МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

Лабораторная работа 2

по дисциплине

«Методы машинного обучения»

по теме

«Изучение библиотек обработки данных»

ИСПОЛНИТЕЛЬ: Чечнев А.А.

ФИО

группа ИУ5-23М

____

подпись

"__"_________2020 г.

ПРЕПОДАВАТЕЛЬ: Гапанюк Ю. Е.

ФИО

____

подпись

"__"_________2020 г.

Москва – 2020

Лабораторная работа 2

Изучение библиотек обработки данных.

Цель лабораторной работы: изучение библиотек обработки данных Pandas и PandaSQL.

Требования к отчету:

Отчет по лабораторной работе должен содержать:

- титульный лист;
- описание задания;
- текст программы;
- экранные формы с примерами выполнения программы.
- Задание:

Часть 1.

Выполните первое демонстрационное задание "demo assignment" под названием "Exploratory data analysis with Pandas" со страницы курса https://mlcourse.ai/assignments (https://mlcourse.ai/assignments)

Условие задания - https://nbviewer.jupyter.org/github/Yorko/mlcourse_open/blob/master/jupyter_english/assignments_demo/assignm flush_cache=true (https://nbviewer.jupyter.org/github/Yorko/mlcourse_open/blob/master/jupyter_english/assignments_demo/assignm flush_cache=true)

Официальный датасет находится здесь, но данные и заголовки хранятся отдельно, что неудобно для анализа - https://archive.ics.uci.edu/ml/datasets/Adult (https://archive.ics.uci.edu/ml/datasets/Adult)

Поэтому готовый набор данных для лабораторной работы удобнее скачать здесь - https://raw.githubusercontent.com/Yorko/mlcourse.ai/master/data/adult.data.csv (https://raw.githubusercontent.com/Yorko/mlcourse.ai/master/data/adult.data.csv) (удобнее всего нажать на данной ссылке правую кнопку мыши и выбрать в контекстном меню пункт "сохранить ссылку", будет предложено сохранить файл в формате CSV)

Пример решения задания - https://www.kaggle.com/kashnitsky/a1-demo-pandas-and-uci-adult-dataset-solution (https://www.kaggle.com/kashnitsky/a1-demo-pandas-and-uci-adult-dataset-solution)

Часть 2.

Выполните следующие запросы с использованием двух различных библиотек - Pandas и PandaSQL:

- один произвольный запрос на соединение двух наборов данных
- один произвольный запрос на группировку набора данных с использованием функций агрегирования
- Сравните время выполнения каждого запроса в Pandas и PandaSQL.

В качестве примеров можно использовать следующие статьи:

https://www.shanelynn.ie/summarising-aggregation-and-grouping-data-in-python-pandas/ (https://www.shanelynn.ie/summarising-aggregation-and-grouping-data-in-python-pandas/)
https://www.shanelynn.ie/merge-join-dataframes-python-pandas-index-1/ (https://www.shanelynn.ie/merge-join-dataframes-python-pandas-index-1/) (в разделе "Example data" данной статьи содержится рекомендуемый

набор данных для проведения экспериментов). Пример сравнения Pandas и PandaSQL - https://github.com/miptgirl/udacity_engagement_analysis/blob/master/pandasql_example.ipynb (https://github.com/miptgirl/udacity_engagement_analysis/blob/master/pandasql_example.ipynb)

mlcourse.ai (https://mlcourse.ai) - Open Machine Learning Course

Author: Yury Kashnitsky (https://www.linkedin.com/in/festline/). Translated and edited by Sergey Isaev (https://www.linkedin.com/in/isvforall/), Artem Trunov (https://www.linkedin.com/in/datamove/), Anastasia Manokhina (https://www.linkedin.com/in/anastasiamanokhina/), and Yuanyuan Pao (https://www.linkedin.com/in/yuanyuanpao/).

Assignment #1 (demo)

Exploratory data analysis with Pandas

Same assignment as a Kaggle Kernel (https://www.kaggle.com/kashnitsky/a1-demo-pandas-and-uci-adult-dataset) + solution (https://www.kaggle.com/kashnitsky/a1-demo-pandas-and-uci-adult-dataset-solution).

In this task you should use Pandas to answer a few questions about the Adult (https://archive.ics.uci.edu/ml/datasets/Adult) dataset. (You don't have to download the data – it's already in the repository). Choose the answers in the web-form (https://docs.google.com/forms/d/1uY7MpI2trKx6FLWZte0uVh3ULV4Cm_tDud0VDFGCOKg).

Unique values of all features (for more information, please see the links above):

- `age` : continuous.
- `workclass` : Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- `fnlwgt` : continuous.
- `education` : Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- `education-num` : continuous.
- `marital-status` : Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- `occupation` : Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- `relationship` : Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- `race` : White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- `sex` : Female, Male.
- `capital-gain` : continuous.

- `capital-loss` : continuous.
- `hours-per-week` : continuous.
- `native-country` : United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
- `salary` : >50K,<=50K

In [1]:

```
import numpy as np
import pandas as pd
pd.set_option('display.max.columns', 100)
# to draw pictures in jupyter notebook
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df = pd.read_csv('data/adult.data', sep=', ')
df.head()
```

```
C:\Users\als\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: ParserWarn
ing: Falling back to the 'python' engine because the 'c' engine does not sup
port regex separators (separators > 1 char and different from '\s+' are inte
rpreted as regex); you can avoid this warning by specifying engine='python'.
  """Entry point for launching an IPython kernel.
```

Out[2]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Fe

1. How many men and women (*sex* feature) are represented in this dataset?

In [3]:

```
df.sex.value_counts()
```

Out[3]:

```
Male      21790
Female    10771
Name: sex, dtype: int64
```

2. What is the average age (*age* feature) of women?

In [4]:

```
df[df.sex == 'Female'].age.mean()
```

Out[4]:

```
36.85823043357163
```

3. What is the percentage of German citizens (*native-country* feature)?

In [5]:

```
df['native-country'].value_counts(normalize=True)['Germany']*100
```

Out[5]:

```
0.42074874850281013
```

4-5. What are the mean and standard deviation of age for those who earn more than 50K per year (salary feature) and those who earn less than 50K per year? **

In [6]:

```
df.salary.value_counts()
```

Out[6]:

```
<=50K    24720
>50K      7841
Name: salary, dtype: int64
```

In [7]:

```
df.groupby(by='salary').agg({'age':['mean','std']})
```

Out[7]:

	age	
	mean	std
salary		
<=50K	36.783738	14.020088
>50K	44.249841	10.519028

6. Is it true that people who earn more than 50K have at least high school education? (*education – Bachelors, Prof-school, Assoc-acdm, Assoc-voc, Masters* or *Doctorate* feature)

Bachelors, Prof-school, Assoc-acdm, Assoc-voc, Masters or Doctorate feature)

In [8]:

```
df[df.salary=='>50K'].education.value_counts()
```

Out[8]:

```
Bachelors       2221
HS-grad         1675
Some-college    1387
Masters          959
Prof-school      423
Assoc-voc        361
Doctorate        306
Assoc-acdm       265
10th              62
11th              60
7th-8th           40
12th              33
9th               27
5th-6th           16
1st-4th            6
Name: education, dtype: int64
```

No

7. Display age statistics for each race (*race* feature) and each gender (*sex* feature). Use *groupby()* and *describe()*. Find the maximum age of men of *Amer-Indian-Eskimo* race.

In [9]:

```
df.groupby(by=['race', 'sex']).age.describe()
```

Out[9]:

race	sex	count	mean	std	min	25%	50%	75%	max
Amer-Indian-Eskimo	**Female**	119.0	37.117647	13.114991	17.0	27.0	36.0	46.00	80.0
	Male	192.0	37.208333	12.049563	17.0	28.0	35.0	45.00	82.0
Asian-Pac-Islander	**Female**	346.0	35.089595	12.300845	17.0	25.0	33.0	43.75	75.0
	Male	693.0	39.073593	12.883944	18.0	29.0	37.0	46.00	90.0
Black	**Female**	1555.0	37.854019	12.637197	17.0	28.0	37.0	46.00	90.0
	Male	1569.0	37.682600	12.882612	17.0	27.0	36.0	46.00	90.0
Other	**Female**	109.0	31.678899	11.631599	17.0	23.0	29.0	39.00	74.0
	Male	162.0	34.654321	11.355531	17.0	26.0	32.0	42.00	77.0
White	**Female**	8642.0	36.811618	14.329093	17.0	25.0	35.0	46.00	90.0
	Male	19174.0	39.652498	13.436029	17.0	29.0	38.0	49.00	90.0

In [10]:

```
df1 = df.groupby(by=['race', 'sex']).age.describe()
df1.loc['Amer-Indian-Eskimo', 'Male']['max']
```

Out[10]:

```
82.0
```

8. Among whom is the proportion of those who earn a lot (>50K) greater: married or single men (*marital-status* feature)? Consider as married those who have a *marital-status* starting with *Married* (Married-civ-spouse, Married-spouse-absent or Married-AF-spouse), the rest are considered bachelors.

In [11]:

```
df[df.salary=='>50K'].groupby(by='marital-status').age.count()
```

Out[11]:

```
marital-status
Divorced                  463
Married-AF-spouse          10
Married-civ-spouse       6692
Married-spouse-absent      34
Never-married             491
Separated                  66
Widowed                    85
Name: age, dtype: int64
```

answer = among married

9. What is the maximum number of hours a person works per week (*hours-per-week* feature)? How many people work such a number of hours, and what is the percentage of those who earn a lot (>50K) among them?

In [12]:

```
#df.sort_values(by='hours-per-week', ascending=False)
mx = df['hours-per-week'].max()
mx
```

Out[12]:

```
99
```

In [13]:

```
df[df['hours-per-week'] == mx].count()
```

Out[13]:

```
age               85
workclass         85
fnlwgt            85
education         85
education-num     85
marital-status    85
occupation        85
relationship      85
race              85
sex               85
capital-gain      85
capital-loss      85
hours-per-week    85
native-country    85
salary            85
dtype: int64
```

In [14]:

```
df[df['hours-per-week'] == mx].salary.value_counts(normalize=True)
```

Out[14]:

```
<=50K    0.705882
>50K     0.294118
Name: salary, dtype: float64
```

answer = 0.705882

10. Count the average time of work (*hours-per-week*) for those who earn a little and a lot (*salary*) for each country (*native-country*). What will these be for Japan?

In [15]:

```
# You code here
```

In [16]:

```
df.columns
```

Out[16]:

```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education-num',
       'marital-status', 'occupation', 'relationship', 'race', 'sex',
       'capital-gain', 'capital-loss', 'hours-per-week', 'native-country',
       'salary'],
      dtype='object')
```

In [17]:

```
df_hpw = df.groupby(by=['native-country', 'salary']).agg({'hours-per-week':'mean'})
```

In [18]:

```
df_hpw.loc['Japan']
```

Out[18]:

	hours-per-week
salary	
<=50K	41.000000
>50K	47.958333

In [19]:

```
df1 = df.iloc[0:4]
df2 = df.iloc[50:53]
```

Получим из таблицы с исходными данными топ3 людей, чей возраст меньше 40

In [20]:

```
import pandasql as ps
import pandas as pd
```

In [21]:

```
simple_query = '''
    SELECT
        age,
        workclass,
        fnlwgt,
        education
    FROM df
    WHERE age < 40
    ORDER BY age desc
    LIMIT 3

    '''
%time df_ps = ps.sqldf(simple_query, locals())
df_ps
```

```
Wall time: 655 ms
```

Out[21]:

	age	workclass	fnlwgt	education
0	39	State-gov	77516	Bachelors
1	39	Private	367260	HS-grad
2	39	Private	365739	Some-college

In [22]:

```
columns = ['age', 'workclass', 'fnlwgt', 'education']
%time df_pd = df.loc[df.age < 40, columns].sort_values(by='age', ascending=False).head(3)
df_pd
```

Wall time: 16.5 ms

Out[22]:

	age	workclass	fnlwgt	education
0	39	State-gov	77516	Bachelors
12603	39	Private	185053	HS-grad
1608	39	Private	379350	10th

In [ ]:

In [55]:

```
def example2_pandasql(data):
    aggr_query = '''
        SELECT
            count(age) as count,
            avg(age)  as mean,
            min(age)   as mean
        FROM data
        GROUP BY race
        '''
    return ps.sqldf(aggr_query, locals()).set_index('age')
```

In [24]:

```
df.groupby(by=['race', 'sex']).age.describe()
```

Out[24]:

race	sex	count	mean	std	min	25%	50%	75%	max
Amer-Indian-Eskimo	**Female**	119.0	37.117647	13.114991	17.0	27.0	36.0	46.00	80.0
	Male	192.0	37.208333	12.049563	17.0	28.0	35.0	45.00	82.0
Asian-Pac-Islander	**Female**	346.0	35.089595	12.300845	17.0	25.0	33.0	43.75	75.0
	Male	693.0	39.073593	12.883944	18.0	29.0	37.0	46.00	90.0
Black	**Female**	1555.0	37.854019	12.637197	17.0	28.0	37.0	46.00	90.0
	Male	1569.0	37.682600	12.882612	17.0	27.0	36.0	46.00	90.0
Other	**Female**	109.0	31.678899	11.631599	17.0	23.0	29.0	39.00	74.0
	Male	162.0	34.654321	11.355531	17.0	26.0	32.0	42.00	77.0
White	**Female**	8642.0	36.811618	14.329093	17.0	25.0	35.0	46.00	90.0
	Male	19174.0	39.652498	13.436029	17.0	29.0	38.0	49.00	90.0

In [23]:

```
%time pd.concat([df1, df2])
```

```
Wall time: 4.99 ms
```

Out[23]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	
50	25	Private	32275	Some-college	10	Married-civ-spouse	Exec-managerial	Wife	Other	F
51	18	Private	226956	HS-grad	9	Never-married	Other-service	Own-child	White	F
52	47	Private	51835	Prof-school	15	Married-civ-spouse	Prof-specialty	Wife	White	F