Московский государственный технический университет им. Н.Э. Баумана.

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №2

Выполнил:
студент группы ИУ5-62Б
Брусникина Мария
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2020 г.

Лабораторная работа №2.

Изучение библиотек обработки данных

Цель лабораторной работы

Изучение библиотеки обработки данных Pandas.

Задание

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

Условие задания - https://nbviewer.jupyter.org/github/Yorko/mlcourse_open/blob/master/jupyter_english/assignments_demo/assignment01_pandas_uci_a 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

Поэтому готовый набор данных для лабораторной работы удобнее скачать здесь - 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)

Содержение лабораторной работы

In [2]:

```
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
# we don't like warnings
# you can comment the following 2 lines if you'd like to
import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
data = pd.read_csv('C:/Users/brusn/Desktop/TMO/lab2/adult.data.txt', sep=",")
data.head()
```

Out[4]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba

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

In [5]:

```
# value_counts() - Return a Series containing counts of unique values.
data['sex'].value_counts()
```

Out[5]:

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

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

In [7]:

```
# .loc[] - Access a group of rows and columns by a boolean array.
# mean() - Compute mean of groups, excluding missing values.
data.loc[data['sex'] == 'Female', 'age'].mean()
```

Out[7]:

```
36.85823043357163
```

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

In [8]:

```
# .shape - Return a tuple representing the dimensionality of the DataFrame.
float((data['native-country'] == 'Germany').sum()) / data.shape[0]
```

Out[8]:

```
0.004207487485028101
```

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 [10]:

```
# .round() - Round a DataFrame to a variable number of decimal places.
# .std() - Compute standard deviation of groups, excluding missing values.
ages1 = data.loc[data['salary'] == '>50K', 'age']
ages2 = data.loc[data['salary'] == '<=50K', 'age']
print("Средний возраст \"богатых\": {0} +- {1} лет, \"бедных\" - {2} +- {3} лет.".format(
    round(ages1.mean()), round(ages1.std(), 1),
    round(ages2.mean()), round(ages2.std(), 1)))
```

```
Средний возраст "богатых": 44.0 +- 10.5 лет, "бедных" - 37.0 +- 14.0 лет.
```

1. 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)

In [11]:

```
data.loc[data['salary'] == '>50K', 'education'].unique() # => No
```

Out[11]:

```
array(['HS-grad', 'Masters', 'Bachelors', 'Some-college', 'Assoc-voc',
       'Doctorate', 'Prof-school', 'Assoc-acdm', '7th-8th', '12th',
       '10th', '11th', '9th', '5th-6th', '1st-4th'], dtype=object)
```

1. 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 [20]:

```
print(data.groupby(['race','sex'])['age'].describe())
#for (race, sex), sub_df in data.groupby(['race', 'sex']):
    #print("Race: {0}, sex: {1}".format(race, sex))
    #print(sub_df['age'].describe())
```

```
                            count       mean        std   min   25%   50%  \
race               sex
Amer-Indian-Eskimo Female    119.0  37.117647  13.114991  17.0  27.0  36.0
                   Male      192.0  37.208333  12.049563  17.0  28.0  35.0
Asian-Pac-Islander Female    346.0  35.089595  12.300845  17.0  25.0  33.0
                   Male      693.0  39.073593  12.883944  18.0  29.0  37.0
Black              Female   1555.0  37.854019  12.637197  17.0  28.0  37.0
                   Male     1569.0  37.682600  12.882612  17.0  27.0  36.0
Other              Female    109.0  31.678899  11.631599  17.0  23.0  29.0
                   Male      162.0  34.654321  11.355531  17.0  26.0  32.0
White              Female   8642.0  36.811618  14.329093  17.0  25.0  35.0
                   Male    19174.0  39.652498  13.436029  17.0  29.0  38.0

                            75%   max
race               sex
Amer-Indian-Eskimo Female  46.00  80.0
                   Male    45.00  82.0
Asian-Pac-Islander Female  43.75  75.0
                   Male    46.00  90.0
Black              Female  46.00  90.0
                   Male    46.00  90.0
Other              Female  39.00  74.0
                   Male    42.00  77.0
White              Female  46.00  90.0
                   Male    49.00  90.0
```

1. 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 [13]:

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

Out[13]:

```
<=50K    7552
>50K      697
Name: salary, dtype: int64
```

In [14]:

```
#женатые
data.loc[(data['sex'] == 'Male') &
     (data['marital-status'].str.startswith('Married')), 'salary'].value_counts()
```

Out[14]:

```
<=50K    7576
>50K     5965
Name: salary, dtype: int64
```

In [15]:

In [15]:

```
data['marital-status'].value_counts()
```

Out[15]:

```
Married-civ-spouse       14976
Never-married            10683
Divorced                  4443
Separated                 1025
Widowed                    993
Married-spouse-absent      418
Married-AF-spouse           23
Name: marital-status, dtype: int64
```

Доля тех, кто зарабатывает много, больше среди женатых.

1. 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 [17]:

```
max_load = data['hours-per-week'].max()
print("Макс. время - {0} ч/нед.".format(max_load))

num_workaholics = data[data['hours-per-week'] == max_load].shape[0]
print("Число таких работников - {0}".format(num_workaholics))

rich_share = float(data[(data['hours-per-week'] == max_load)
                   & (data['salary'] == '>50K')].shape[0]) / num_workaholics
print("Среди них процент \"богатых\" - {0}%".format(int(100 * rich_share)))
```

```
Макс. время - 99 ч/нед.
Число таких работников - 85
Среди них процент "богатых" - 29%
```

1. 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 [18]:

```
# .crosstab() - Compute a simple cross tabulation of two (or more) factors.
pd.crosstab(data['native-country'], data['salary'],
            values=data['hours-per-week'], aggfunc=np.mean).T
```

Out[18]:

native-country	?	Cambodia	Canada	China	Columbia	Cuba	Dominican-Republic	Ecuador	El-Salvador	England	France
salary											
<=50K	40.164760	41.416667	37.914634	37.381818	38.684211	37.985714	42.338235	38.041667	36.030928	40.483333	41.058824
>50K	45.547945	40.000000	45.641026	38.900000	50.000000	42.440000	47.000000	48.750000	45.000000	44.533333	50.750000

In [ ]:

```

```