Московский Государственный Технический Университет
им. Н.Э. Баумана

Отчет по лабораторной работе №2
по курсу
Технологии Машинного Обучения

Выполнила:
Костян Алина
ИУ5-53

_____

Проверил:
Гапанюк Ю.Е.

_____

Москва, 2019

Разведочный анализ данных с Pandas

Exploratory data analysis with Pandas

Уникальные значения всех фич:

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)

%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
data = pd.read_csv('adult.data.txt')
data.head()
```

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

1. Как много мужчин и женщин представлено в этом наборе данных?

In [3]:

```
data['sex'].value_counts()
```

Out[3]:

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

2. Какой средний возраст женщин?

In [4]:

```
data.loc[data['sex'] == ' Female', 'age'].mean()
```

Out[4]:

```
36.85823043357163
```

3. Какой процент жителей Германии?

In [5]:

```
(float((data['native-country'] == ' Germany').sum()) / data.shape[0])*100
```

Out[5]:

```
0.42074874850281013
```

4,5. Среднее значение и стандартное отклонение в возрасте для тех, кто зарабатывает больше 50. тыс в год и тех, кто получает меньше 50 тысяч в год?

Зарплата больше 50 тысяч в год

Среднее значение

In [6]:

```
data.loc[data['salary'] == ' >50K', 'age'].mean()
```

Out[6]:

```
44.24984058155847
```

Стандартное отклонение

In [7]:

```
data.loc[data['salary'] == ' >50K', 'age'].std()
```

Out[7]:

```
10.519027719851826
```

Зарплата 50 тысяч и меньше

Среднее значение

In [8]:

```
data.loc[data['salary'] == ' <=50K', 'age'].mean()
```

Out[8]:

```
36.78373786407767
```

Стандартное отклонение

In [9]:

```
data.loc[data['salary'] == ' <=50K', 'age'].std()
```

Out[9]:

```
14.02008849082488
```

6. Правда ли что люди которые получают больше 50 тысяч имеют хотя бы школьное образование?

In [10]:

```
HighSC = {' Bachelors', ' Prof-school', ' Assoc-acdm', ' Assoc-voc', ' Masters',
' Doctorate'}
```

In [11]:

```
for i in data.loc[data['salary'] == ' <=50K', 'education'].unique():
    if i not in HighSC:
        print('Не правда')
        break
```

Не правда

7. Отобразите статистику вораста для каждой расы и каждого пола. Используйте groupby() и describe(). Найдите максимальный возраст мужчин Американской-инди-эскимосской расы.

In [12]:

```
for (race, sex), sub in data.groupby(['race', 'sex']):
    print(f"Race: {race}, sex: {sex}")
    print(sub['age'].describe())
```

```
Race:  Amer-Indian-Eskimo, sex:  Female
count    119.000000
mean      37.117647
std       13.114991
min       17.000000
25%       27.000000
50%       36.000000
75%       46.000000
max       80.000000
Name: age, dtype: float64
Race:  Amer-Indian-Eskimo, sex:  Male
count    192.000000
mean      37.208333
std       12.049563
min       17.000000
25%       28.000000
50%       35.000000
75%       45.000000
max       82.000000
Name: age, dtype: float64
Race:  Asian-Pac-Islander, sex:  Female
count    346.000000
mean      35.089595
std       12.300845
min       17.000000
25%       25.000000
50%       33.000000
75%       43.750000
max       75.000000
Name: age, dtype: float64
Race:  Asian-Pac-Islander, sex:  Male
count    693.000000
mean      39.073593
std       12.883944
min       18.000000
25%       29.000000
50%       37.000000
75%       46.000000
max       90.000000
Name: age, dtype: float64
Race:  Black, sex:  Female
count    1555.000000
mean       37.854019
std        12.637197
min        17.000000
25%        28.000000
50%        37.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Race:  Black, sex:  Male
count    1569.000000
mean       37.682600
std        12.882612
min        17.000000
25%        27.000000
50%        36.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Race:  Other, sex:  Female
```

```
count    109.000000
mean      31.678899
std       11.631599
min       17.000000
25%       23.000000
50%       29.000000
75%       39.000000
max       74.000000
Name: age, dtype: float64
Race:  Other, sex:  Male
count    162.000000
mean      34.654321
std       11.355531
min       17.000000
25%       26.000000
50%       32.000000
75%       42.000000
max       77.000000
Name: age, dtype: float64
Race:  White, sex:  Female
count    8642.000000
mean       36.811618
std        14.329093
min        17.000000
25%        25.000000
50%        35.000000
75%        46.000000
max        90.000000
Name: age, dtype: float64
Race:  White, sex:  Male
count    19174.000000
mean        39.652498
std         13.436029
min         17.000000
25%         29.000000
50%         38.000000
75%         49.000000
max         90.000000
Name: age, dtype: float64
```

Определим самый большой возраст среди мужчин расы АмерканскоИндийскихЭскимо

In [13]:

```
cake=data.loc[data['race'] == ' Amer-Indian-Eskimo']
cake.loc[cake['sex'] == ' Male', 'age'].max()
```

Out[13]:

```
82
```

8. Доля каких мужчин больше среди тех, кто зарабатывает больше 50 тысяч, женатых или холостяков?

In [14]:

```
not_married_men = data.loc[(data['sex'] == ' Male') &
     (data['marital-status'].isin([' Never-married',
                                   ' Separated',
                                   ' Divorced',
                                   ' Widowed']))]

married_men = data.loc[(data['sex'] == ' Male') &
     (data['marital-status'].isin([' Married-civ-spouse',
                                   ' Married-spouse-absent',
                                   ' Married-AF-spouse']))]
print (f"Доля неженатых мужчин {(not_married_men['salary'] == ' >50K').sum()}")
print (f"Доля женатых мужчин {(married_men['salary'] == ' >50K').sum()}\n")

if ((not_married_men['salary'] == ' >50K').sum() > (married_men['salary'] == ' >
50K').sum()):
       print('Доля неженатых мужчин больше')
elif ((married_men['salary'] == ' >50K').sum() > (not_married_men['salary'] == '
>50K').sum()):
       print('Доля женатых мужчин больше')
else:
       print('Доли женатых и неженатых мужчин равны')
```

```
Доля неженатых мужчин 697
Доля женатых мужчин 5965

Доля женатых мужчин больше
```

9. Какое максимальное количество часов человек работает в неделю? Как много людей работают столько часов и каков процент тех кто зарабатывает больше 50 тысяч среди них?

In [15]:

```
maxxi = (data['hours-per-week']).max()
print (f"Максимальное количество часов в неделю: {maxxi}")

coun = data.loc[data['hours-per-week'] == 99]
countn = coun.shape[0]
print (f"Количество работающих :столько времени {countn}")

perc = float(coun.loc[data['salary'] == ' >50K'].shape[0]) / countn * 100
print (f"Процент тех, кто зарабатывает более 50 тысяч {perc}")
```

```
Максимальное количество часов в неделю: 99
Количество работающих :столько времени 85
Процент тех, кто зарабатывает более 50 тысяч 29.411764705882355
```

10. Посчитаете среднее время работы в неделю для тех кто получает много и мало, для каждой страны. Какими они будут для Японии?

In [16]:

```
rich = data.loc[data['salary'] == ' >50K']
poor = data.loc[data['salary'] == ' <=50K']

print ("Среднее пооличество часов работы в неделю \n")
for country in data['native-country'].unique():
    print(country)
    print(f"Зарплата больше 50 тысяч: {rich.loc[rich['native-country'] == country,
 'hours-per-week'].mean()}")
    print(f"Зарплата меньше 50 тысяч: {poor.loc[poor['native-country'] == country,
 'hours-per-week'].mean()}\n")
```

Среднее пооличество часов работы в неделю

```
 United-States
Зарплата больше 50 тысяч: 45.50536884674383
Зарплата меньше 50 тысяч: 38.79912723305605

 Cuba
Зарплата больше 50 тысяч: 42.44
Зарплата меньше 50 тысяч: 37.98571428571429

 Jamaica
Зарплата больше 50 тысяч: 41.1
Зарплата меньше 50 тысяч: 38.23943661971831

 India
Зарплата больше 50 тысяч: 46.475
Зарплата меньше 50 тысяч: 38.233333333333334

 ?
Зарплата больше 50 тысяч: 45.54794520547945
Зарплата меньше 50 тысяч: 40.16475972540046

 Mexico
Зарплата больше 50 тысяч: 46.57575757575758
Зарплата меньше 50 тысяч: 40.00327868852459

 South
Зарплата больше 50 тысяч: 51.4375
Зарплата меньше 50 тысяч: 40.15625

 Puerto-Rico
Зарплата больше 50 тысяч: 39.416666666666664
Зарплата меньше 50 тысяч: 38.470588235294116

 Honduras
Зарплата больше 50 тысяч: 60.0
Зарплата меньше 50 тысяч: 34.333333333333336

 England
Зарплата больше 50 тысяч: 44.53333333333333
Зарплата меньше 50 тысяч: 40.483333333333334

 Canada
Зарплата больше 50 тысяч: 45.64102564102564
Зарплата меньше 50 тысяч: 37.91463414634146

 Germany
Зарплата больше 50 тысяч: 44.97727272727273
Зарплата меньше 50 тысяч: 39.13978494623656

 Iran
Зарплата больше 50 тысяч: 47.5
Зарплата меньше 50 тысяч: 41.44

 Philippines
Зарплата больше 50 тысяч: 43.032786885245905
Зарплата меньше 50 тысяч: 38.065693430656935

 Italy
Зарплата больше 50 тысяч: 45.4
Зарплата меньше 50 тысяч: 39.625
```

```
 Poland
Зарплата больше 50 тысяч: 39.0
Зарплата меньше 50 тысяч: 38.166666666666664

 Columbia
Зарплата больше 50 тысяч: 50.0
Зарплата меньше 50 тысяч: 38.68421052631579

 Cambodia
Зарплата больше 50 тысяч: 40.0
Зарплата меньше 50 тысяч: 41.416666666666664

 Thailand
Зарплата больше 50 тысяч: 58.333333333333336
Зарплата меньше 50 тысяч: 42.86666666666667

 Ecuador
Зарплата больше 50 тысяч: 48.75
Зарплата меньше 50 тысяч: 38.041666666666664

 Laos
Зарплата больше 50 тысяч: 40.0
Зарплата меньше 50 тысяч: 40.375

 Taiwan
Зарплата больше 50 тысяч: 46.8
Зарплата меньше 50 тысяч: 33.774193548387096

 Haiti
Зарплата больше 50 тысяч: 42.75
Зарплата меньше 50 тысяч: 36.325

 Portugal
Зарплата больше 50 тысяч: 41.5
Зарплата меньше 50 тысяч: 41.93939393939394

 Dominican-Republic
Зарплата больше 50 тысяч: 47.0
Зарплата меньше 50 тысяч: 42.338235294117645

 El-Salvador
Зарплата больше 50 тысяч: 45.0
Зарплата меньше 50 тысяч: 36.03092783505155

 France
Зарплата больше 50 тысяч: 50.75
Зарплата меньше 50 тысяч: 41.05882352941177

 Guatemala
Зарплата больше 50 тысяч: 36.666666666666664
Зарплата меньше 50 тысяч: 39.36065573770492

 China
Зарплата больше 50 тысяч: 38.9
Зарплата меньше 50 тысяч: 37.38181818181818

 Japan
Зарплата больше 50 тысяч: 47.958333333333336
Зарплата меньше 50 тысяч: 41.0
```

```
 Yugoslavia
Зарплата больше 50 тысяч: 49.5
Зарплата меньше 50 тысяч: 41.6

 Peru
Зарплата больше 50 тысяч: 40.0
Зарплата меньше 50 тысяч: 35.06896551724138

 Outlying-US(Guam-USVI-etc)
Зарплата больше 50 тысяч: nan
Зарплата меньше 50 тысяч: 41.857142857142854

 Scotland
Зарплата больше 50 тысяч: 46.666666666666664
Зарплата меньше 50 тысяч: 39.44444444444444

 Trinadad&Tobago
Зарплата больше 50 тысяч: 40.0
Зарплата меньше 50 тысяч: 37.05882352941177

 Greece
Зарплата больше 50 тысяч: 50.625
Зарплата меньше 50 тысяч: 41.80952380952381

 Nicaragua
Зарплата больше 50 тысяч: 37.5
Зарплата меньше 50 тысяч: 36.09375

 Vietnam
Зарплата больше 50 тысяч: 39.2
Зарплата меньше 50 тысяч: 37.193548387096776

 Hong
Зарплата больше 50 тысяч: 45.0
Зарплата меньше 50 тысяч: 39.142857142857146

 Ireland
Зарплата больше 50 тысяч: 48.0
Зарплата меньше 50 тысяч: 40.94736842105263

 Hungary
Зарплата больше 50 тысяч: 50.0
Зарплата меньше 50 тысяч: 31.3

 Holand-Netherlands
Зарплата больше 50 тысяч: nan
Зарплата меньше 50 тысяч: 40.0
```

In [17]:

```
print("Среднее количество часов для Японии")
print(f"Зарплата больше 50 тысяч: {rich.loc[rich['native-country'] == ' Japan', 'ho
urs-per-week'].mean()}")
print(f"Зарплата меньше 50 тысяч: {poor.loc[poor['native-country'] == ' Japan', 'ho
urs-per-week'].mean()}\n")
```

```
Среднее количество часов для Японии
Зарплата больше 50 тысяч: 47.958333333333336
Зарплата меньше 50 тысяч: 41.0
```

Объединение данных

In [2]:

```
import pandas as pd
```

Используем два набора данных

In [4]:

```
Data1 = pd.read_csv("user_device.csv")
Data2 = pd.read_csv("user_usage.csv")
```

In [5]:

```
Data1.head()
```

Out[5]:

	use_id	user_id	platform	platform_version	device	use_type_id
0	22782	26980	ios	10.2	iPhone7,2	2
1	22783	29628	android	6.0	Nexus 5	3
2	22784	28473	android	5.1	SM-G903F	1
3	22785	15200	ios	10.2	iPhone7,2	3
4	22786	28239	android	6.0	ONE E1003	1

In [6]:

```
Data2.head()
```

Out[6]:

	outgoing_mins_per_month	outgoing_sms_per_month	monthly_mb	use_id
0	21.97	4.82	1557.33	22787
1	1710.08	136.88	7267.55	22788
2	1710.08	136.88	7267.55	22789
3	94.46	35.17	519.12	22790
4	71.59	79.26	1557.33	22792

In [11]:

```
print(f"Data1: {Data1.shape}")
print(f"Data2: {Data2.shape}")
```

```
Data1: (272, 6)
Data2: (240, 4)
```

Для "склеивания" используем "Outer Merge"

In [12]:

```
bresult = pd.merge(Data2, Data1[['use_id', 'platform', 'device']], on='use_id',
how='outer', indicator=True)
```

In [14]:

```
result
```

Out[14]:

	outgoing_mins_per_month	outgoing_sms_per_month	monthly_mb	use_id	platform	d
0	21.97	4.82	1557.33	22787	android	GT-I
1	1710.08	136.88	7267.55	22788	android	G
2	1710.08	136.88	7267.55	22789	android	G
3	94.46	35.17	519.12	22790	android	D
4	71.59	79.26	1557.33	22792	android	G
5	71.59	79.26	1557.33	22793	android	G
6	71.59	79.26	519.12	22794	android	G
7	71.59	79.26	519.12	22795	android	G
8	30.92	22.77	3114.67	22799	android	ONE A
9	69.80	14.70	25955.55	22801	android	GT-I
10	554.41	150.06	3114.67	22804	android	G
11	189.10	24.08	519.12	22805	android	GT-I
12	283.30	107.47	15573.33	22806	android	A
13	324.34	92.52	519.12	22808	android	G
14	797.06	7.67	519.12	22813	android	D
15	797.06	7.67	15573.33	22814	android	D
16	797.06	7.67	15573.33	22815	android	D
17	797.06	7.67	15573.33	22816	android	D
18	797.06	7.67	15573.33	22817	android	D
19	78.80	327.33	10382.21	22819	android	HTC r
20	78.80	327.33	15573.33	22820	android	HTC r
21	78.80	327.33	15573.33	22822	android	HTC r
22	164.10	192.64	3114.67	22823	android	G
23	208.26	91.76	5191.12	22824	android	G

	outgoing_mins_per_month	outgoing_sms_per_month	monthly_mb	use_id	platform	d
24	681.44	47.35	1271.39	22829	ios	iPho
25	324.27	91.50	519.12	22830	android	G
26	85.97	26.94	407.01	22831	android	I8
27	244.88	105.95	1557.33	22832	android	D
28	135.09	42.02	5191.12	22833	android	E
29	57.49	16.73	15573.33	22839	android	A
...	...	...	...	...	...	
323	NaN	NaN	NaN	22976	ios	iPho
324	NaN	NaN	NaN	22983	ios	iPho
325	NaN	NaN	NaN	22984	ios	iPho
326	NaN	NaN	NaN	22990	android	HU VNS
327	NaN	NaN	NaN	22993	android	N
328	NaN	NaN	NaN	22996	ios	iPho
329	NaN	NaN	NaN	23000	android	HU VNS
330	NaN	NaN	NaN	23001	android	G
331	NaN	NaN	NaN	23004	ios	iPho
332	NaN	NaN	NaN	23006	ios	iPho
333	NaN	NaN	NaN	23007	ios	iPho
334	NaN	NaN	NaN	23008	ios	iPho
335	NaN	NaN	NaN	23009	ios	iPho
336	NaN	NaN	NaN	23010	ios	iPho
337	NaN	NaN	NaN	23011	ios	iPho
338	NaN	NaN	NaN	23014	ios	iPho
339	NaN	NaN	NaN	23022	ios	iP
340	NaN	NaN	NaN	23025	ios	iPho
341	NaN	NaN	NaN	23033	ios	iPho
342	NaN	NaN	NaN	23034	android	J3
343	NaN	NaN	NaN	23035	ios	iPho
344	NaN	NaN	NaN	23037	ios	iPho
345	NaN	NaN	NaN	23038	ios	iPho
346	NaN	NaN	NaN	23042	android	G
347	NaN	NaN	NaN	23045	ios	iPho
348	NaN	NaN	NaN	23047	ios	iPho

	outgoing_mins_per_month	outgoing_sms_per_month	monthly_mb	use_id	platform	d
349	NaN	NaN	NaN	23048	android	ONEI A
350	NaN	NaN	NaN	23050	ios	iPhoı
351	NaN	NaN	NaN	23051	ios	iPhoı
352	NaN	NaN	NaN	23052	ios	iPhoı

353 rows × 7 columns

Проверим, сколько было уникальных id в обеих таблицах

In [17]:

```
print(pd.concat([Data1['use_id'], Data2['use_id']]).unique().shape[0])
```

```
353
```

In [ ]:

```
data_cop = data.copy()
```

Значит объединение прошло без утери данных