

## ЛАБОРАТОРНА РОБОТА № 3

**Тема:** Введення в аналіз даних на Python.

### Хід роботи:

Pandas - це бібліотека Python, що надає можливості для проведення аналізу даних. За її допомогою зручно завантажувати, проводити обробку та аналізувати табличні дані за допомогою SQL-подібних запитів.

Основними структурними даними в Pandas є класи Series та DataFrame. Перший з них являє собою одновимірний масив даних деякого фіксованого типу. Другий - це двовимірна структура даних, що являє собою таблицю, кожен стовпчик якої вміщує дані одного типу. За допомогою бібліотеки Pandas проведемо аналіз даних. Працювати будемо з даними про клієнтів банку, що цікавиться чи буде рахуватись заборгованість по платежу на 90 та більше днів при видачі кредиту.

### Задача 1.

Прочитайте дані з файлу data.csv

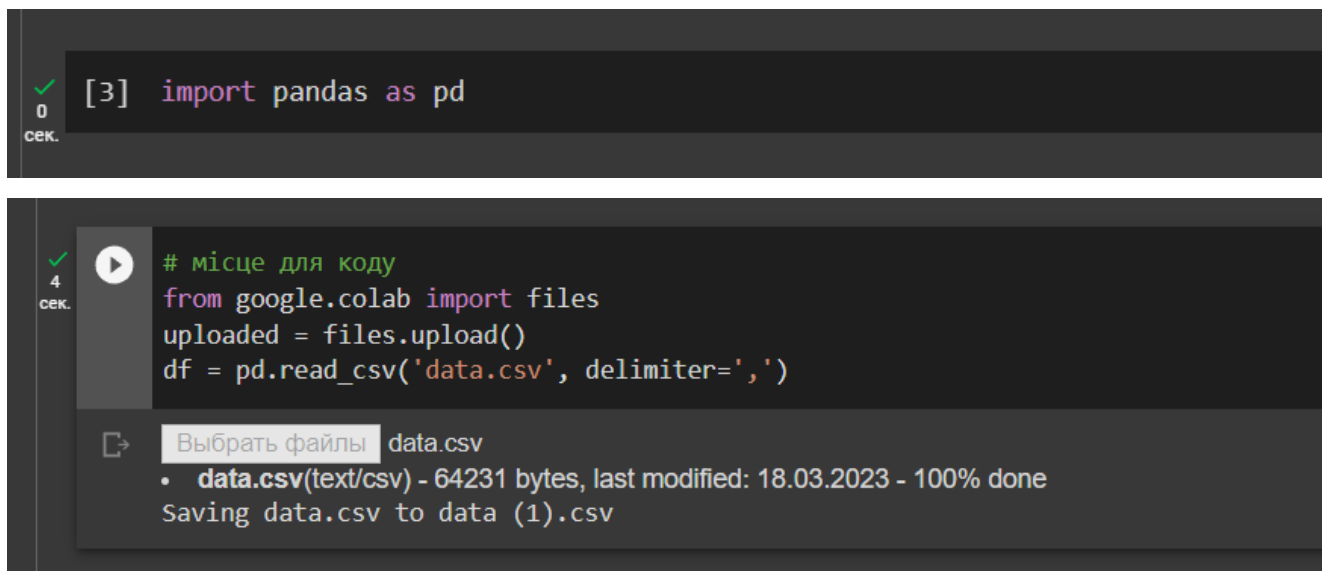


Рис.1.1. Зчитування файлу

					ДУ «Житомирська політехніка».23.122.08.000 – Лр3						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Дяченко В.В.			Звіт з лабораторної роботи			Лім.	Арк.	Аркушіє	
Перевір.										1	
Керівник								ФІКТ Гр. КН-20-1(1)			
Н. контр.											
Зав. каф.											

Задача 2.

Виведіть опис даних, що було прочитано.

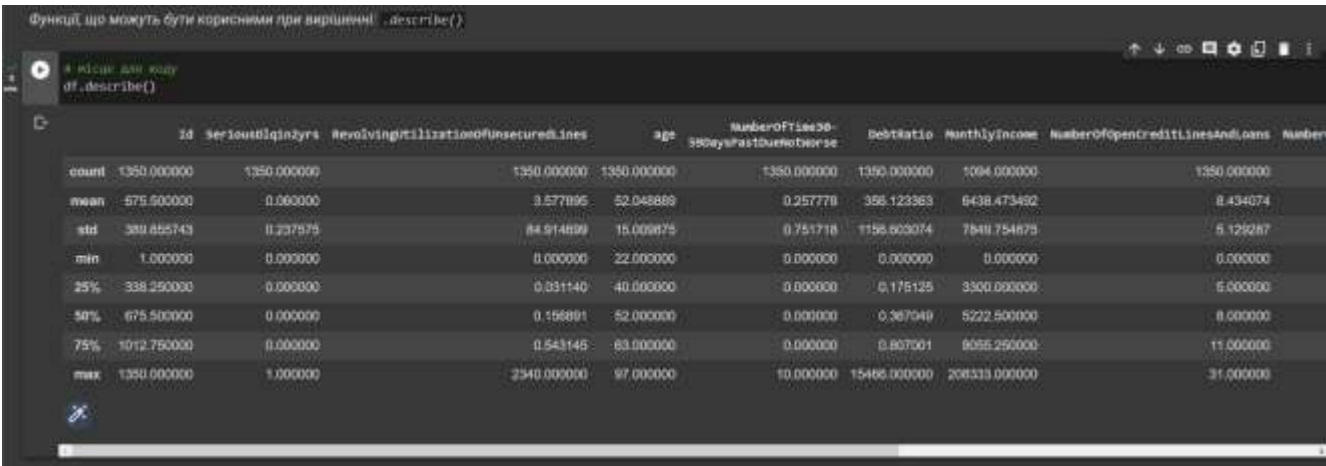


Рис.2.1. Вивід даних.

Задача 3.

Відобразіть декілька перших та декілька останніх записів.

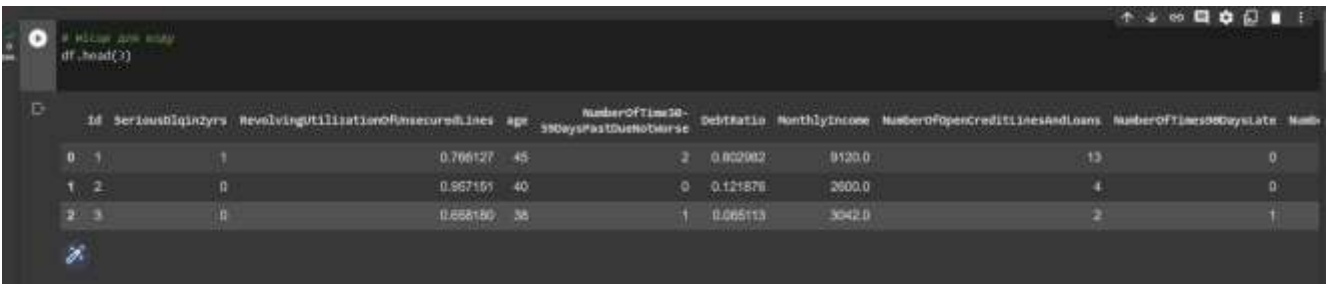


Рис.3.1. Вивід 3 перших записів.

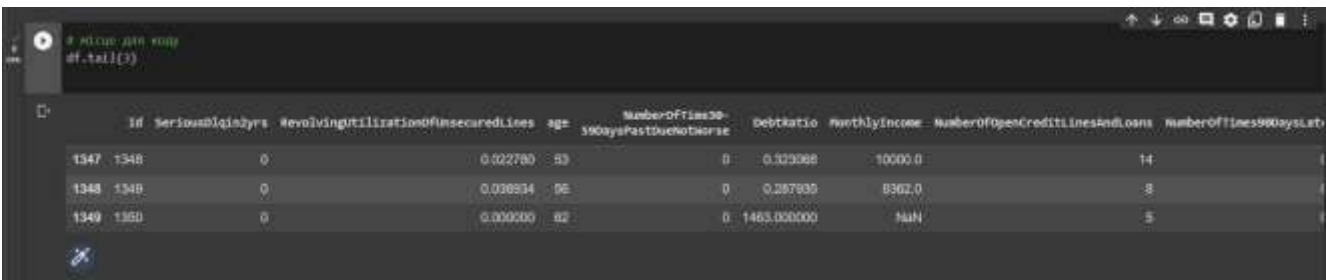


Рис.3.2. Вивід 3 останніх записів.

Дані функції приймають одну int змінну (по замовчуванню 5) - к-ть рядків, що необхідно вивести.

Задача 4.

Прочитайте у файлі DataDictionary-ua.txt, що означають стовпчики матриці. Якому типу належить кожен стовпчик (дійсний, цілий, категоріальний)?

```

# місце для відповіді
import re
uploaded = files.upload()
name = 0
description = 1
name_list = []
desc_list = []
type_list = []
chars_to_remove = "\n"
with open('DataDictionary-ua.txt', 'r') as f:
    for i, line in enumerate(f, 0):
        if i==name:
            name_list.append(line.replace(chars_to_remove, ""))
            name = name + 3
        elif i==description:
            desc_list.append(line.replace(chars_to_remove, ""))
            description = description + 3
#print(name_list)
#print(desc_list)
i = 0;
for line in desc_list:
    result = re.search(r'(.+?)\s+(.+)', line)
    if result:
        column_name, column_description = result.groups()
        if re.search(r'\binteger\b', column_description):
            column_type = 'цілий'
        elif re.search(r'\breal\b', column_description) or re.search(r'%',
column_description):
            column_type = 'дійсний'
        else:
            column_type = 'категоріальний'
        desc_list = column_type
        print(f"{name_list[i]} -> {desc_list}")
        i+=1

```



 Выбрать файлы DataDictionary-ua.txt  
 • **DataDictionary-ua.txt**(text/plain) - 1642 bytes, last modified: 18.03.2023 - 100% done  
 Saving DataDictionary-ua.txt to DataDictionary-ua (31).txt  
 SeriousDlqin2yrs -> категоріальний  
 RevolvingUtilizationOfUnsecuredLines -> дійсний  
 age -> цілий  
 NumberOfTime30-59DaysPastDueNotWorse -> цілий  
 DebtRatio -> дійсний  
 MonthlyIncome -> дійсний  
 NumberOfOpenCreditLinesAndLoans -> цілий  
 NumberOfTimes90DaysLate -> цілий  
 NumberRealEstateLoansOrLines -> цілий  
 NumberOfTime60-89DaysPastDueNotWorse -> цілий  
 NumberOfDependents -> цілий

Рис.4.1. Результат виконання.

		Дяченко В.В.			ДУ «Житомирська політехніка».23.122.08.000 – Лр3	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

### Задача 5.

Цей код вибирає всі рядки, де MonthlyIncome не є NaN за допомогою методу notnull(), створює булеву маску з True та False значеннями та застосовує її до обох стовпців DebtRatio та MonthlyIncome використовуючи метод loc. Значення в стовпці DebtRatio будуть помножені на значення в стовпці MonthlyIncome лише для тих рядків, де MonthlyIncome не є NaN.

```
mask = data['MonthlyIncome'].notnull()
data.loc[mask, 'DebtRatio'] = data.loc[mask, 'DebtRatio'] * data.loc[mask, 'MonthlyIncome']
print(data['DebtRatio'])
```

```
0      7323.197016
1      336.878123
2      258.914887
3      118.963951
4      1584.975094
...
1345    232.944005
1346    1200.699024
1347    3230.676930
1348    2407.712069
1349    1463.000000
Name: DebtRatio, Length: 1350, dtype: float64
```

Рис.5.1. Результат виконання.

### Задача 6.

Змініть ім'я стовпчика на Debt.

```
# міняємо дані
data.rename(columns={'DebtRatio': 'Debt'}, inplace=True)
data.head(1)
```

	Id	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTime30-59DaysPastDueNotWorse	Debt	MonthlyIncome	numberOfOpenCreditLinesAndLoans
0	1	1	0.766127	45	2	7323.197016	6120.0	13

Рис.6.1. Результат заміни.

### Задача 7.

Обчисліть щомісячний дохід та привласніть всім клієнтам з невідомим доходом отримане число.

```
# місце для коду
mean_income = data['MonthlyIncome'].mean()
data.loc[data['MonthlyIncome'].isnull(), 'MonthlyIncome'] = mean_income
data.head(8)
```

	Id	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTime30-59DaysPastDueNotMorae	Debt	MonthlyIncome	NumberOfOpenCreditLinesAndLoans
0	1	1	0.786127	45	2	7323.187016	8120.000000	13
1	2	0	0.857151	40	0	316.878123	2600.000000	4
2	3	0	0.658160	38	1	258.914887	3042.000000	2
3	4	0	0.233810	30	0	118.963851	3300.000000	5
4	5	0	0.907239	49	1	1584.975094	63588.000000	7
5	6	0	0.213179	74	0	1314.624392	3500.000000	3
6	7	0	0.305682	57	0	5710.000000	6438.473482	6
7	8	0	0.754464	39	0	734.780059	3500.000000	1

Рис.7.1.Результат присвоєння.

### Задача 8.

Використовуйте метод `groupby`, оцініть ймовірність неповернення кредиту (`SeriousDlqin2yrs=1`) для різних кількості утриманців (`NumberOfDependents`). Проробіть аналогічну процедуру для різних значень стовпчика `NumberRealEstateLoansOrLines`.

```
[66] # місце для коду
grouped_data = data.groupby('NumberOfDependents')['SeriousDlqin2yrs'].mean()
print(grouped_data)
print('-----')
grouped_data = data.groupby('NumberRealEstateLoansOrLines')['SeriousDlqin2yrs'].mean()
print(grouped_data)
```

NumberOfDependents	
0.0	0.041397
1.0	0.089844
2.0	0.110465
3.0	0.057143
4.0	0.033333
5.0	0.000000
6.0	0.000000
8.0	0.000000

Name: SeriousDlqin2yrs, dtype: float64

```
-----
```

NumberRealEstateLoansOrLines	
0	0.056863
1	0.048729
2	0.063158
3	0.145455
4	0.105263
5	0.000000
6	1.000000
8	0.000000

Name: SeriousDlqin2yrs, dtype: float64

Рис.8.1. Результат обчислення.

		Дяченко В.В.			ДУ «Житомирська політехніка».23.122.08.000 – ПрЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## Задача 9.

### 9а.

Побудуйте графік розсіювання на вісях age та Debt. Синім відмітте клієнтів без серйозних заборгованостей ( $\text{SeriousDlqin2yrs} = 0$ ) та червоним - боржників ( $\text{SeriousDlqin2yrs} = 1$ ).

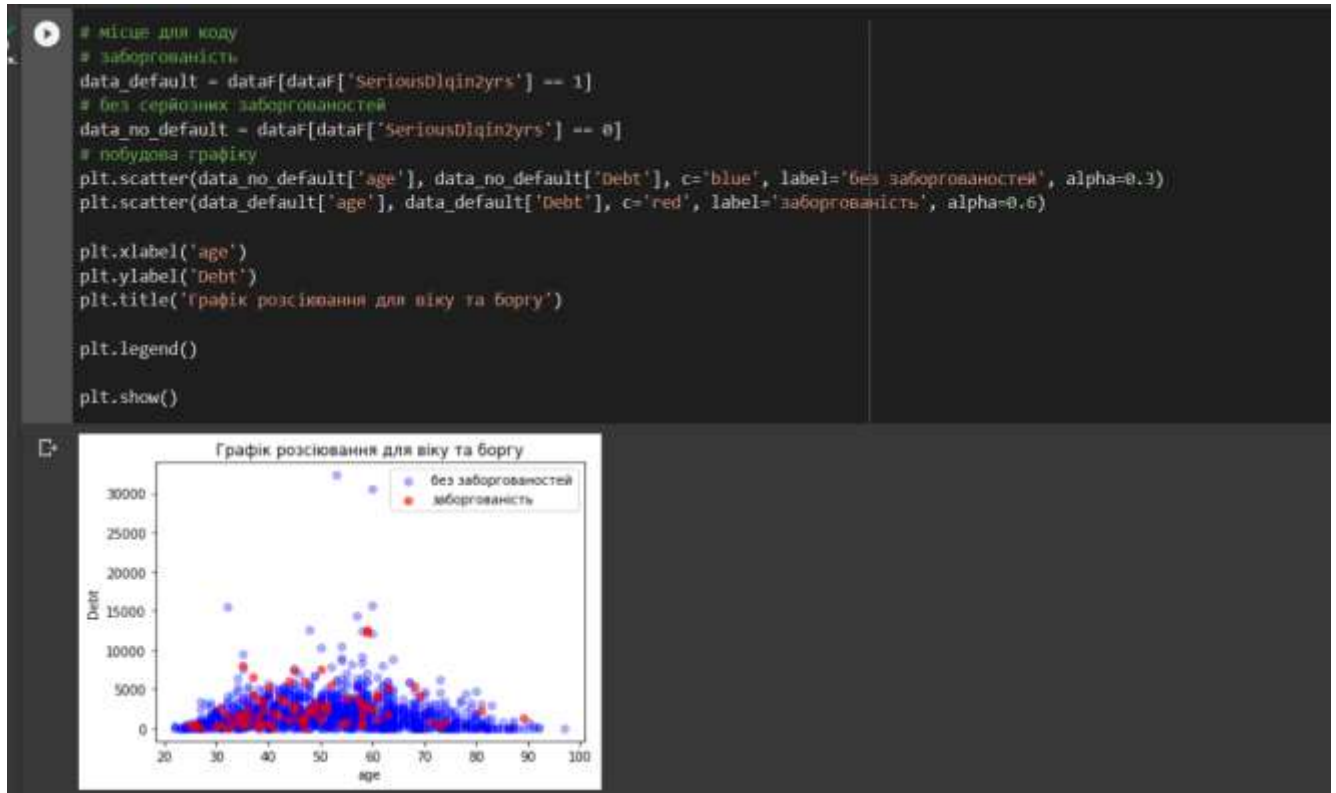


Рис.9.1. Графік розсіювання.

### 9б.

Побудуйте на одньому графіку дві **нормовані** щільності розподілення: червону – для місячного доходу клієнтів з заборгованостями, синю - для місячного доходу клієнтів без заборгованостей. По вісі абсцис відобразіть значення до 25000.

```
import numpy as np
# Вибираємо клієнтів з заборгованістю та без заборгованостей
with_debt = dataF.loc[dataF['SeriousDlqin2yrs'] == 1, 'MonthlyIncome']
without_debt = dataF.loc[dataF['SeriousDlqin2yrs'] == 0, 'MonthlyIncome']

# Обчислюємо параметри нормального розподілу
mean_with_debt, std_with_debt = with_debt.mean(), with_debt.std()
mean_without_debt, std_without_debt = without_debt.mean(), without_debt.std()

# Задаємо діапазон значень для графіка
x = range(25000)
```

		Дяченко В.В.			ДУ «Житомирська політехніка».23.122.08.000 – ПрЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

```
# Рахуємо нормовані щільності
pdf_with_debt = (1/(std_with_debt * np.sqrt(2 * np.pi))) * np.exp(-0.5 * ((x -
    mean_with_debt) / std_with_debt)**2)
pdf_without_debt = (1/(std_without_debt * np.sqrt(2 * np.pi))) * np.exp(-
    0.5 * ((x - mean_without_debt) / std_without_debt)**2)

# Побудова графіку
plt.plot(x, pdf_with_debt, color='red', label='Debt')
plt.plot(x, pdf_without_debt, color='blue', label='No Debt')
plt.xlabel('Monthly Income')
plt.ylabel('Normalized Density')
plt.title('Monthly Income Distribution')
plt.legend()
plt.show()
```

Нормована щільність розподілу - це функція, яка показує ймовірність того, що випадкова величина знаходиться в деякому діапазоні значень. Вона нормована таким чином, щоб інтеграл від цієї функції по всьому простору значень був рівний одиниці.

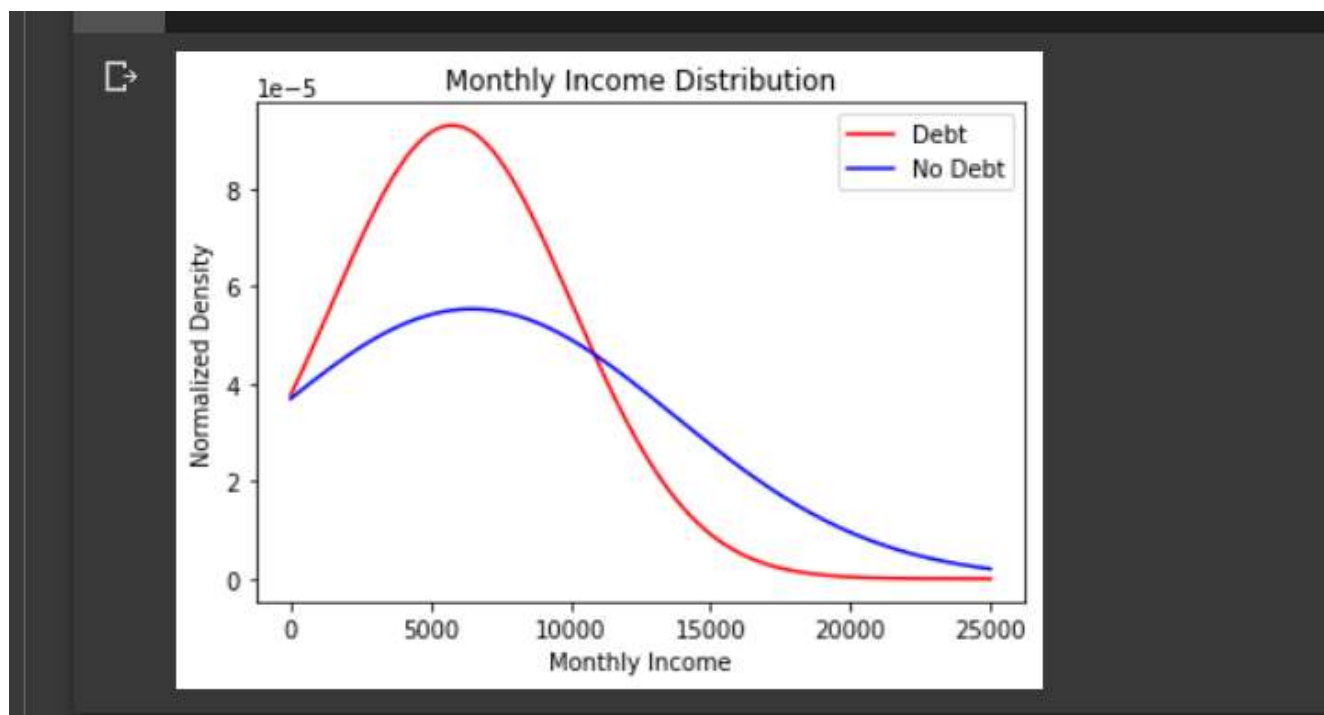


Рис.9.2. Нормовані щільності розподілення.

Нормовані щільності розподілу дозволяють описувати інформацію про розподіл випадкової величини в більш зручному для аналізу вигляді. Одна з основних властивостей нормованих щільностей полягає в тому, що їх інтеграл по

		Дяченко В.В.			ДУ «Житомирська політехніка».23.122.08.000 – Пр3	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		



всьому простору значень випадкової величини дорівнює одиниці. Це означає, що вони надають інформацію про те, яка частина значень випадкової величини припадає на певний діапазон. Нормовані щільності дозволяють здійснювати порівняння різних розподілів випадкових величин, зокрема, визначати ймовірності того, що значення випадкової величини потрапить в певний діапазон значень.

### 9с.

Візуалізуйте попарні залежності між небінарними ознаками 'age', 'MonthlyIncome', 'NumberOfDependents'. Обмежте при цьому місячний дохід значенням 25000. Які закономірності ви можете спостерігати на отриманих графіках?

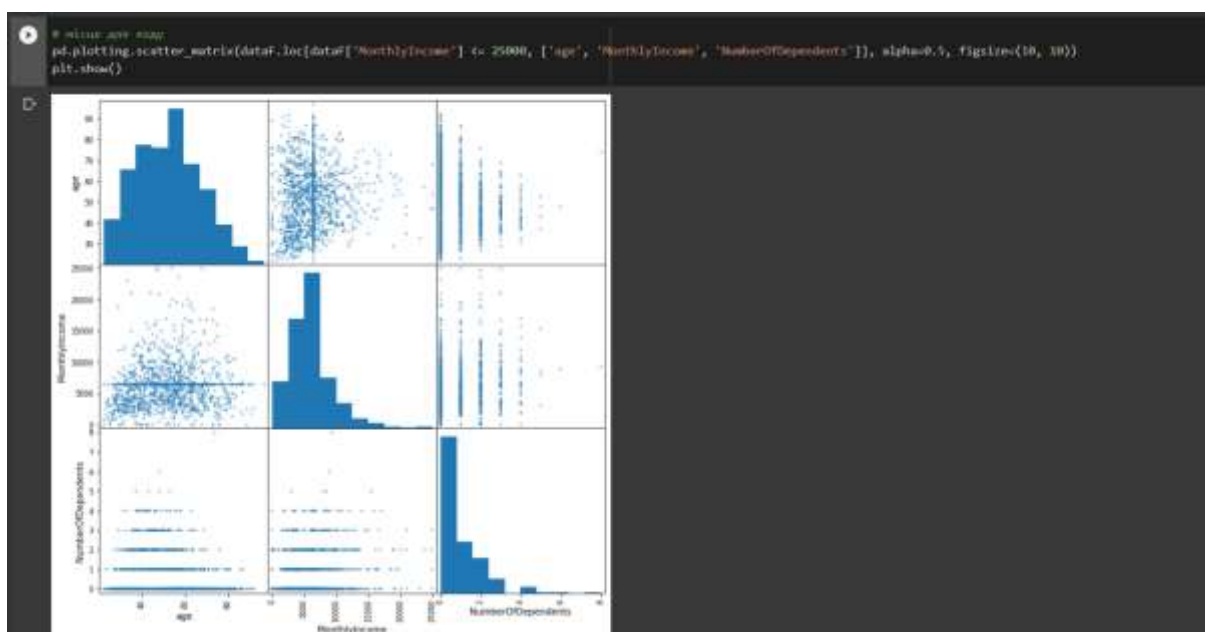


Рис.9.3. Попарні залежності між небінарними ознаками.

Графіки ознак при зміні їх порядку, змінюють своє положення, тобто ми маємо лише 5 різних графіків.

**Висновки:** виконавши дану лабораторну роботу ми вивчили та використали для аналізу даних на практиці бібліотеку Pandas, мова програмування Python.