Корисні посилання

Початок роботи

- Anaconda (дистрибутив для Python, що також вміщує Jupyter Notebook та сам Python): https://www.anaconda.com/download/ (https://www.anaconda.com/download/)
- документація по Jupyter: https://jupyter.readthedocs.io/en/latest/ (https://jupyter.readthedocs.io/en/latest/)
- документація по pip (для встановлення пакетів з Python за допомогою pip install): https://pip.readthedocs.io/en/latest/ (https://pip.readthedocs.io/en/latest/)
- PyCharm: https://www.jetbrains.com/pycharm/ (https://www.jetbrains.com/pycharm/)
- Google Colab: https://colab.research.google.com/ (https://colab.research.google.com/)
- Kaagle: https://www.kaggle.com/ (https://www.kaggle.com/)

Загальні дані

- всі питання задаємо сюди: https://www.google.com/ (https://www.google.com/)
- відповіді на проблемні питання шукаємо тут (знайти можна більшість): https://stackoverflow.com/ (https://stackoverflow.com/)
- A visual introduction to machine learning: http://www.r2d3.us/visual-intro-to-machine-learning-part-1/ (http://www.r2d3.us/visual-intro-to-machine-learning-part-1/)

Google Colab

- Початок роботи з Google Colab: https://colab.research.google.com/notebooks/welcome.ipynb?hl=uk#scrollTo=5fCEDCU_qrC0 (https://colab.research.google.com/notebooks/welcome.ipynb?hl=uk#scrollTo=5fCEDCU_qrC0)
- Наука про дані: https://colab.research.google.com/notebooks/welcome.ipynb?hl=uk#scrollTo=UdRyKR44dcNI (https://colab.research.google.com/notebooks/welcome.ipynb?hl=uk#scrollTo=UdRyKR44dcNI)
- Машинне навчання: https://colab.research.google.com/notebooks/welcome.ipynb?hl=uk#scrollTo=OwuxHmxllTwN (https://colab.research.google.com/notebooks/welcome.ipynb?hl=uk#scrollTo=OwuxHmxllTwN)

Kaagle

- Intro to Machine Learning: https://www.kaggle.com/learn/intro-to-machine-learning (https://www.kaggle.com/learn/intro-to-machine-learning)

pandas

- документація: http://pandas.pydata.org/pandas-docs/stable/ (http://pandas.pydata.org/pandas-docs/stable/)
- 10 minutes to pandas: https://pandas.pydata.org/pandas-docs/stable/10min.html (https://pandas.pydata.org/pandas-docs/stable/10min.html)
- Pandas Tutorial: DataFrames in Python: https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python (https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python)
- Cheet Sheet: https://www.analyticsvidhya.com/blog/2015/07/11-steps-perform-data-analysis-pandas-python/ (https://www.analyticsvidhya.com/blog/2015/07/11-steps-perform-data-analysis-pandas-python/)
- Visualization: http://pandas.pydata.org/pandas-docs/stable/visualization.html (http://pandas.pydata.org/pandas-docs/stable/visualization.html)

sklearn

- документація та багато іншого: http://scikit-learn.org/stable/ (http://scikit-learn.org/stable/)

Інші бібліотеки

- matplotlib: https://matplotlib.org/users/pyplot_tutorial.html (https://matplotlib.org/users/pyplot_tutorial.html)
- seaborn: http://seaborn.pydata.org/ (http://seaborn.pydata.org/)

Лабораторна робота 1: робота з Pandas.

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.

In [3]:

```
import pandas as pd
```

Основними структурними даними в Pandas є класи Series та DataFrame. Перший з ним являє собою одновимірний масив даних деякого фіксованоо типу. Другий - це двовимірна структура даних, що являє мобою таблицю, кажен стовпчик якої вміщує дані одного типу. Можно представлять её как словарь объектов типа Series.

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.

1

Прочитайте дані з файлу data.csv

Функції, що можуть бути корисними при вирішенні: `pd.read_csv(..., delimiter=',')`

In [4]:

```
df = pd.read_csv("data.csv")
```

2

Виведіть опис даних, що було прочитано.

Функції, що можуть бути корисними при вирішенні: `.describe()`

In [19]:

```
df.describe()
```

Out[19]:

	Id	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	59D
count	1350.000000	1350.000000	1350.000000	1350.000000	
mean	675.500000	0.060000	3.577895	52.048889	
std	389.855743	0.237575	84.914699	15.009875	
min	1.000000	0.000000	0.000000	22.000000	
25%	338.250000	0.000000	0.031140	40.000000	
50%	675.500000	0.000000	0.156891	52.000000	
75%	1012.750000	0.000000	0.543145	63.000000	
max	1350.000000	1.000000	2340.000000	97.000000	

3

Відобразіть декілька перших та декілька останніх записів.

Функції, що можуть бути корисними при вирішенні: `.head(), .tail()`

Які параметри можна передати даним функціям?

In [6]:

```
df.head() #Shows the beginning of the file

df.tail() #Shows the beginning of the file
```

Out[6]:

	Id	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTin 59DaysPastDueNotW
1345	1346	0	0.000000	39	
1346	1347	0	0.045694	49	
1347	1348	0	0.022780	53	
1348	1349	0	0.036934	56	
1349	1350	0	0.000000	62	

4

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

In [7]:

```
pd.read_csv('DataDictionary-ua.txt', delimiter='\t')
```

Out[7]:

	SeriousDlqin2yrs
0	Чи відбудеться відтермінування платежу на 90 а...
1	RevolvingUtilizationOfUnsecuredLines
2	Процент грошей, що залишились на всіх кредитни...
3	age
4	Вік (integer)
5	NumberOfTime30-59DaysPastDueNotWorse
6	Кількість короткотривалих прострочек платежу (...
7	DebtRatio
8	Відсотки від заборгованості за місяць, алімент...
9	MonthlyIncome
10	Дохід за місяць (real)
11	NumberOfOpenCreditLinesAndLoans
12	Кількість кредитів (або кредитних карт) за вик...
13	NumberOfTimes90DaysLate
14	Кредитна історія: кількість серьйохних простро...
15	NumberRealEstateLoansOrLines
16	Кількість іпотек (integer)
17	NumberOfTime60-89DaysPastDueNotWorse
18	Кількість середньотермінових прострочек платеж...
19	NumberOfDependents
20	Кількість утриманців в сім'ї, включаюси самого...

5

Зверніть увагу, що стовпчик `DebtRatio` вміщує неправдоподібні дані. Тількм значенея, що відповідають відомому доходу за місяць, є відношеннями. Всі інші - абсолютні значення виплат відсотків за місяць.

Виправте дані, зробивши всі значення стовпчика `DebtRatio` абсолютними (помножте їх на `MonthlyIncome`). Щоб ваша програма працювала швидко на повних даних, спробуйте не використовувати цикл.

Функції, що можуть бути корисними при рішенні:

Звертання до елементів DataFrame:

- елемент: `data.loc[i, 'назваСтовпчика']`
- стовпчик: `data['назваСтовпчика']`

- підматрица: `data.loc[a:b, списокНазвСтовпчиків]`

Умовна індексація:

- `data.loc[data['стовпчик'] > 20, списокНазвСтовпчиків]`

краще писати так:

- `i = data['стовпчик'] > 20 # вектор True та False`
- `data.loc[i, 'назваСтовпчик']`

В підматрицях номера рядків наслідуються від початкової.

- `pandas.isnull(скаляр иабо масив)` - перевірка чи є значення невизначеним (`NaN`)
- `pandas.notnull(скаляр або місив)` - перевірка чи є занчення визначеним (не `NaN`)

In [8]:

```
df = pd.read_csv("data.csv")
tf_vec = pd.isnull(df['MonthlyIncome'])
df.loc[tf_vec, 'DebtRatio'] = df.loc[tf_vec, 'DebtRatio']*df.loc[tf_vec, 'MonthlyIncome']
df.tail()
```

Out[8]:

	Id	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTim 59DaysPastDueNotW
1345	1346	0	0.000000	39	
1346	1347	0	0.045694	49	
1347	1348	0	0.022780	53	
1348	1349	0	0.036934	56	
1349	1350	0	0.000000	62	

6

Змініть ім'я стовпчика на `Debt` .

Функції, що можуть бути корисними при вирішенні: `.rename(columns={'староеИмя':'новоеИмя'}, inplace=True)`

In [9]:

```
df.rename(columns={'DebtRatio':'Debt'}, inplace=True)
df.tail()
```

Out[9]:

	Id	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTin 59DaysPastDueNotW
1345	1346	0	0.000000	39	
1346	1347	0	0.045694	49	
1347	1348	0	0.022780	53	
1348	1349	0	0.036934	56	
1349	1350	0	0.000000	62	

7

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

Функції, що можуть бути корисними при вирішенні: `.mean()`

Інші описові статистики: https://pandas.pydata.org/pandas-docs/stable/reference/frame.html#computations-descriptive-stats (https://pandas.pydata.org/pandas-docs/stable/reference/frame.html#computations-descriptive-stats)

In [10]:

```
df_mean = df['MonthlyIncome'].mean()
tf_vec = pd.isnull(df['MonthlyIncome'])
df.loc[tf_vec, 'MonthlyIncome'] = df_mean
df.tail()
```

Out[10]:

	Id	SeriousDlqin2yrs	RevolvingUtilizationOfUnsecuredLines	age	NumberOfTin 59DaysPastDueNotW
1345	1346	0	0.000000	39	
1346	1347	0	0.045694	49	
1347	1348	0	0.022780	53	
1348	1349	0	0.036934	56	
1349	1350	0	0.000000	62	

8

Використовуйте метод `groupby` , оцініть ймовірність неповернення кредиту (`SeriousDlqin2yrs=1`) для різних кількості утриманців (`NumberOfDependents`).

Проробіть аналогічну процедуру для різних значень стовпчика `NumberRealEstateLoansOrLines`

Підказка: `data['стовпчик1'].groupby(data['стовпчик2']).mean()` *-- розрахунок середніх значень стовпчика1 по групам зі стовпчика2*

In [11]:

```
first_groupby = df['SeriousDlqin2yrs'].groupby(df['NumberOfDependents']).mean()
second_groupby = df['SeriousDlqin2yrs'].groupby(df['NumberRealEstateLoansOrLines']).mean(
second_groupby
```

Out[11]:

```
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
```

Візуалізація даних

In [12]:

```
import matplotlib.pyplot as plt

#  функція, що дозволяє виводити графіки прямо до ноутбуку (колабу)
%matplotlib inline
```

Matplotlib дозволяє зручно візуалізувати табличні дані.

Функції, що можуть бути корисними при вирішенні:

- Рисування:
  - `plt.plot(x, y)` див.детальніше http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot (http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)
  - `plt.show()`
  - `plt.scatter(x, y)` - графік розсіювання, див. http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.scatter (http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.scatter)
  - `plt.hist()` - гістограма, див. http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.hist (http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.hist)
- Рисування декількох графіків на одному:

  `fig, ax = plt.subplots()  ax.hist(...)  ax.hist(...)  plt.show()`
- Логарифмічна шкала:
  - `ax.set_xscale('log')` или `ax.set_yscale('log')`
- Обмеження області графіку:
  - `ax.axis([x1, x2, y1, y2])`

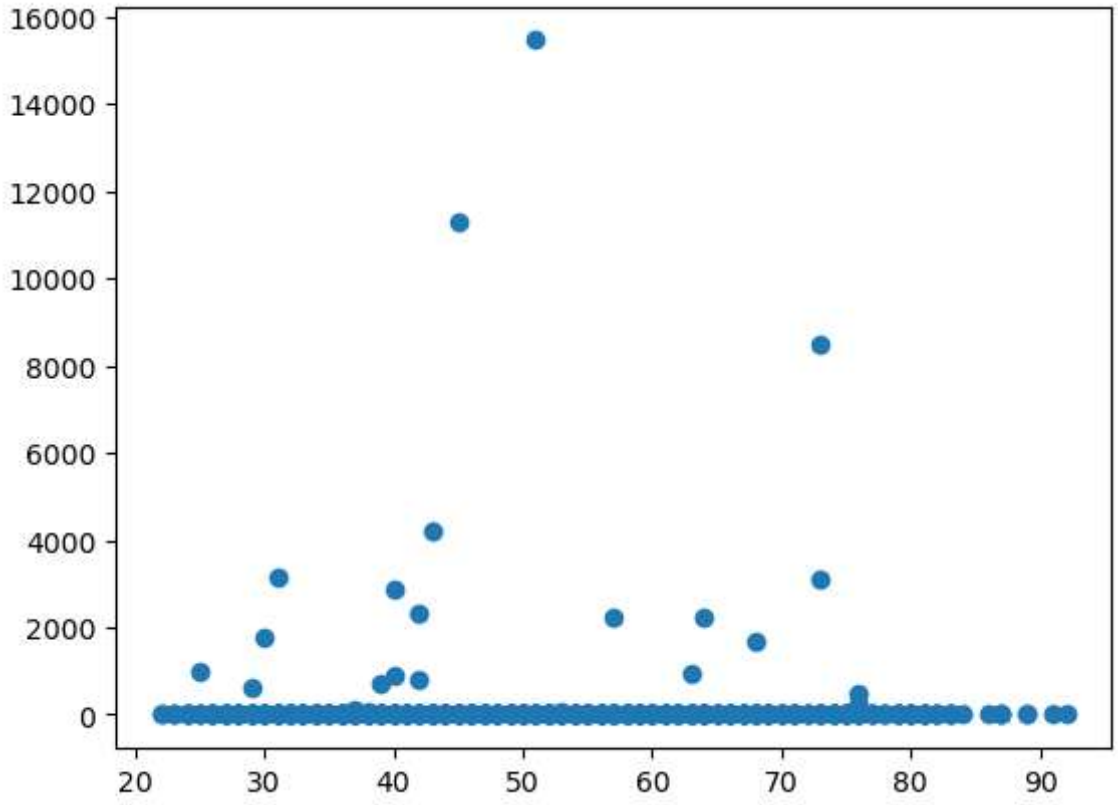
9a

In [13]:

```
plt.scatter(df['age'], df['Debt'])
```

Out[13]:

```
<matplotlib.collections.PathCollection at 0x1dd9b20d3f0>
```

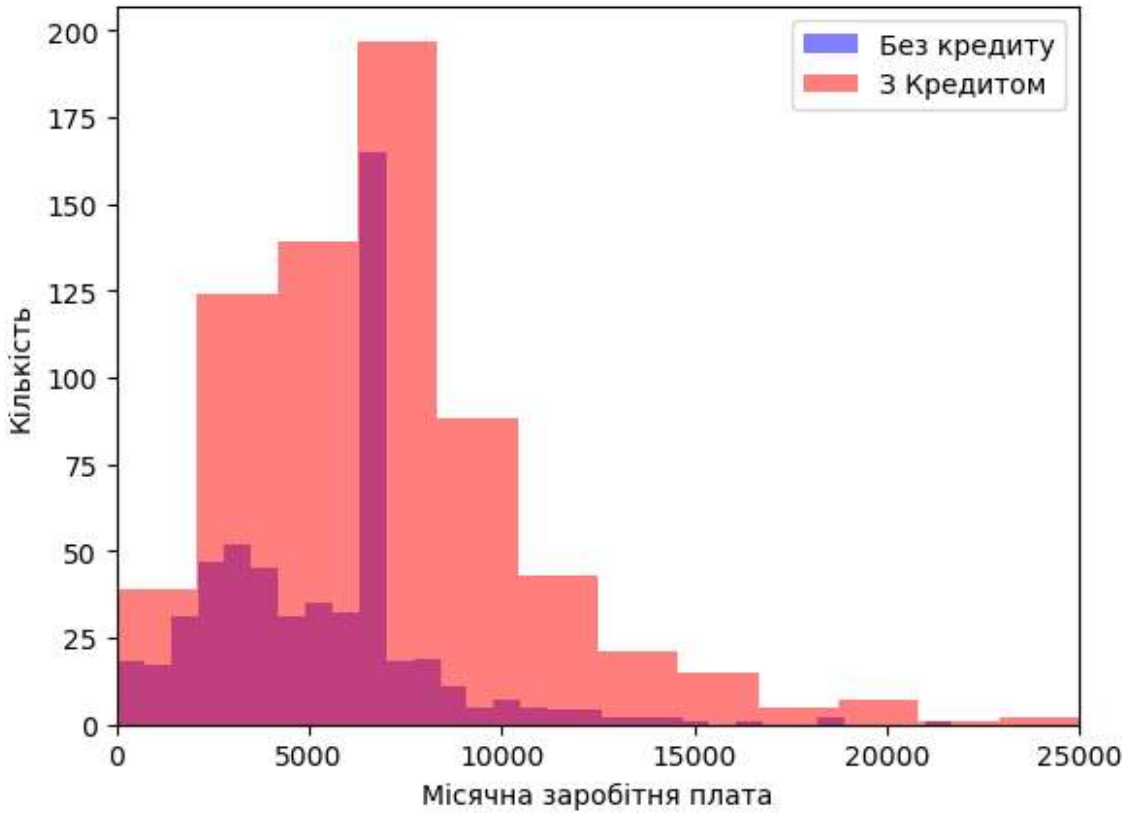


9b

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

In [14]:

```
tf_with_debt = df.loc[df.NumberOfOpenCreditLinesAndLoans > 7, 'MonthlyIncome']
tf_without_debt = df.loc[df.NumberOfOpenCreditLinesAndLoans < 7, 'MonthlyIncome']

tf_with_debt

plt.hist(tf_without_debt, bins=100, color='blue', alpha=0.5, label='Без кредиту')
plt.hist(tf_with_debt, bins=100, color='red', alpha=0.5, label='З Кредитом')


plt.xlim(0, 25000)
plt.xlabel('Місячна заробітня плата')
plt.ylabel('Кількість')

plt.legend()

plt.show()
df.describe()["NumberOfOpenCreditLinesAndLoans"]
```



Out[14]:

```
count    1350.000000
mean        8.434074
std         5.129287
min         0.000000
25%         5.000000
50%         8.000000
75%        11.000000
max        31.000000
Name: NumberOfOpenCreditLinesAndLoans, dtype: float64
```

9c*

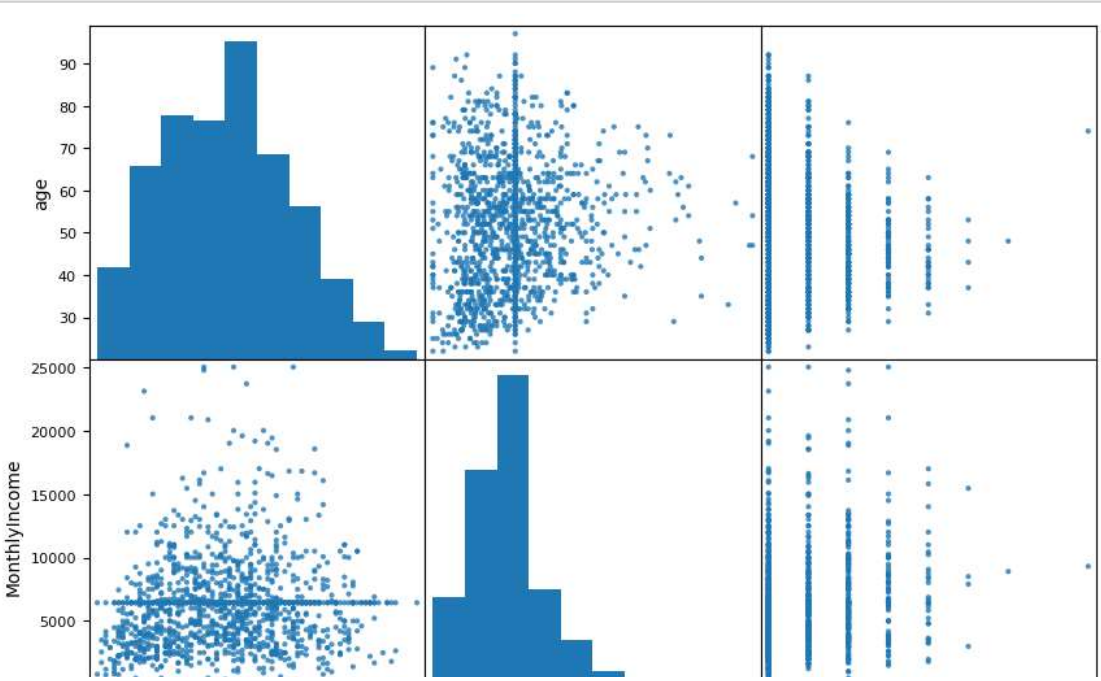
Візуалізуйте попарні залежності між небінарними ознаками попарные зависимости между небинарными признаками `'age', 'MonthlyIncome', 'NumberOfDependents'` . Обмежте при цьому місячний дохід значенням 25000.

Які закономірності ви можете спостерігати на отриманих графіках?

Функції, що можуть бути корисними при вирішенні: `pd.plotting.scatter_matrix()`

In [22]:

```
filtered_data = df[df['MonthlyIncome'] <=25000]
features = ['age', 'MonthlyIncome', 'NumberOfDependents']

pd.plotting.scatter_matrix(filtered_data[features], alpha=0.8, figsize=(10, 10), diagonal

plt.show()
```



In [ ]: