Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления
КАФЕДРА Системы обработки информации и управления

Отчет по лабораторной работе №2
«Изучение библиотек обработки данных.»
по курсу «Технологии машинного обучения»

Выполнил:
Студент группы ИУ5Ц-81Б
Тураев Глеб

Проверил:
Преподаватель кафедры ИУ5
Гапанюк Ю.Е.

Москва 2020

Цель лабораторной работы: Изучить библиотеку обработки данных Pandas.

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

Выполнение лабораторной работы:

1) **Текстовое описание данных**

- **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

2) **Импортируем библиотеки:**

Осуществим импорт библиотек с помощью команды **import:**

```
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')
```

3) **Осуществим загрузку датасета и выведем первые пять строк:**

```
[40] data = pd.read_csv('/adult.data.csv')
     data.head()
```

	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

4) **Узнаем количество мужчин и женщин, представленных в этом наборе данных (касательно половых признаков) (ПУНКТ №1):**

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

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

5) **Выясним средний возраст женщин (касательно возрастной характеристики) (ПУНКТ №2):**

```
[42] print(round(float(data.loc[data['sex']=='Female', ['age']].mean())))
```

```
37
```

6) **Определим процентную долю граждан Германии? (ПУНКТ №3)**

```
[43] print(round(float(data.loc[data['native-country']=='Germany', ['native-country']].count()/data['native-country'].count()*100),2),'%')
```

```
0.42 %
```

7) **Уточним среднее и стандартное отклонение возраста для тех, кто зарабатывает более 50 тыс/год и тех, кто зарабатывает менее 50 тыс/год (ПУНКТ №4-5):**

```
[44] print('Standard deviation for those who earn <= 50K:', float (data.loc[data['salary']=='<=50K', ['age']].std()))
     print('Mean deviation for those who earn <= 50K:', float (data.loc[data['salary']=='<=50K', ['age']].mad()))
     print('Standard deviation for those who earn > 50K:', float (data.loc[data['salary']=='>50K', ['age']].std()))
     print('Mean deviation for those who earn > 50K:', float (data.loc[data['salary']=='>50K', ['age']].mad()))
```

```
Standard deviation for those who earn <= 50K: 14.020088490824813
Mean deviation for those who earn <= 50K: 11.467855024821914
Standard deviation for those who earn > 50K: 10.519027719851785
Mean deviation for those who earn > 50K: 8.47674579194268
```

8) **Выясним, правда ли, что люди, зарабатывавшие более 50 тыс., имеют по крайней мере высшее образование (образование: бакалавры, проф-школа, ДОЦ-acdm, ДОЦ-voc, магистры или докторантура характеристика) (ПУНКТ №6):**

```
[45] highEduList = ['Bachelors', 'Prof-school', 'Assoc-acdm', 'Assoc-voc', 'Masters', 'Doctorate']
     dataList = list(data.loc[data['salary']=='>50K', 'education'].unique())
     fl = True;
     for a in dataList:
       for b in highEduList:
         if a==b:
           fl = True;
           break;
         if a!=b:
           fl = False;
     print(fl);
```

```
False
```

9) **Отобразим статистику возраста для каждой расы (функция расы) и каждого пола (функция пола). Используем для этого groupby() и describe(). Найдем максимальный возраст мужчин Амер-индо-эскимосской расы (ПУНКТ №7):**

```
[46] data.groupby(["race", "sex"])["age"].describe()
```

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

```
[47] data[(data["race"] == "Amer-Indian-Eskimo") & (data["sex"] == "Male")]["age"].max()
```

```
82
```

10) **Получим результаты, среди кого больше доля тех, кто зарабатывает много (>50 тыс.): женатых или одиноких мужчин (>характеристика семейного положения) (Считать женатыми тех, кто имеет семейное положение, начиная с женатого (женат – гражданский супруг, женат – отсутствующий супруг или женат – AF – супруг), остальных считать холостяками) (ПУНКТ №8):**

```
[48] def is_married(m):
         return m.startswith("Married")

     data["married"] = data["marital-status"].map(is_married)
     (data[(data["sex"] == "Male") & (data["salary"] == ">50K")]
          ["married"].value_counts())
```

```
True     5965
False     697
Name: married, dtype: int64
```

11) Узнаем максимальное количество часов, которое человек работает в неделю, какое количество людей работает такое количество часов и, каков процент тех, кто зарабатывает много (>50K) среди них (ПУНКТ №9):

```
[49] m = data["hours-per-week"].max()
     print("Maximum is {} hours/week;".format(m))

     people = data[data["hours-per-week"]==m]
     c = people.shape[0]
     print("{} people work this time at week;".format(c))

     s = people[people["salary"] == ">50K"].shape[0]
     print("{0:%} get >50K salary.".format(s/c))
```

```
Maximum is 99 hours/week;
85 people work this time at week;
29.411765% get >50K salary.
```

12) Подсчитаем среднее время работы (часов в неделю) для тех, кто зарабатывает мало и много (заработная плата) для каждой страны (родной страны). И что, если это будет Япония (ПУНКТ №10):

```
[50] p = pd.crosstab(data["native-country"], data["salary"], values=data['hours-per-week'], aggfunc="mean")
     print(p);
```

salary	<=50K	>50K
native-country		
?	40.164760	45.547945
Cambodia	41.416667	40.000000
Canada	37.914634	45.641026
China	37.381818	38.900000
Columbia	38.684211	50.000000
Cuba	37.985714	42.440000
Dominican-Republic	42.338235	47.000000
Ecuador	38.041667	48.750000
El-Salvador	36.030928	45.000000
England	40.483333	44.533333
France	41.058824	50.750000
Germany	39.139785	44.977273
Greece	41.809524	50.625000
Guatemala	39.360656	36.666667
Haiti	36.325000	42.750000
Holand-Netherlands	40.000000	NaN
Honduras	34.333333	60.000000
Hong	39.142857	45.000000
Hungary	31.300000	50.000000
India	38.233333	46.475000
Iran	41.440000	47.500000
Ireland	40.947368	48.000000
Italy	39.625000	45.400000
Jamaica	38.239437	41.100000
Japan	41.000000	47.958333
Laos	40.375000	40.000000
Mexico	40.003279	46.575758
Nicaragua	36.093750	37.500000
Outlying-US(Guam-USVI-etc)	41.857143	NaN
Peru	35.068966	40.000000
Philippines	38.065693	43.032787
Poland	38.166667	39.000000
Portugal	41.939394	41.500000
Puerto-Rico	38.470588	39.416667
Scotland	39.444444	46.666667
South	40.156250	51.437500
Taiwan	33.774194	46.800000
Thailand	42.866667	58.333333
Trinadad&Tobago	37.058824	40.000000
United-States	38.799127	45.505369
Vietnam	37.193548	39.200000
Yugoslavia	41.600000	49.500000