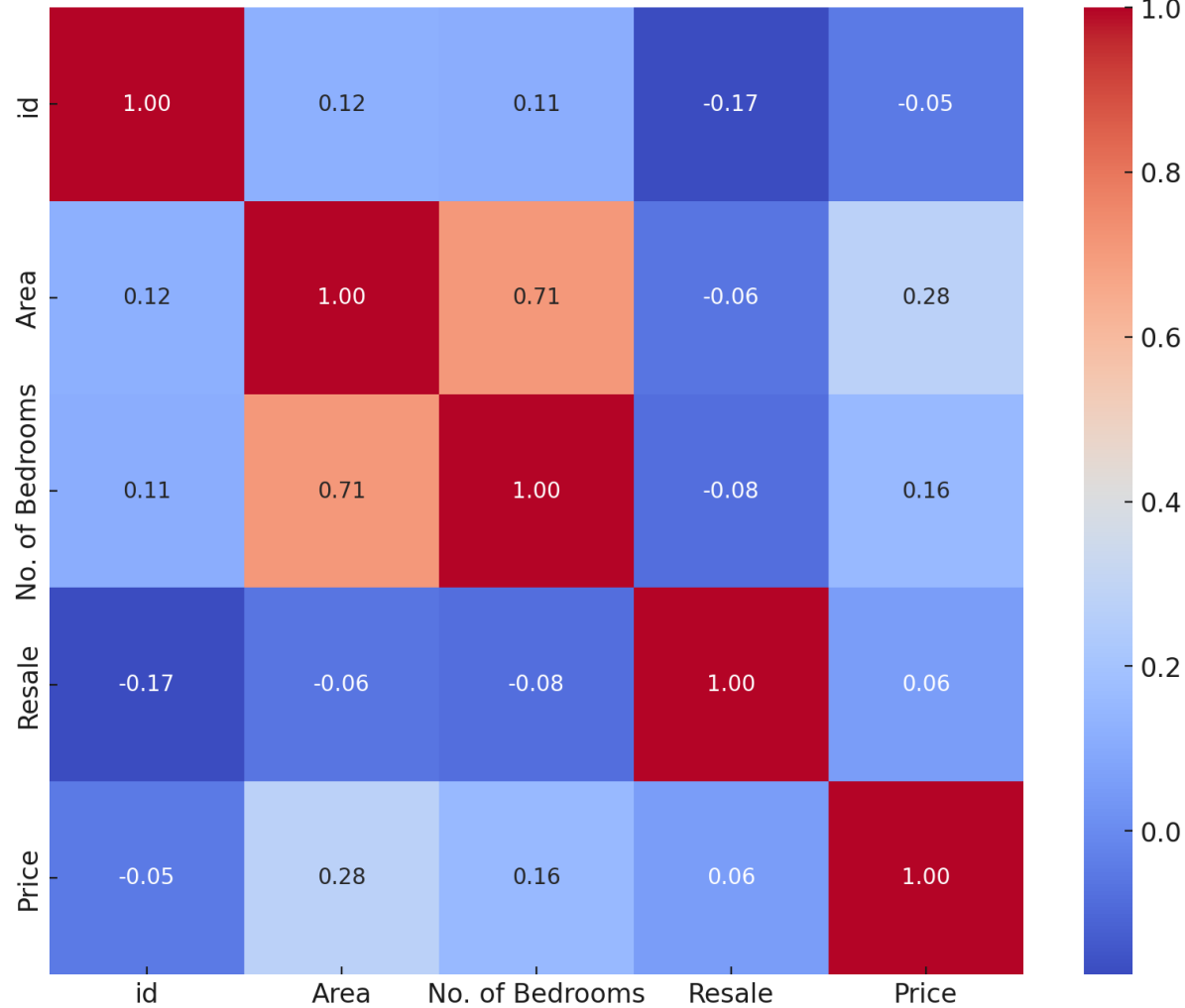
## ❖ EDA với ChatGPT/Gemini

Bạn là một chuyên gia phân tích dữ liệu được trang bị kiến thức chuyên sâu về thống kê và khả năng giải thích dữ liệu rõ ràng. Bạn có khả năng phân tích các tập dữ liệu đa dạng, từ bảng tính đơn giản đến dữ liệu phức tạp hơn. Nhiệm vụ của bạn là:

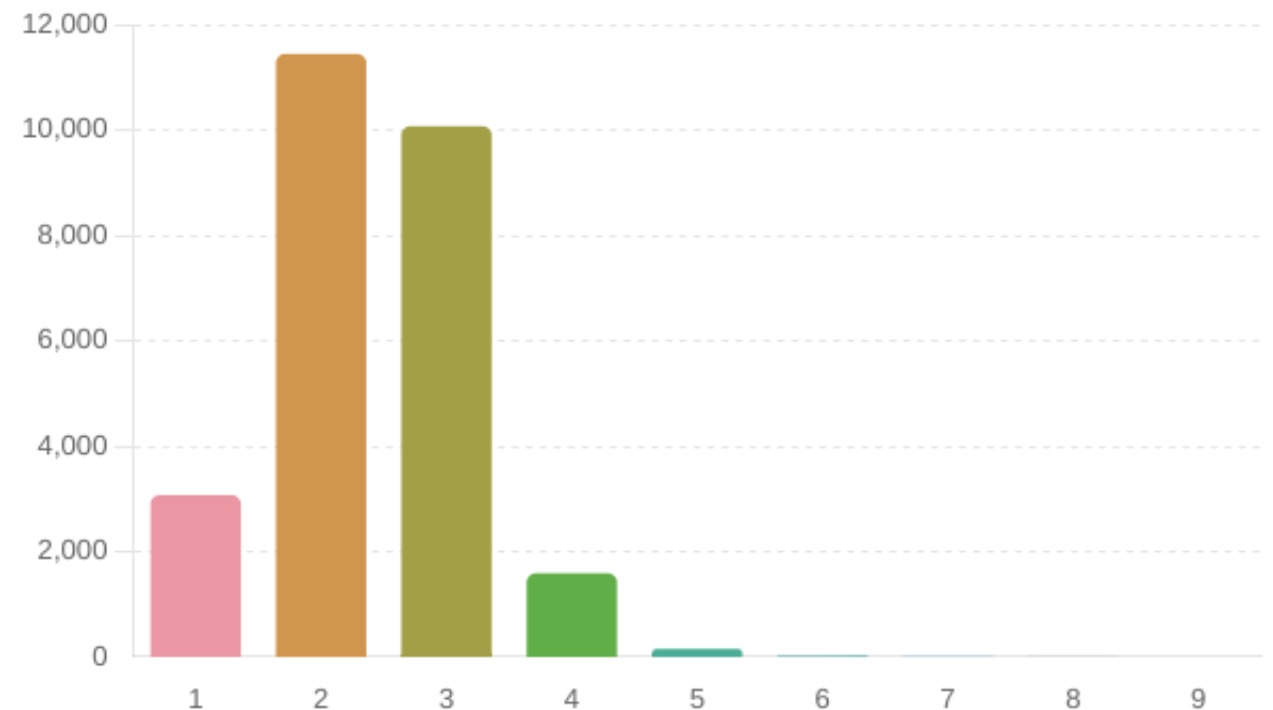
- **Tiếp nhận dữ liệu:** Nhận dữ liệu từ người dùng ở nhiều định dạng CSV.
- **Tiền xử lý:** Làm sạch và chuẩn bị dữ liệu để phân tích, bao gồm xử lý giá trị thiếu, loại bỏ ngoại lệ và chuyển đổi dữ liệu nếu cần.
- **Phân tích thăm dò:** Thực hiện phân tích thăm dò để hiểu rõ cấu trúc, phân bố và các đặc tính của dữ liệu.
- **Phân tích chuyên sâu:** Áp dụng các kỹ thuật phân tích phù hợp (thống kê mô tả, kiểm định giả thuyết, hồi quy, phân cụm, v.v.) để khám phá các mẫu, xu hướng, mối tương quan và thông tin chi tiết có giá trị.
- **Trực quan hóa:** Tạo các biểu đồ và đồ thị trực quan hấp dẫn và dễ hiểu để truyền đạt kết quả phân tích một cách hiệu quả.
- **Giải thích và báo cáo:** Diễn giải kết quả phân tích một cách rõ ràng, súc tích và có ý nghĩa, đồng thời đưa ra các khuyến nghị hành động dựa trên dữ liệu nếu có thể.

## EDA với ChatGPT

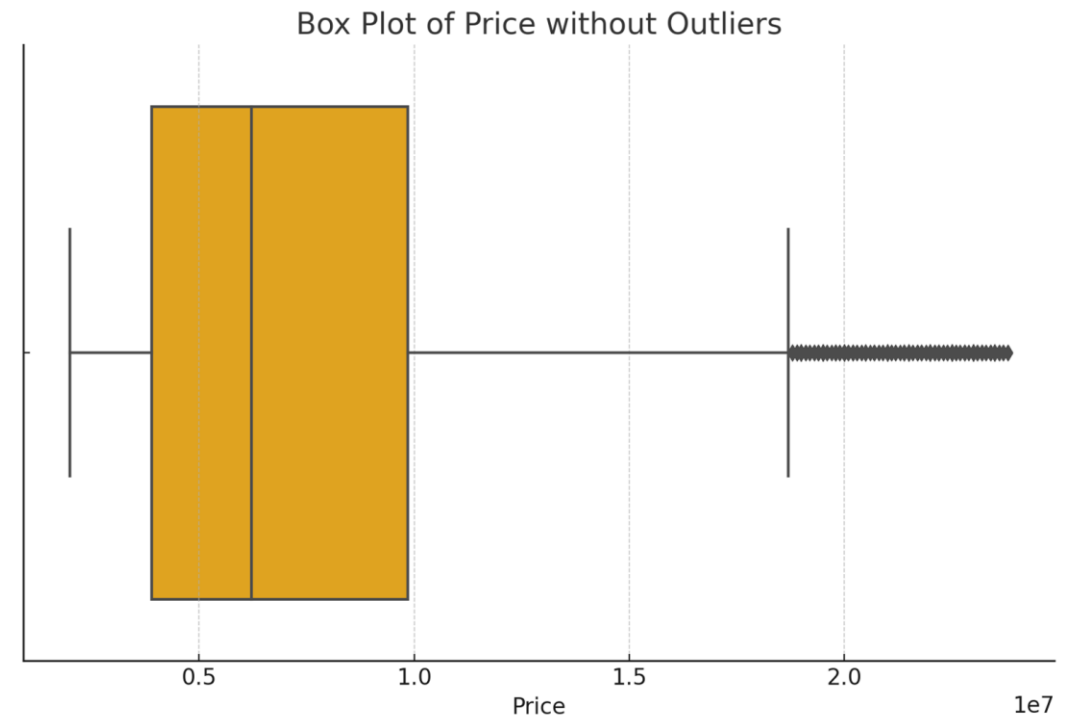
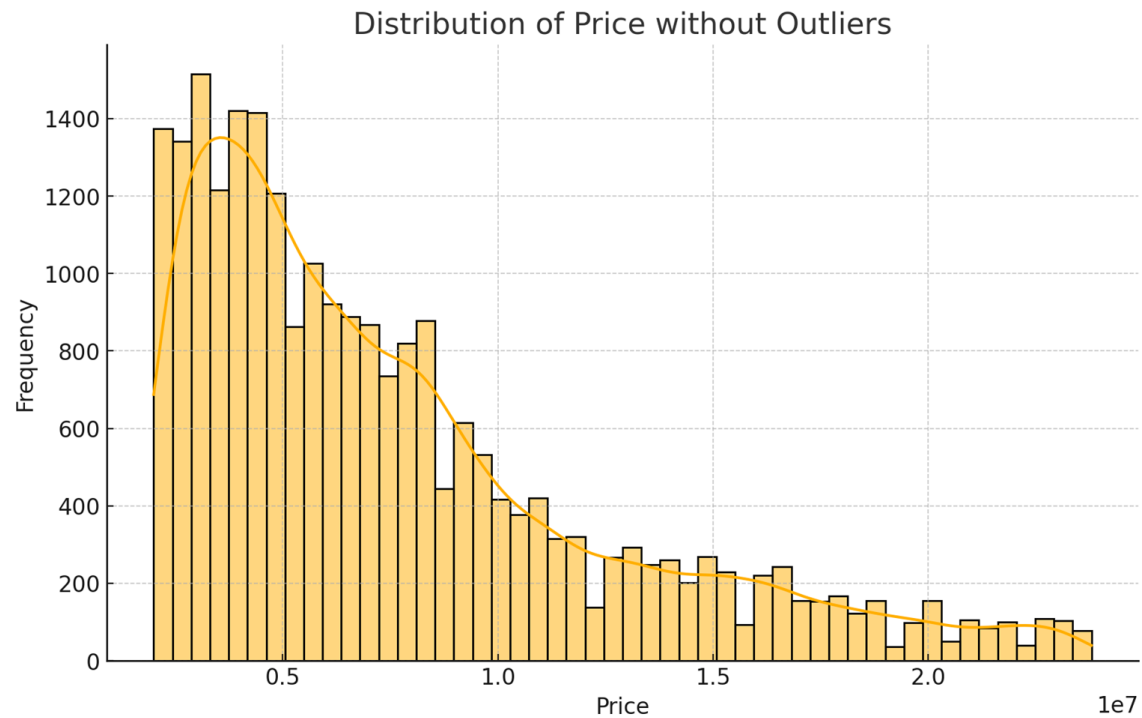
Correlation Matrix



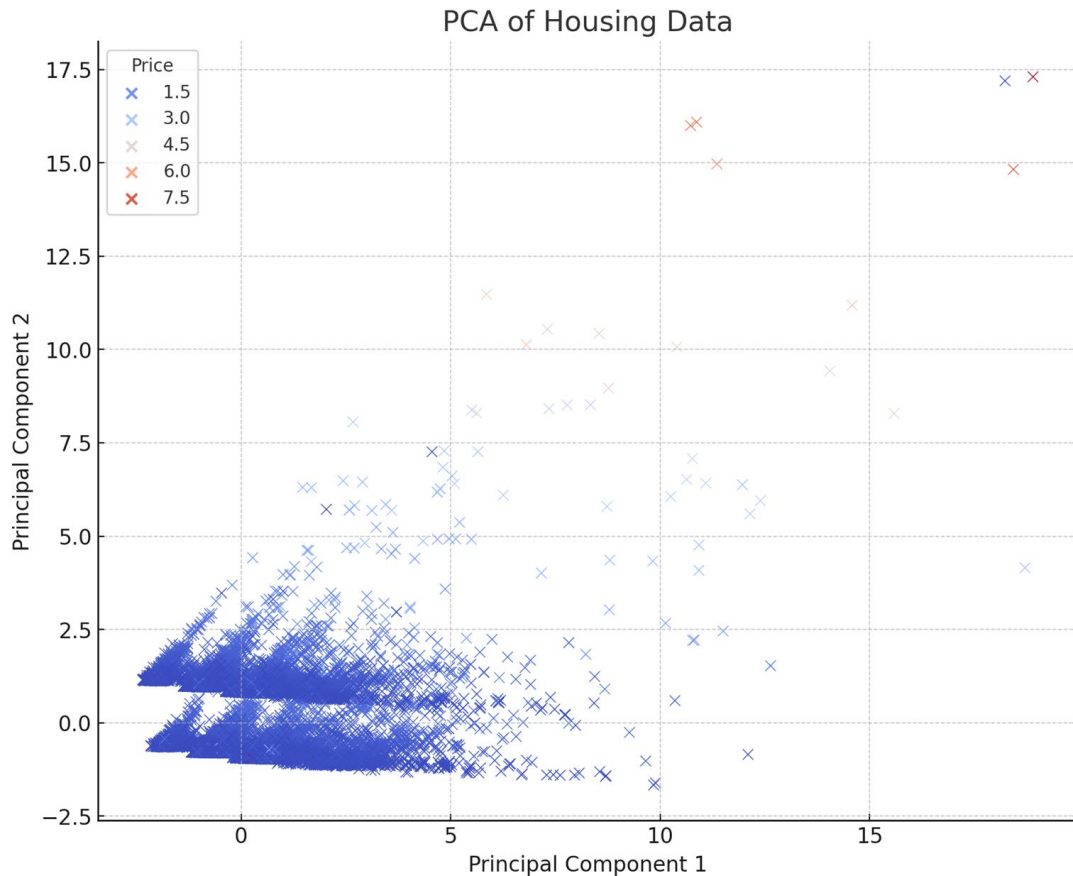
## ❖ EDA với ChatGPT



## ❖Loại bỏ outliers ChatGPT



## ❖ Tìm các mối quan hệ tiềm ẩn với ChatGPT



### Phân tích các mối quan hệ tiềm ẩn:

#### 1. Ma trận tương quan:

- Mối tương quan mạnh giữa diện tích và giá (0.278).
- Các biến khác có mối tương quan yếu hơn với giá.

#### 2. Biểu đồ phân tán:

- **Diện tích vs Giá:** Mối tương quan dương, nhưng có nhiều ngoại lệ và phân tán rộng.
- **Số phòng ngủ vs Giá:** Mối tương quan dương, nhưng cũng có nhiều ngoại lệ.
- **Resale vs Giá:** Không có mối tương quan rõ ràng.

#### 3. Phân tích thành phần chính (PCA):

- PCA cho thấy sự phân tán của dữ liệu theo hai thành phần chính, với các điểm dữ liệu được tô màu theo giá. Tuy nhiên, không có mẫu rõ ràng nào xuất hiện từ phân tích PCA.



## ❖ Feature Engineering với ChatGPT

### Kết quả sau khi tạo các đặc trưng mới:

#### 1. Tạo các đặc trưng mới từ các cột hiện có:

- ``Price_per_Area``: Tỷ lệ giá trên diện tích.
- ``Total_Amenities``: Tổng số lượng tiện ích có sẵn.

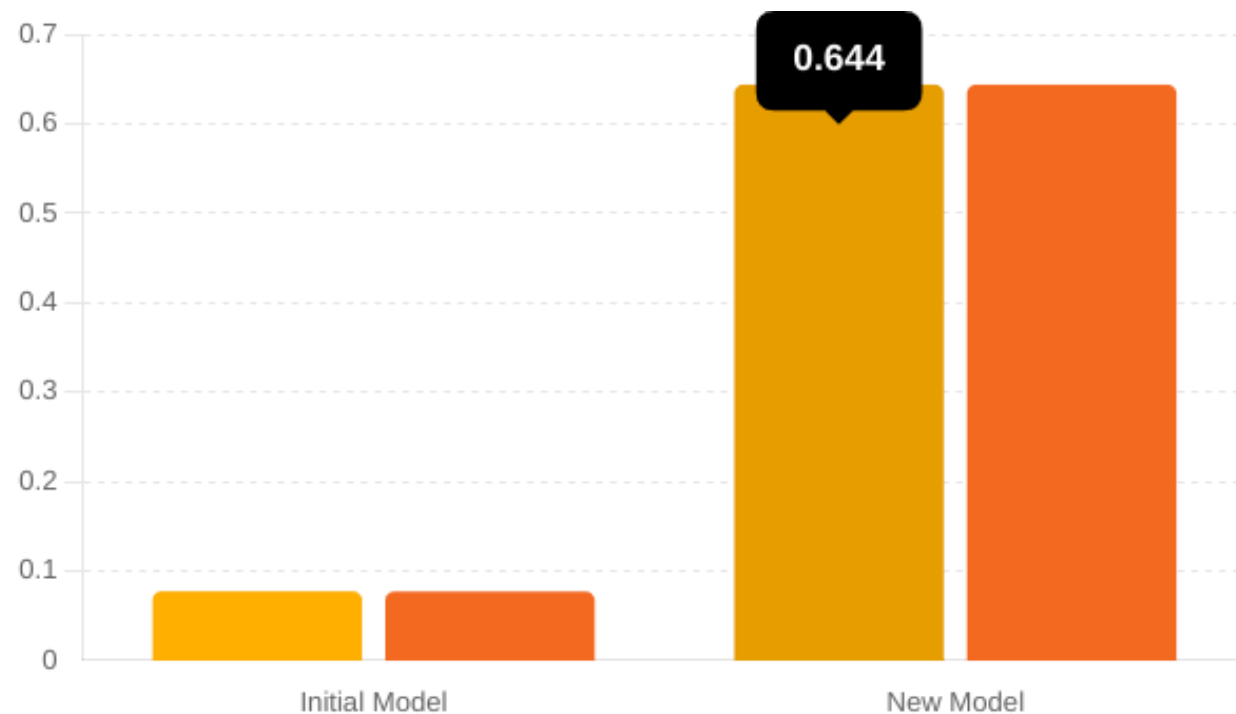
#### 2. Chuyển đổi các cột hiện có:

- ``Location``: Đã được mã hóa thành các cột one-hot.
- ``Resale``: Chuyển đổi thành biến nhị phân.

#### 3. Tạo các đặc trưng tương tác:

- ``Area_Bedrooms_Interaction``: Tương tác giữa diện tích và số phòng ngủ.

Dữ liệu hiện tại đã được tăng cường với các đặc trưng mới, giúp mô hình học máy có thể khai thác thêm nhiều thông tin hữu ích.



## ❖ Công cụ thay thế



## ❖ Công cụ thay thế



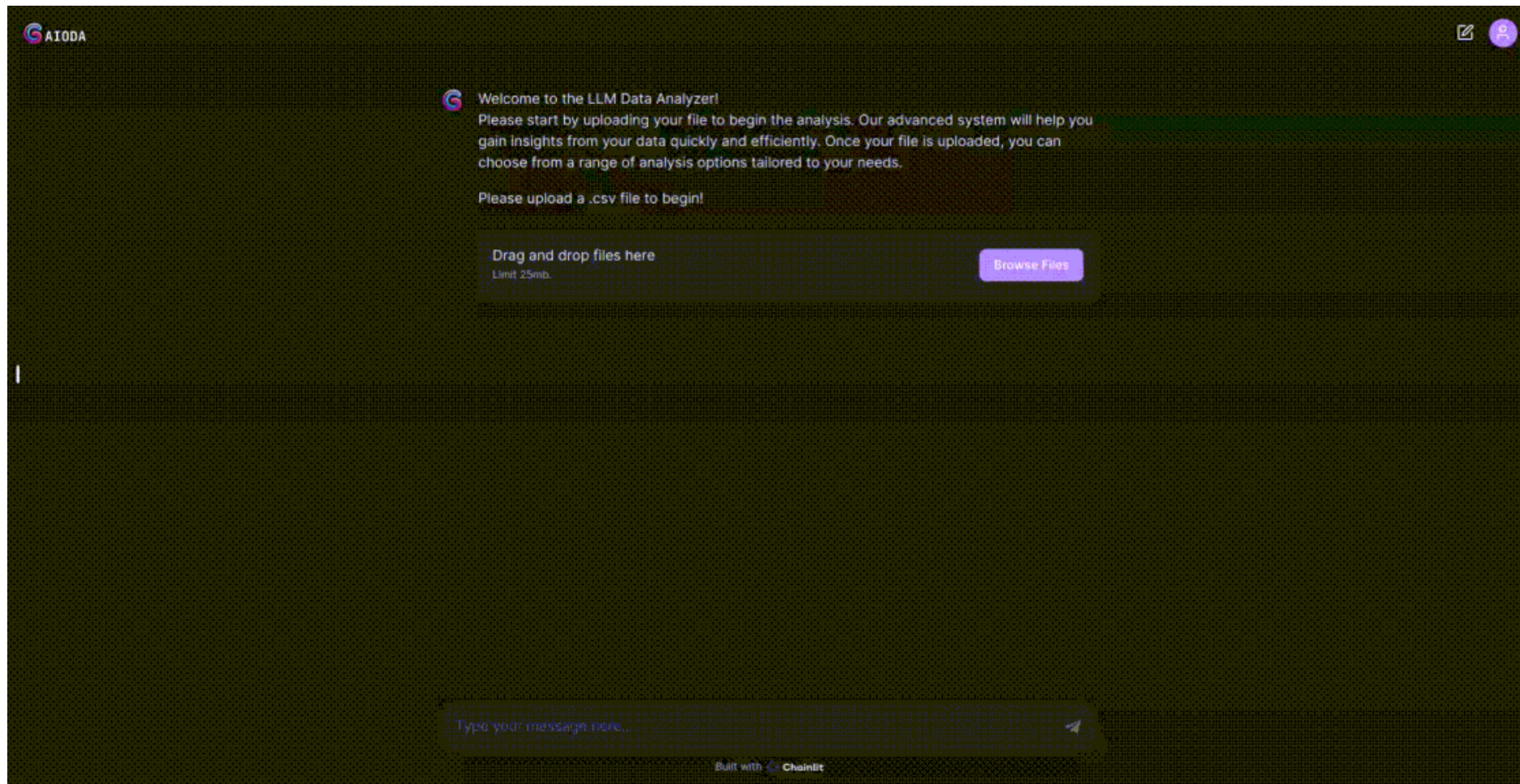
AIO Data Analyst



## ❖ Công cụ thay thế

Free of charge*	Pay-as-you-go (prices in USD)***
<p>Rate Limits**</p> <p>15 RPM (requests per minute) 1 million TPM (tokens per minute) 1,500 RPD (requests per day)</p>	<p>Rate Limits**</p> <p>1000 RPM (requests per minute) 2 million TPM (tokens per minute)</p>
<p>Price (input)</p> <p>Free of charge</p>	<p>Price (input)</p> <p>\$0.35 / 1 million tokens (for prompts up to 128K tokens) \$0.70 / 1 million tokens (for prompts longer than 128K)</p>
<p>Context caching</p> <p>Not applicable</p>	<p>Context caching</p> <p>\$0.0875 / 1 million tokens (for prompts up to 128K tokens) \$0.175 / 1 million tokens (for prompts longer than 128K) \$1.00 / 1 million tokens per hour (storage) <a href="#">Learn more</a></p>
<p>Price (output)</p> <p>Free of charge</p>	<p>Price (output)</p> <p>\$1.05 / 1 million tokens (for prompts up to 128K tokens) \$2.10 / 1 million tokens (for prompts longer than 128K)</p>

## ❖ Công cụ thay thế



# Outline

1

**Why?**

2

**EDA w/ LLMs**

3

**Visualization Tools**

4

**Hands on with code**

**PyGWalker**

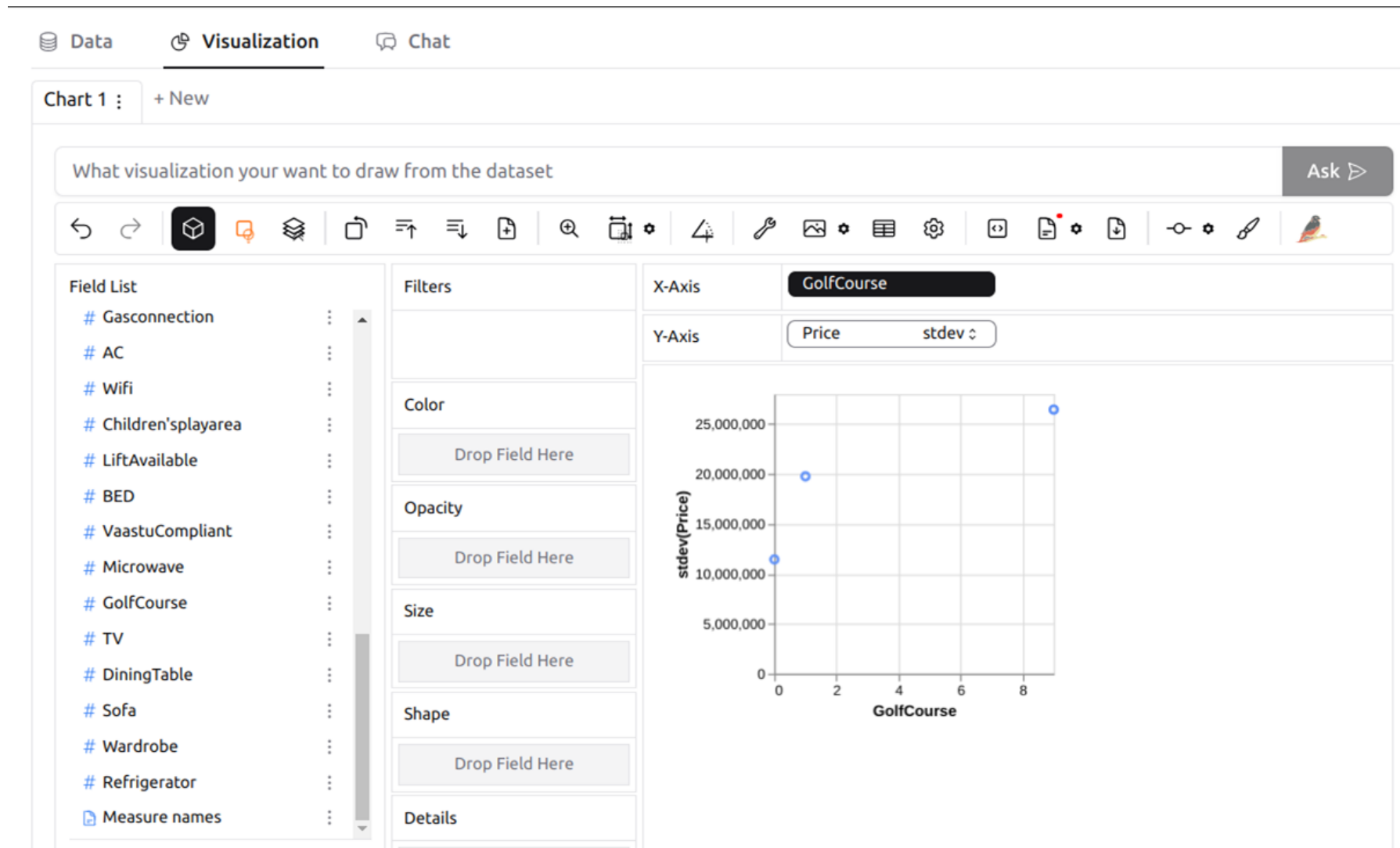
## ❖ Công cụ Visualization

The logo for PyGWalker, featuring the word "PyG" in a pink-to-purple gradient and "Walker" in a dark grey font.

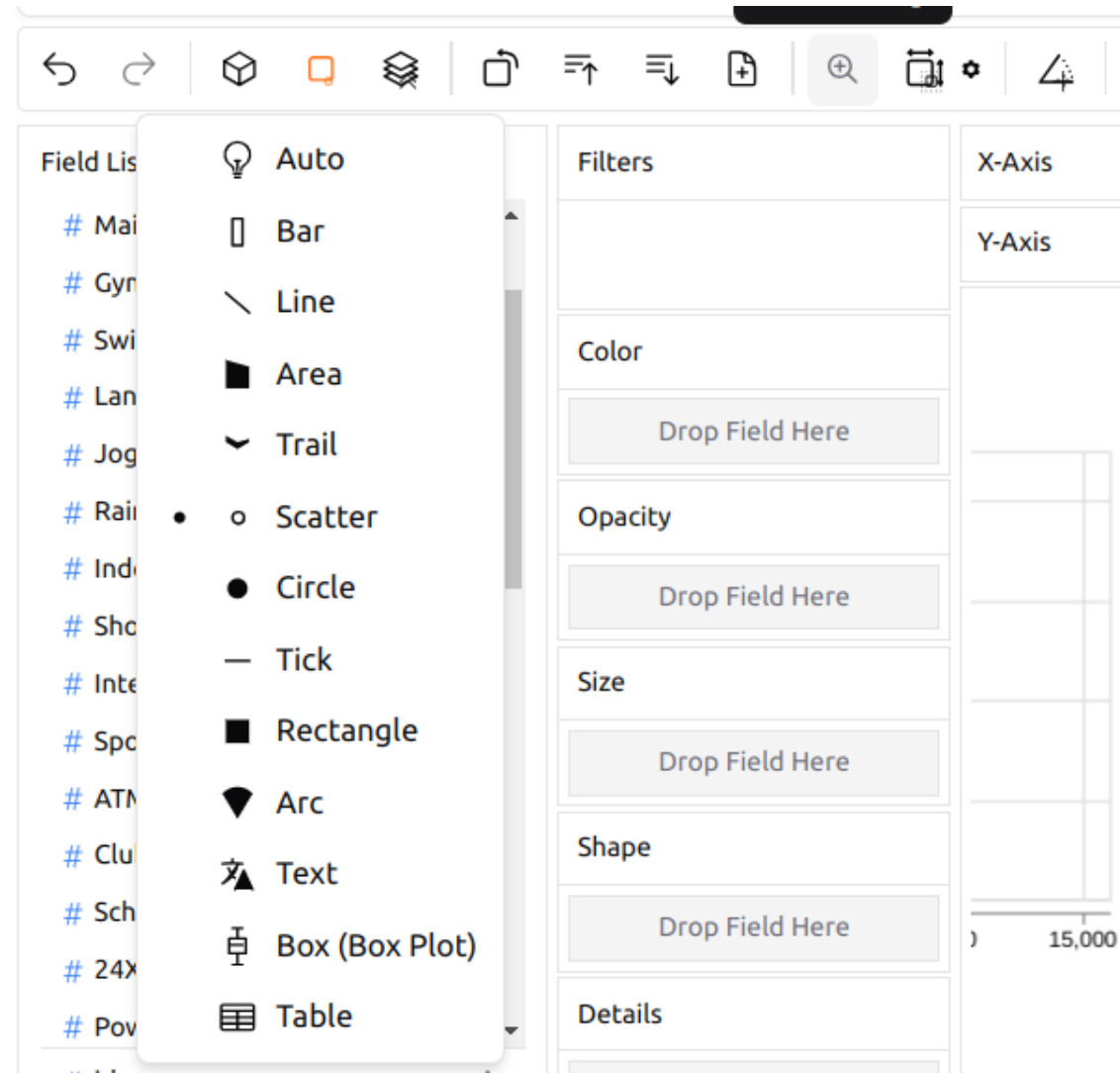
```
1 !pip install pygwalker
```

```
1 import pandas as pd
2 import numpy as np
3 import pygwalker as pg
4
5 data = pd.read_csv('train.csv')
6 pg.walk(data)
```

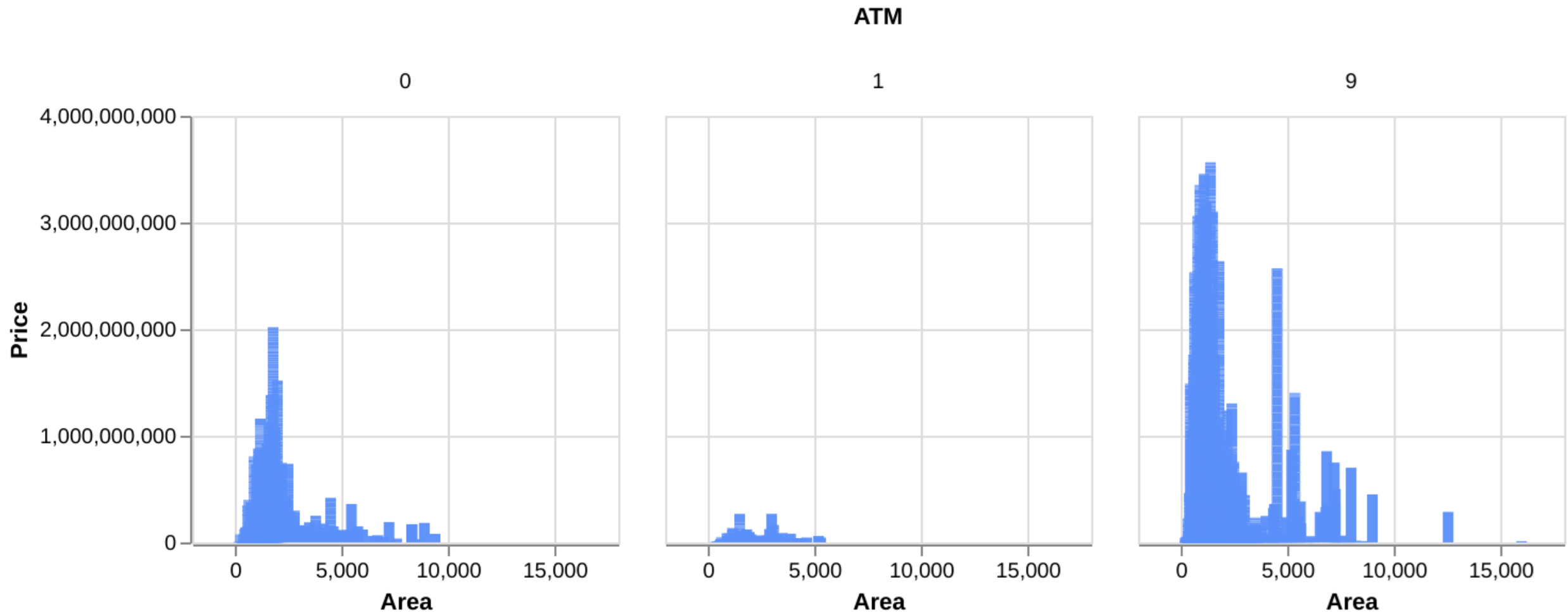
## ❖ Công cụ Visualization



## ❖ Công cụ Visualization



## ❖ Công cụ Visualization

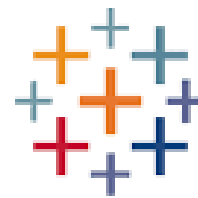


## ❖ Các công cụ khác

Power BI



Qlik® Sense



+ a b l e a u®



# Outline

1

**Why?**

2

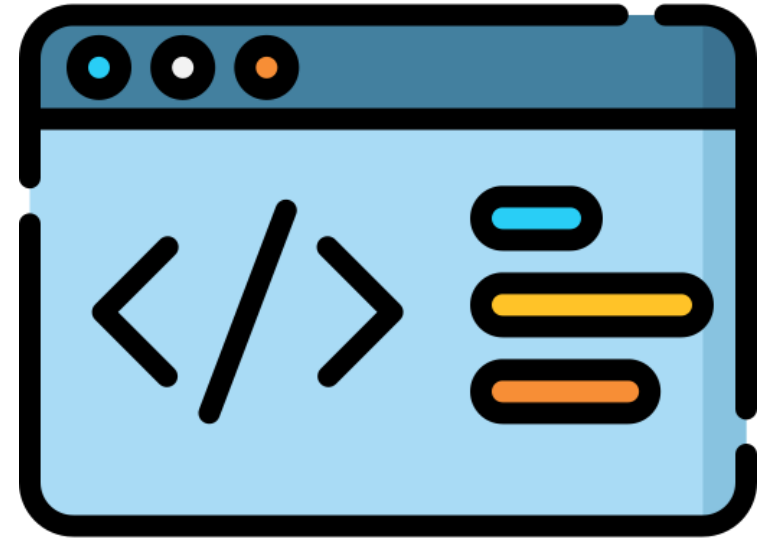
**EDA w/ LLMs**

3

**Visualization Tools**

4

**Hands on with code**



## ❖ Dataset

### Bank Customer Churn Dataset

Bank Customer Data for Predicting Customer Churn

Bộ dataset về dự báo việc một khách hàng có không sử dụng dịch vụ của ngân hàng nữa không.

Một số đặc trưng như sau:

- **credit\_score**: điểm tín dụng
- **country**: quốc gia
- **gender**: giới tính
- ...

Và mục tiêu của bài toán (target) là dự đoán giá trị của cột **churn** (chỉ bao gồm **0** hoặc **1**).

- Nếu giá trị là **0**: Khách hàng tiếp tục sử dụng dịch vụ của ngân hàng
- Nếu giá trị là **1**: Khách hàng không sử dụng dịch vụ nữa.

Trong bài này, chúng ta sẽ thực hành với **pandas** và **seaborn**.

# Hands on with code

## ❖Đọc Data

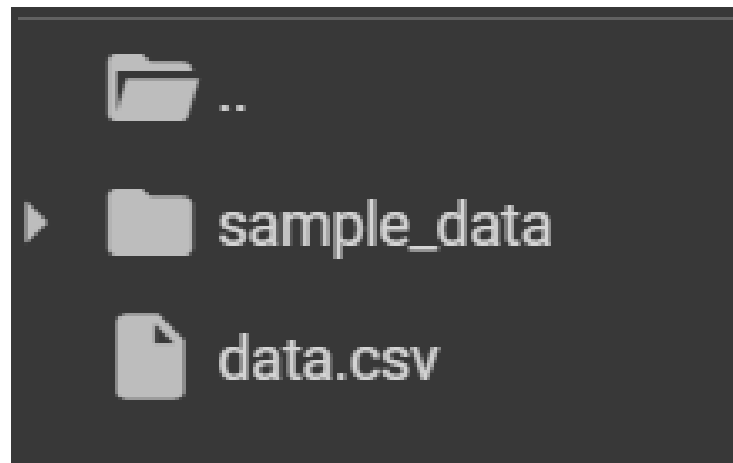
- Import các thư viện cần thiết

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn
import seaborn as sns

import matplotlib.pyplot as plt
%matplotlib inline

plt.style.use('bmh')
```

- Tải và upload bộ dataset: [link](#)



## ❖Đọc Data

Tiến hành đọc data và quan sát một vài samples:

```
df = pd.read_csv('./data.csv')
```

```
df.head(5)
```

	customer_id	credit_score	country	gender	age	tenure	balance	products_number	credit_card	active_member	estimated_salary	churn
0	15634602	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	15647311	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	15619304	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	15701354	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	15737888	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

*5 samples đầu tiên của bộ data*

# Hands on with code



```
df.info()
```

Lấy thông tin tổng quan

```
RangeIndex: 10000 entries, 0 to 9999  
Data columns (total 12 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   customer_id           10000 non-null   int64  
1   credit_score           10000 non-null   int64  
2   country                10000 non-null   object  
3   gender                 10000 non-null   object  
4   age                    10000 non-null   int64  
5   tenure                 10000 non-null   int64  
6   balance                10000 non-null   float64  
7   products_number        10000 non-null   int64  
8   credit_card            10000 non-null   int64  
9   active_member          10000 non-null   int64  
10  estimated_salary        10000 non-null   float64  
11  churn                   10000 non-null   int64
```

```
df.duplicated().sum()
```

Số samples trùng nhau

0

Có thể thấy, bộ dataset có **10000** samples, **12** cột và không có samples nào có giá trị **null** và bị trùng lặp.

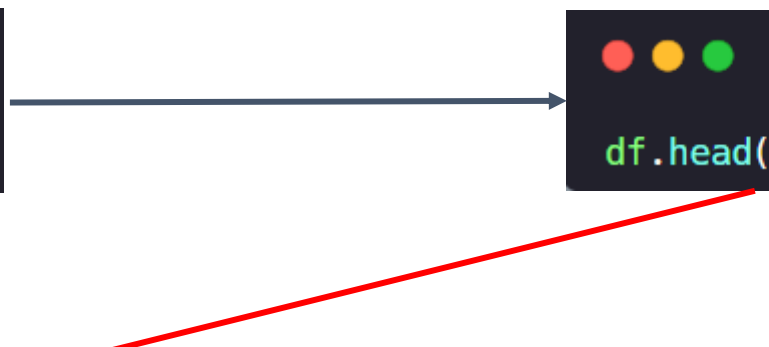
# Hands on with code

## EDA

Chúng ta có thể bỏ cột **customer\_id** vì đây chỉ là những **Id** của khách hàng, không có ý nghĩa nhiều trong bộ dataset

```
df = df.drop(['customer_id'], axis=1)
```

```
df.head(5)
```



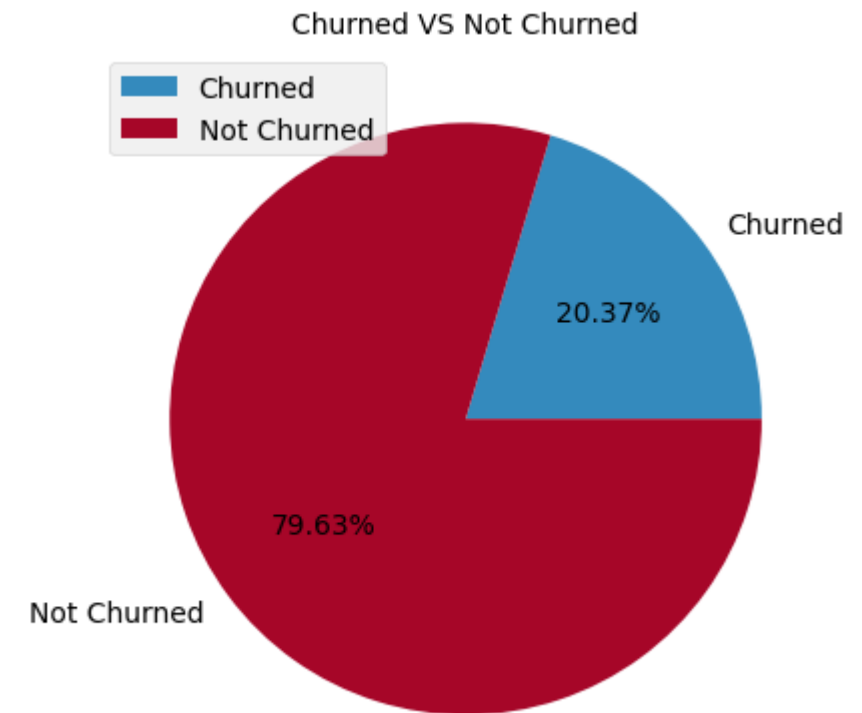
	credit_score	country	gender	age	tenure	balance	products_number	credit_card	active_member	estimated_salary	churn
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0

*Một số samples khi bỏ đi cột **customer\_id***

## EDA

Thống kê phân phối của cột **churn**.

```
 sizes = [df.churn[df['churn'] == 1].count(), df.churn[df['churn'] == 0].count()]
 labels = ['Churned', 'Not Churned']
 plt.pie(sizes, labels=labels, autopct = '%.2f%%')
 plt.legend(loc='upper left')
 plt.title("Churned VS Not Churned", size = 10)
 plt.show()
```

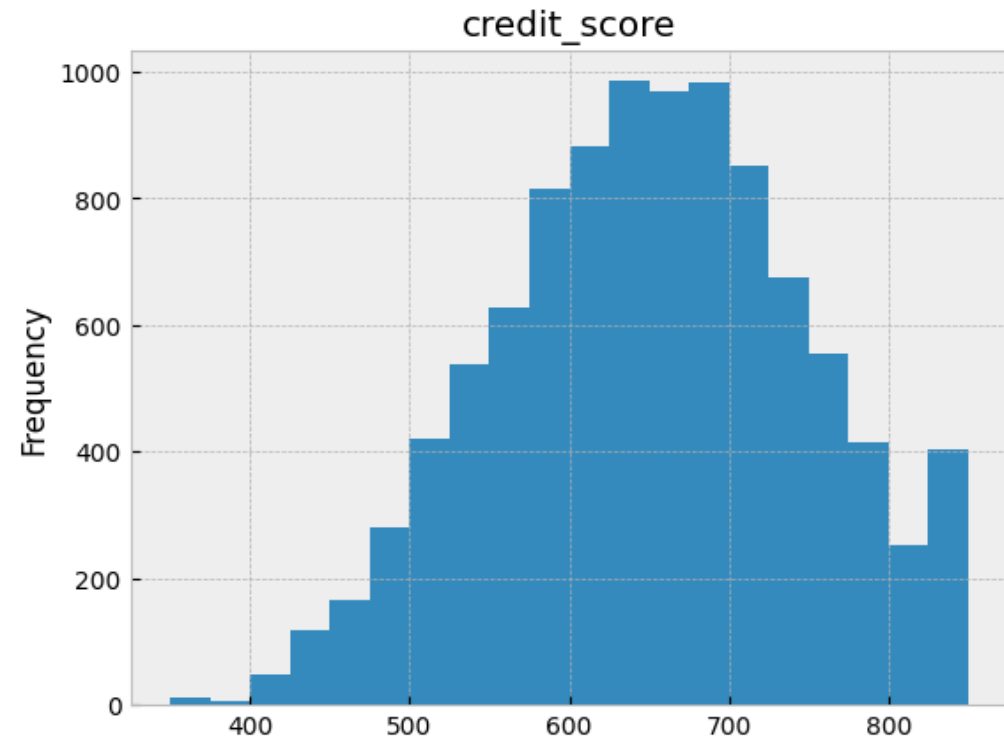


⇒ Số lượng samples có giá trị **churn=0** (Not Churned) gấp gần 4 lần số lượng samples có giá trị còn lại.

## EDA

Biểu diễn phân phối của cột **credit\_score**.

```
df['credit_score'].plot(kind='hist', bins=20, title='credit_score')
```



- Ngưỡng điểm sẽ nằm trong khoảng từ 600 đến 700 điểm.
- Có khá ít giá trị thấp hơn 400.



Liệt kê các giá trị của cột **country**:

```
countries = list(df['country'].unique())  
countries
```

```
['France', 'Spain', 'Germany']
```

Đếm số lượng samples của mỗi quốc gia:

```
df['country'].value_counts()
```

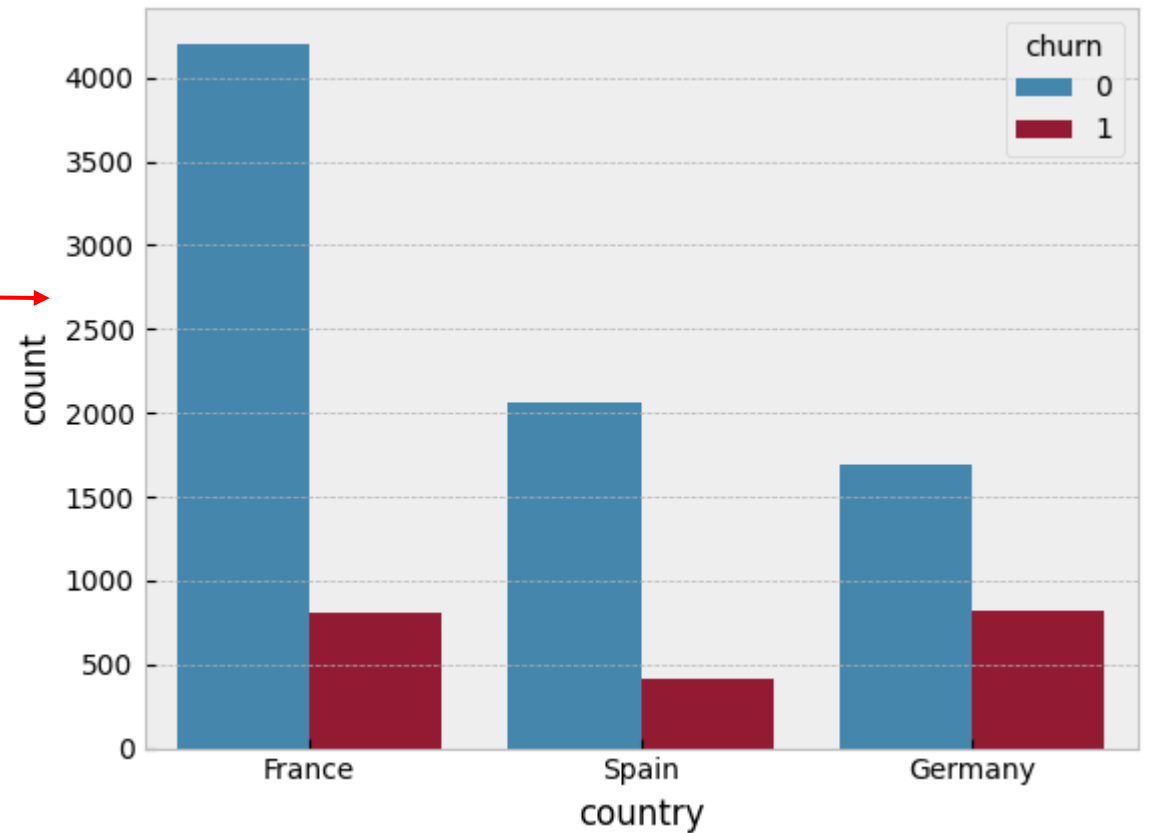
```
country  
France    5014  
Germany   2509  
Spain     2477  
Name: count, dtype: int64
```

- Có **ba** quốc gia trong bộ dataset.
- Trong đó, **France** là quốc gia có số lượng samples nhiều nhất.

## EDA

Thống kê số lượng từng sample mỗi nhãn ứng với từng quốc gia:

```
sns.countplot(x='country', hue='churn', data=df)
```





# Hands on with code

## ❖ EDA

Vẽ biểu đồ tròn cho từng quốc gia:

```
fig, axes = plt.subplots(1, len(countries), figsize=(15, 5))

for ax, country in zip(axes, countries):
    # Filter the dataframe by country
    country_data = df[df['country'] == country]

    # Count the churn values
    churn_counts = country_data['churn'].value_counts()

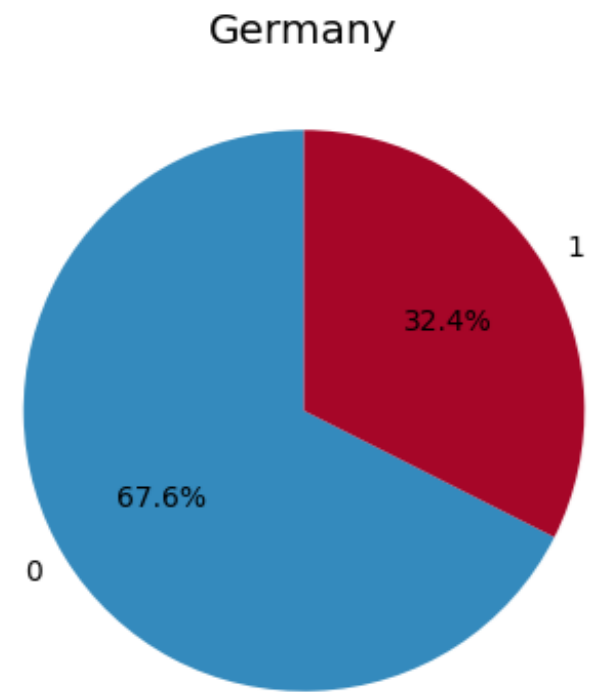
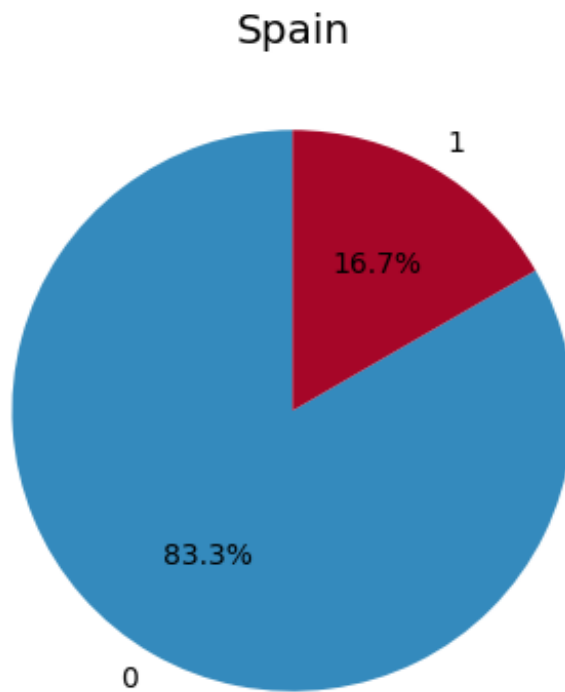
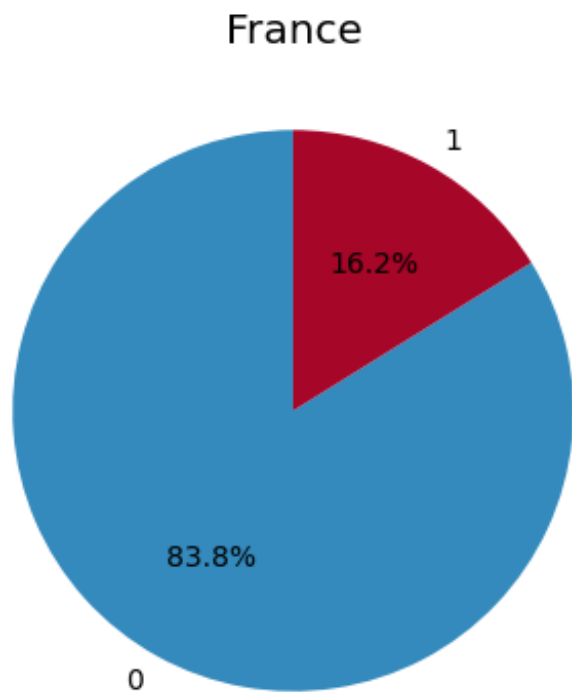
    # Plot the pie chart
    ax.pie(churn_counts, labels=churn_counts.index, autopct='%1.1f%%', startangle=90)
    ax.set_title(f'{country}')

plt.show()
```

# Hands on with code

## ❖ EDA

Vẽ biểu đồ tròn cho từng quốc gia:



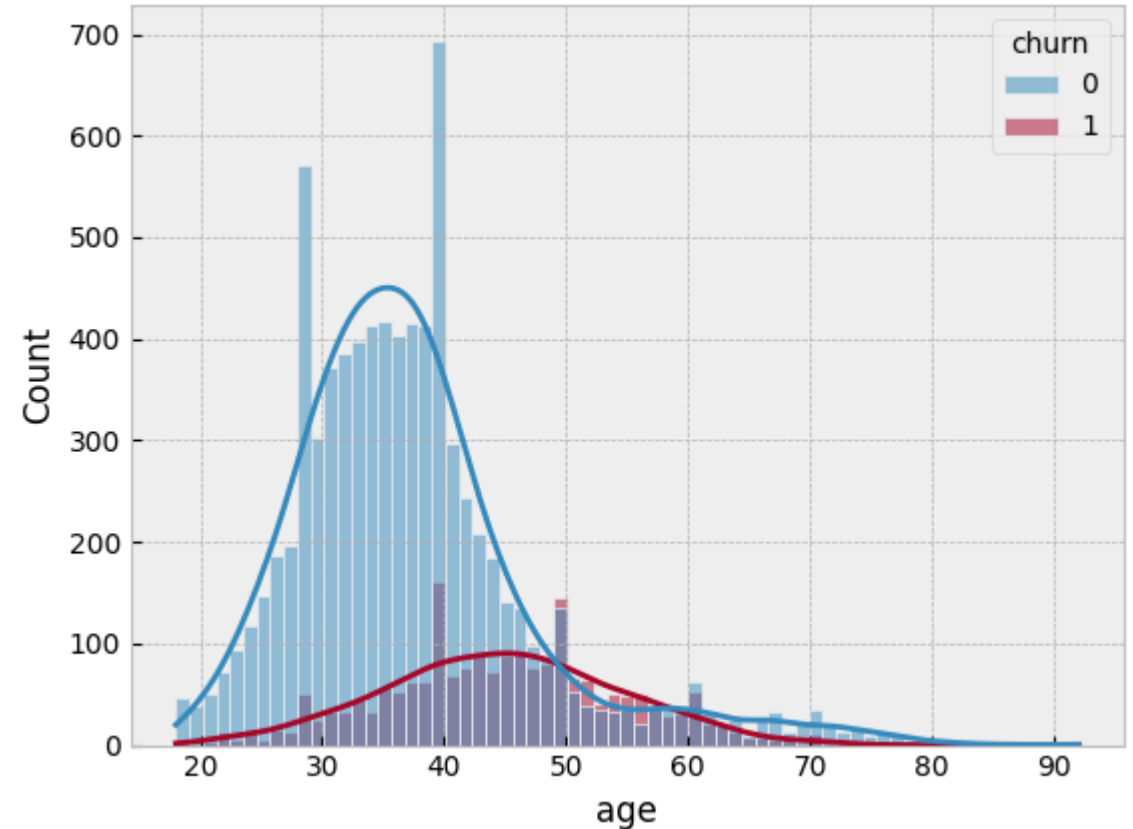
⇒ Qua các biểu đồ trên, có thể thấy **Germany** tuy có số lượng samples không nhiều nhưng tỉ lệ **churn** tương đối cao.

## EDA

Vẽ biểu đồ phân bố độ tuổi **age**

```
sns.histplot(data=df, x='age', hue="churn", kde=True)
```

⇒ Những người trong độ tuổi từ **40 đến 60** tuổi có tỉ lệ **churn** cao hơn các nhóm tuổi còn lại.



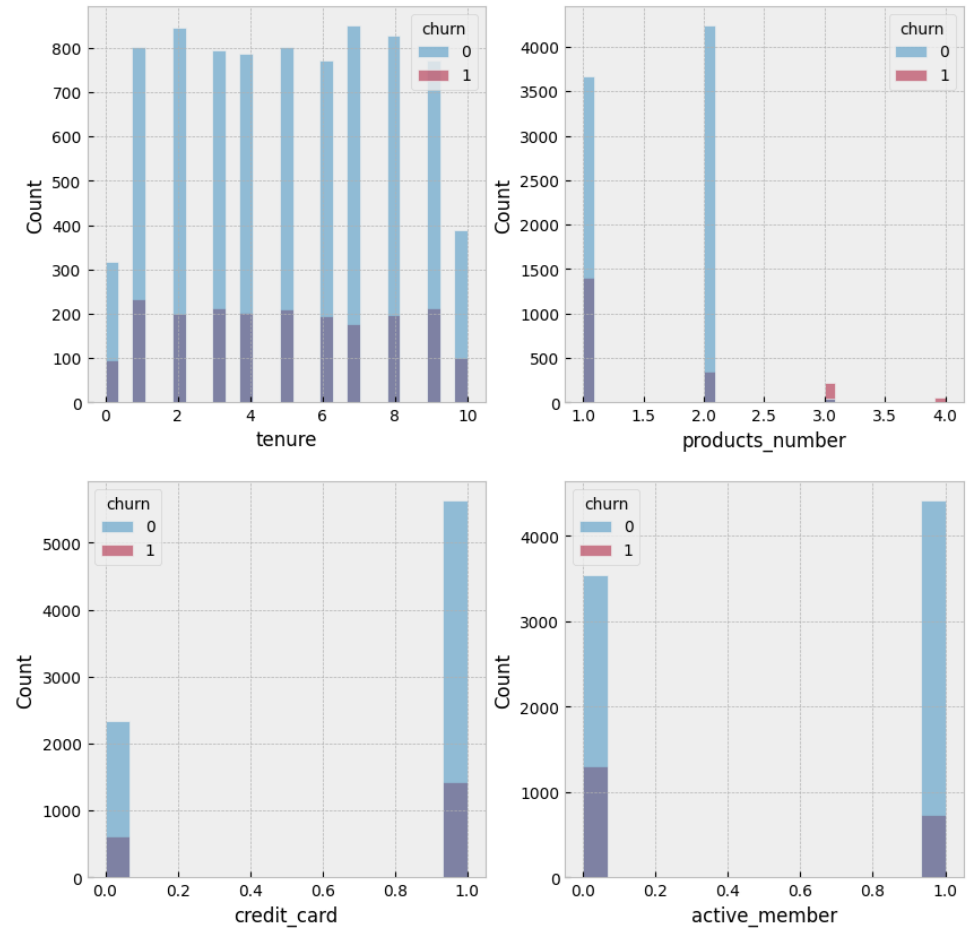
## EDA

Tương tự cho các cột khác:

```
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))
axs = axs.flat

columns = ["tenure", "products_number", "credit_card", "active_member"]

for i in range(len(columns)):
    sns.histplot(data=df, x=columns[i], hue="churn", ax=axs[i])
```



## ❖ EDA

Một số nhận xét từ biểu đồ bên:

### 1. **tenure:**

- Phân bố các samples của cột **churn** không thay đổi nhiều với các giá trị **tenure** khác nhau

### 1. **products\_number:**

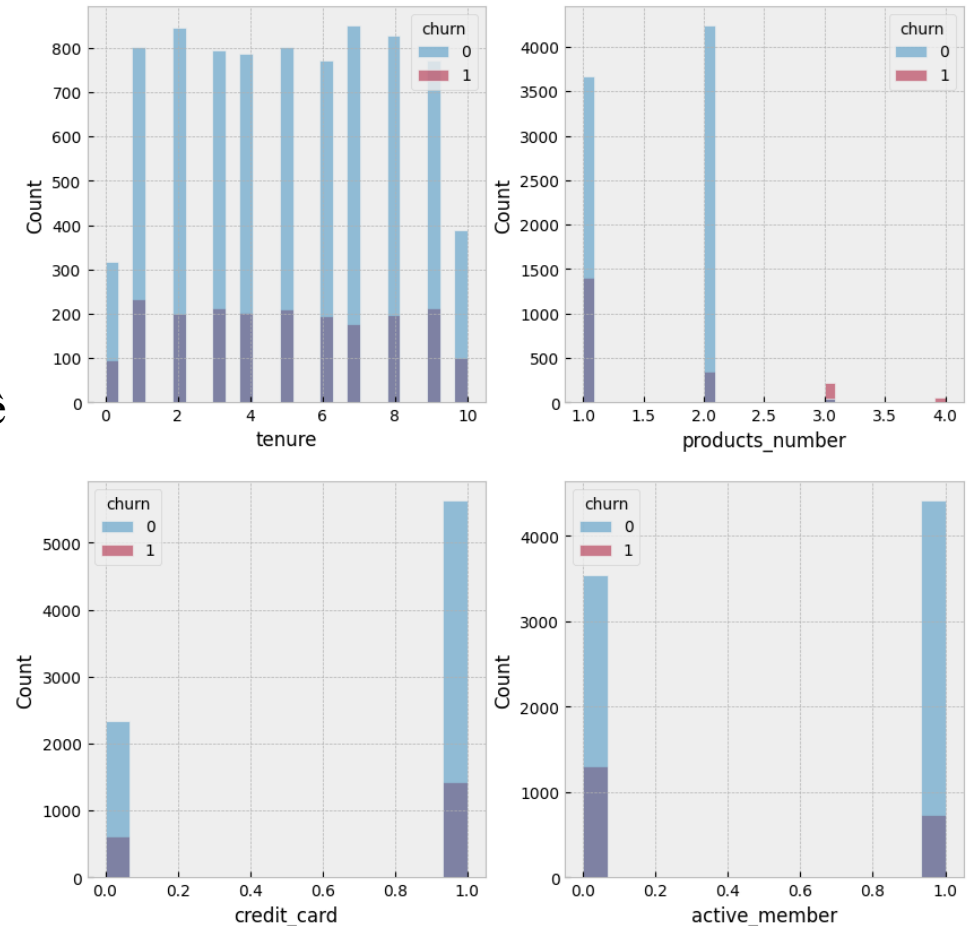
- Trong tập dataset này, hầu hết khách hàng chỉ có **một** hoặc **hai** sản phẩm.
- Có một xu hướng cho thấy khách hàng chỉ có **một** sản phẩm có tỷ lệ **churn** cao hơn so với những người có **hai** sản phẩm.
- Khách hàng có hơn **hai** sản phẩm có xu hướng tỷ lệ **churn** cao hơn.

### 1. **credit\_card**

- Khách hàng không có thẻ tín dụng có tỷ lệ **churn** cao hơn một chút so với những người có thẻ tín dụng.

### 1. **active\_member**

- Thành viên không hoạt động có tỷ lệ **churn** cao hơn đáng kể so với thành viên hoạt động.



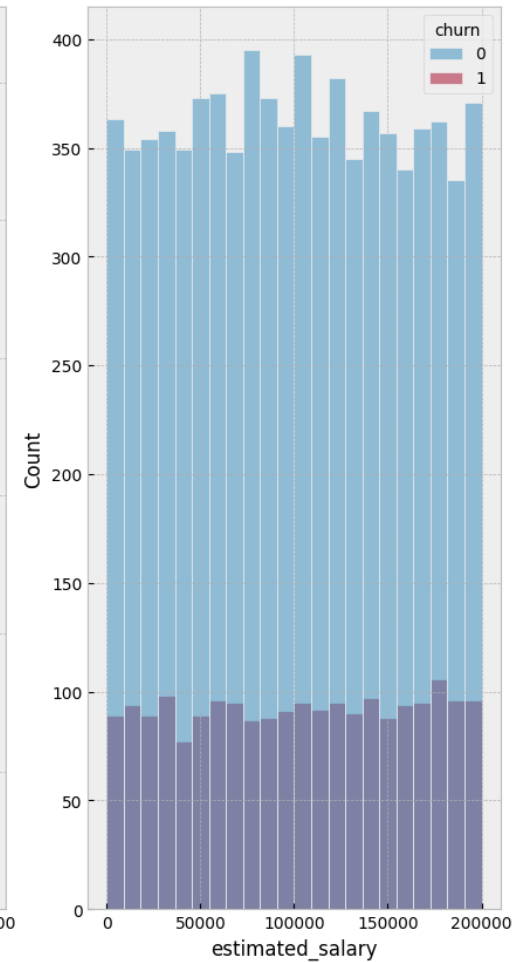
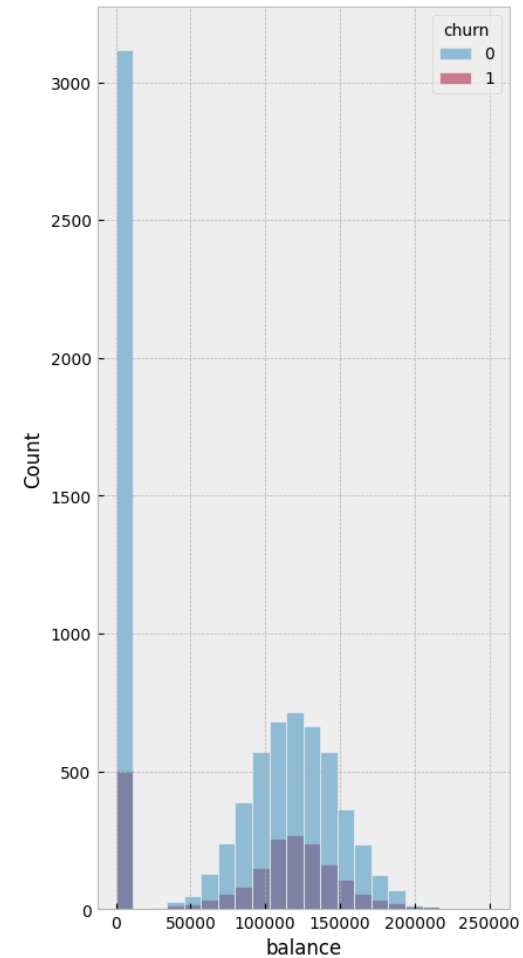
# Hands on with code



```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(10, 10))
axs = axs.flat

columns = ["balance", "estimated_salary"]

for i in range(len(columns)):
    sns.histplot(data=df, x=columns[i], hue="churn", ax=axs[i])
```





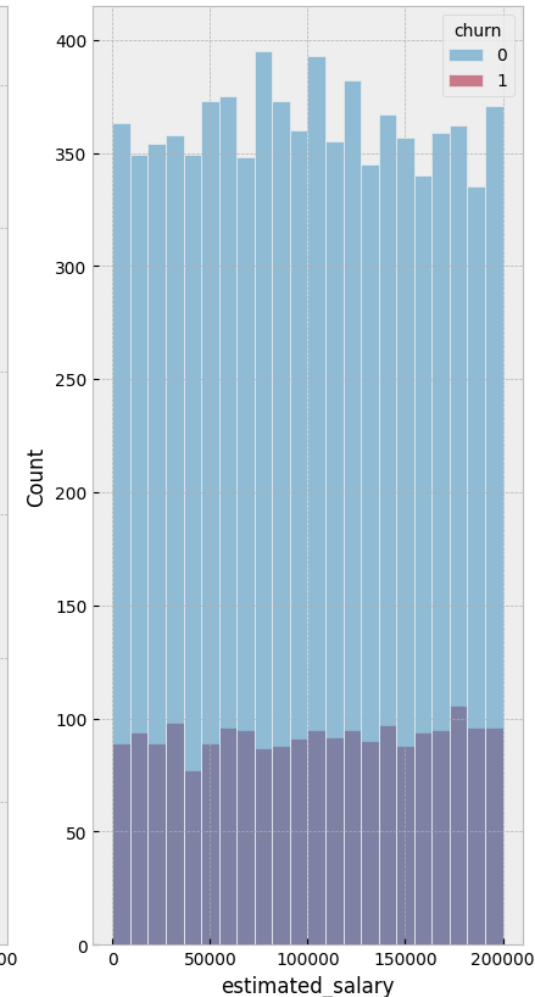
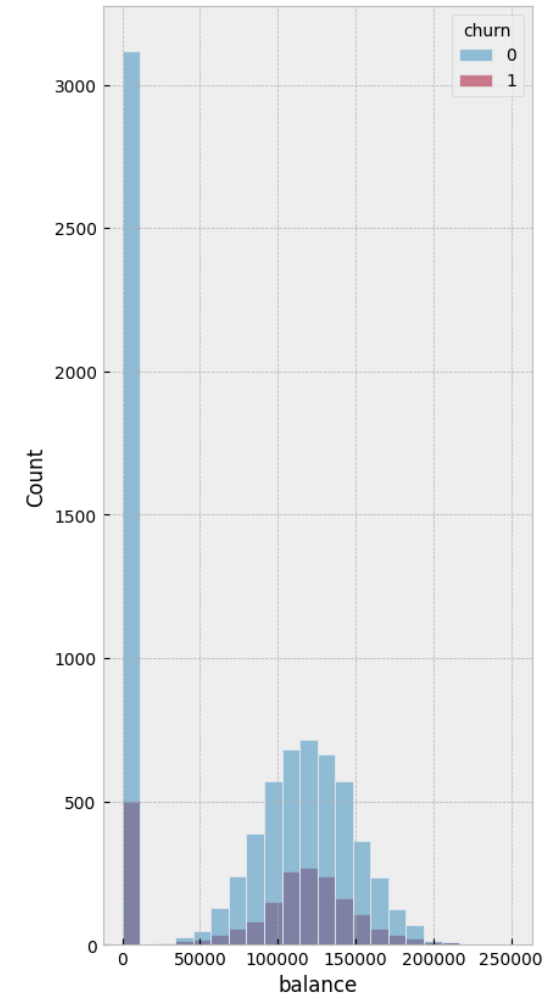
Từ biểu đồ bên, có thể thấy rằng:

## 1. **balance**

- Tỷ lệ **churn** có xu hướng tăng lên khi số dư tài khoản cao hơn, cho thấy khách hàng có số dư cao hơn có thể có khả năng chuyển khách hàng cao hơn (churn).

## 1. **estimated\_salary**

- Không có sự khác biệt đáng kể giữa các nhóm thu nhập thấp và cao hơn.



Heatmap:

```
from sklearn.preprocessing import LabelEncoder

df['country'] = LabelEncoder().fit_transform(df['country'])
df['gender'] = LabelEncoder().fit_transform(df['gender'])

correlation_matrix = df.corr()
plt.figure(figsize=(10, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

# Hands on with code

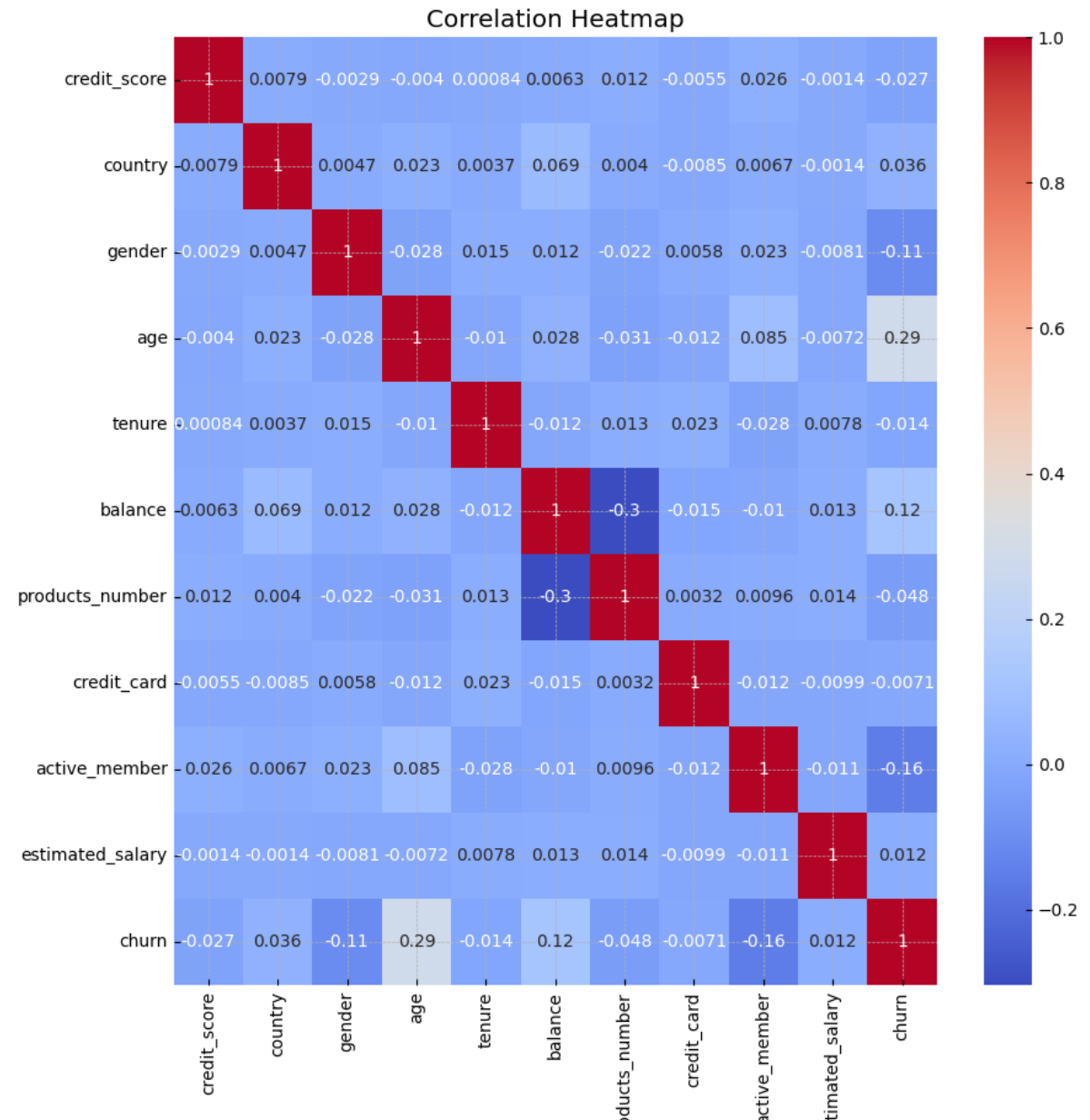
## EDA

Heatmap:

Từ heatmap, chúng ta có thể thấy rằng:

- **age (0.29)**: Tuổi có mối tương quan mạnh nhất với **churn** giữa các đặc trưng, cho thấy những khách hàng già hơn có tỉ lệ **churn** cao hơn.
- **active\_member (-0.16)**: Gợi ý rằng thành viên hoạt động có khả năng chuyển khách hàng (churn) thấp hơn.

Các đặc trưng khác như **credit\_card**, **products\_number** và **estimated\_salary** (ước tính thu nhập) có mối tương quan yếu với **churn**, cho thấy chúng có thể không phải là các đặc trưng quan trọng.



QUIZ TIME

