Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Отчёт к лабораторным работам по курсу

«Методы машинного обучения»

Лабораторная работа №1 «Создание "истории о данных"»

Выполнил:

студент(ка) группы ИУ5И-21М Лю Бэйбэй

подпись, дата

Проверил:

к.т.н., доц., Виноградовой М.В.

подпись, дата

Москва, 2022 г.

1. описание задания

Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь.

Для лабораторных работ не рекомендуется выбирать датасеты очень большого размера.

Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
4. Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.
5. История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

Сформировать отчет и разместить его в своем репозитории на github.

2. Текст программы и экранные формы с примерами выполнения программы.

Импортирование необходимых библиотек.

```
import pandas as pd
import random
import matplotlib.pyplot as plt
import math as math
import seaborn as sns
import numpy as np
%matplotlib inline
import re
```

Импортирование данных.

```
url = "E:\liu\lab\lab1\Customers.csv"
dataset = pd.read_csv(url)
dataset=dataset.head(1000)
dataset
```

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
0	1	Male	19	15000	39	Healthcare	1	4
1	2	Male	21	35000	81	Engineer	3	3
2	3	Female	20	86000	6	Engineer	1	1
3	4	Female	23	59000	77	Lawyer	0	2
4	5	Female	31	38000	40	Entertainment	2	6
...	...	...	...	...	...	...	...	...
995	996	Male	65	56583	65	Healthcare	2	3
996	997	Female	17	185843	36	Artist	4	3
997	998	Female	31	171825	22	Entertainment	10	4
998	999	Male	24	77976	95	Artist	11	3
999	1000	Male	97	66312	75	Entertainment	0	1

1000 rows × 8 columns

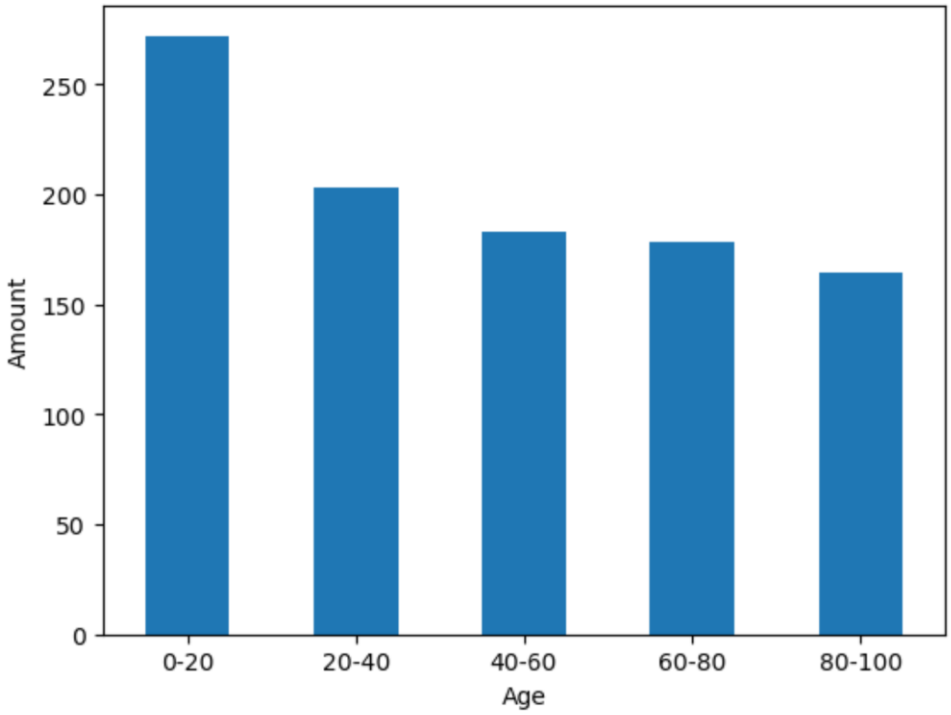
Дезагрегировать данные по возрасту и подсчитали количество данных в каждой категории.

```
x=['0-20','20-40','40-60','60-80','80-100']
ages=pd.cut(dataset['Age'], [0,20,40,60,80,100], right=False, include_lowest=True)
print(ages.value_counts())
```

```
[20, 40)     272
[40, 60)     203
[0, 20)      183
[60, 80)     178
[80, 100)    164
Name: Age, dtype: int64
```

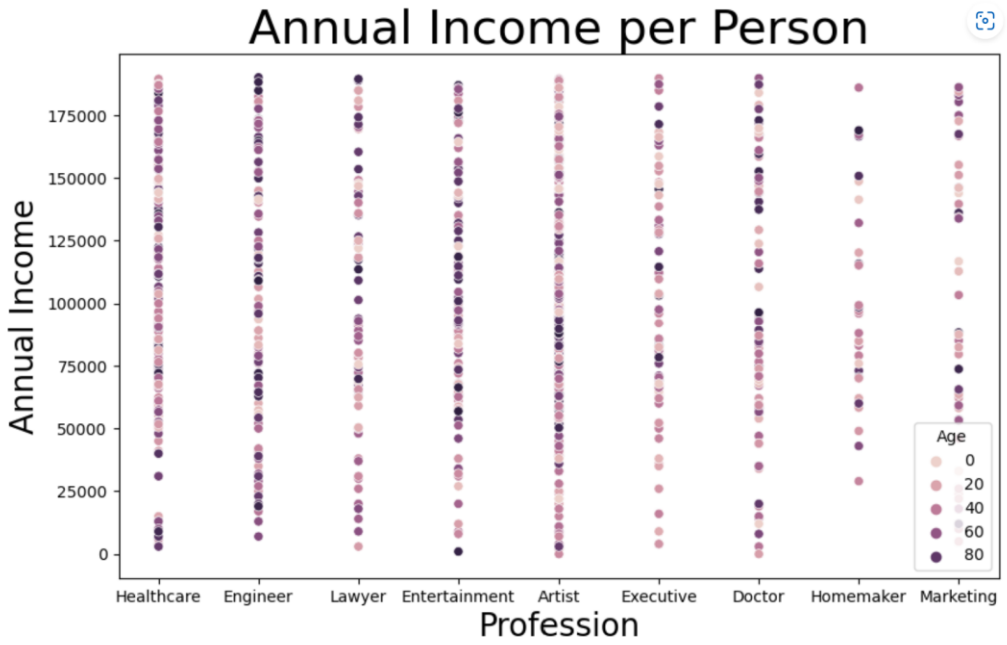
Создание гистограмм.

```
values=ages.value_counts().values
df=pd.DataFrame(values,index=x)
df.plot(kind='bar',legend=False)
plt.xticks(rotation=0)
plt.xlabel('Age')
plt.ylabel('Amount')
plt.show()
```



Визуализация распределения годового дохода с помощью диаграммы рассеяния.

```
plt.figure(figsize=(10,6))
sns.scatterplot(x="Profession",y="Annual Income ($)",hue="Age",data=dataset)
plt.title("Annual Income per Person",fontsize=30)
plt.xlabel("Profession",fontsize=20)
plt.ylabel("Annual Income",fontsize=20)
plt.show()
```



Визуализация с помощью тепловой карты, корреляция отдельных столбцов.

Дезагрегировать данные по полу и подсчитать количество в каждой категории.

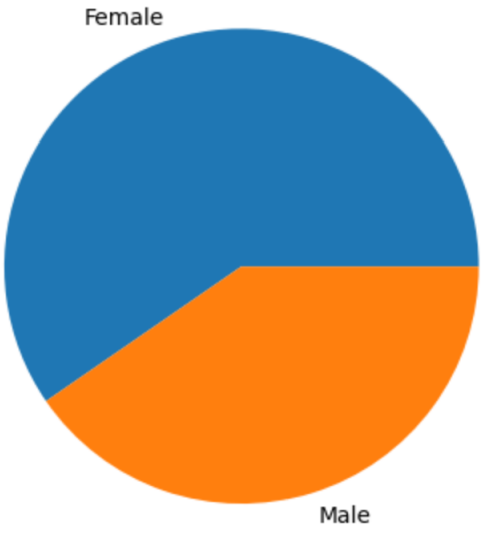
```
gender = dataset['Gender'].value_counts()
gender
```

```
Female    596
Male      404
Name: Gender, dtype: int64
```

Создание круговых диаграмм

```
count=gender.array
list_gender = gender.index.to_list()
plt.pie(count,labels=list_gender)
```

```
([<matplotlib.patches.Wedge at 0x180322b4d60>,
  <matplotlib.patches.Wedge at 0x180322b4430>],
 [Text(-0.32674579165876383, 1.050350983068654, 'Female'),
  Text(0.3267458899997187, -1.0503509524765007, 'Male')])
```



Визуализация распределения индексов потребления с помощью коробочных диаграмм.

```
plt.figure(figsize=(11,7))
sns.boxplot(x="Family Size", y="Spending Score (1-100)",
            hue="Gender",data=dataset)
```

```
<AxesSubplot:xlabel='Family Size', ylabel='Spending Score (1-100)'>
```