МГТУ им. Н. Э. Баумана

кафедра ИУ5

курс «Технологии машинного обучения»

Лабораторная работа №2

«Изучение библиотек обработки данных»

ВЫПОЛНИЛ:

Болгова А. В.

Группа ИУ5-61Б

ПРОВЕРИЛ:

Гапанюк Ю. Е.

Москва, 2020 г.

Цель лабораторной работы: изучение библиотек 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_adult.ipynb?flush_cache=true

Официальный датасет находится здесь:
https://archive.ics.uci.edu/ml/datasets/Adult

Готовый набор данных для лабораторной работы здесь:
https://raw.githubusercontent.com/Yorko/mlcourse.ai/master/data/adult.data.csv

Выполнение работы

```
In [89]: 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

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

```
In [90]: data = pd.read_csv('../data/adult.data.csv', sep=",")
         data.head()
```

Out[90]:

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

```
In [91]: # 1. How many men and women (sex feature) are represented in this dataset?
         data['sex'].value_counts()
```

```
Out[91]: Male      21790
         Female    10771
         Name: sex, dtype: int64
```

```
In [92]: # 2. What is the average age (age feature) of women?
         data.loc[data['sex'] == 'Female', 'age'].mean()
```

```
Out[92]: 36.85823043357163
```

```
In [93]: # 3. What is the percentage of German citizens (native-country feature)?
         float((data['native-country'] == 'Germany').sum()) / data.shape[0]
```

```
Out[93]: 0.004207487485028101
```

```
In [94]: # 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?
         ages1 = data.loc[data['salary'] == '>50K', 'age']
         ages2 = data.loc[data['salary'] == '<=50K', 'age']
         print("The average age of the rich: {0} +- {1} years, poor - {2} +- {3} years.".format(
             round(ages1.mean()), round(ages1.std(), 1),
             round(ages2.mean()), round(ages2.std(), 1)))
```

```
The average age of the rich: 44.0 +- 10.5 years, poor - 37.0 +- 14.0 years.
```

```
In [95]:  # 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)
          data.loc[data['salary'] == '>50K', 'education'].unique()

Out[95]: 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)


In [96]:  # 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.
          for (race, sex), sub_df in data.groupby(['race', 'sex']):
              print("Race: {0}, sex: {1}".format(race, sex))
              print(sub_df['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
```

In [97]: # 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.
         data['marital-status'].value_counts()

Out[97]: 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

In [98]: data.loc[(data['sex'] == 'Male') &
                (data['marital-status'].str.startswith('Married')), 'salary'].value_counts()

Out[98]: <=50K   7576
         >50K   5965
         Name: salary, dtype: int64

In [99]: data.loc[(data['sex'] == 'Male') &
                (data['marital-status'].isin(['Never-married',
                                             'Separated',
                                             'Divorced',
                                             'Widowed'])), 'salary'].value_counts()

Out[99]: <=50K   7552
         >50K   697
         Name: salary, dtype: int64
```

```python
# 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?

max_load = data['hours-per-week'].max()
print("Max time - {0} hours./week.".format(max_load))

num_workaholics = data[data['hours-per-week'] == max_load].shape[0]
print("Total number of such hard workers {0}".format(num_workaholics))

rich_share = float(data[(data['hours-per-week'] == max_load)
                & (data['salary'] == '>50K')].shape[0]) / num_workaholics
print("Percentage of rich among them {0}%".format(int(100 * rich_share)))
```

```
Max time - 99 hours./week.
Total number of such hard workers 85
Percentage of rich among them 29%
```

```python
# 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?

pd.crosstab(data['native-country'], data['salary'],
            values=data['hours-per-week'], aggfunc=np.mean).T
```

native-country	?	Cambodia	Canada	China	Columbia	Cuba	Dominican-Republic	Ecuador	El-Salvador	England	France	Germany	Greece	Gua
salary														
<=50K	40.164760	41.416667	37.914634	37.381818	38.684211	37.985714	42.338235	38.041667	36.030928	40.483333	41.058824	39.139785	41.809524	39.
>50K	45.547945	40.000000	45.641026	38.900000	50.000000	42.440000	47.000000	48.750000	45.000000	44.533333	50.750000	44.977273	50.625000	36.