

Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»



Отчет
Лабораторные работа № 2
«Изучение библиотек обработки данных»
По курсу «Технологии машинного обучения»

ИСПОЛНИТЕЛЬ:

Молева Анастасия
Группа ИУ5-61Б

_____ 2020 г.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

_____ 2020 г.

Москва 2020

Цель лабораторной работы: изучение библиотеки обработки данных Pandas.

Выполнение задания:

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

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

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

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

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

```
36.85823043357163
```

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

```
##-- First method
print("Germany: ", data['native-country'].value_counts()['Germany'])
print("All: ", data['native-country'].count())
print("Germany(perc): ", round(data['native-country'].value_counts()['Germany'] / data['native-country'].count() * 100, 2), '%')
```

```
Germany: 137
All: 32561
Germany(perc): 0.42 %
```

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? ¶

```
a = [round(data.loc[data['salary']=='>50K', 'age'].mean(), 0), round(data.loc[data['salary']=='>50K', 'age'].std(), 2)]
b = [round(data.loc[data['salary']=='<=50K', 'age'].mean(), 0), round(data.loc[data['salary']=='<=50K', 'age'].std(), 1)]
df = pd.DataFrame([a, b], columns=['mean', 'std'], index=['>50K', '<=50K'])
df
```

	mean	std
>50K	44.0	10.52
<=50K	37.0	14.00

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'].value_counts()
```

```
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
```

По большей части, это правда

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
```

```
df = pd.DataFrame(data.loc[data['race']=='Amer-Indian-Eskimo', 'age'].max(), columns=['Max Age'], index=['Amer-Indian-Eskimo'])
df
```

	Max Age
Amer-Indian-Eskimo	82

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.

```
stat = data.loc[data['salary']=='>50K', 'marital-status'].value_counts()
stat
married_count = 0
for i in stat.items():
    if i[0].startswith('Married'):
        married_count += i[1]
all_stat = data.loc[data['salary']=='>50K', 'marital-status'].count()

married = married_count / all_stat * 100
df = pd.DataFrame([married, 100-married], columns=['%'], index=['Marry', 'Not marry'])
df
```

	%
Marry	85.90741
Not marry	14.09259

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?

```
data['hours-per-week'].describe()['max']
```

```
99.0
```

```
many_hours = data.loc[data['hours-per-week']==99, 'workclass'].count()
many_hours
```

```
85
```

```

salary = data.loc[data['hours-per-week']==99, 'salary']
count_big_salary = 0
for i in salary.items():
    if i[1] == '>50K':
        count_big_salary += 1
count_big_salary
small_salary = many_hours - count_big_salary
perc_big_salary = count_big_salary / many_hours * 100
perc_small_salary = small_salary / many_hours * 100

```

```

df = pd.DataFrame([[count_big_salary, perc_big_salary], [small_salary, perc_small_salary]],
                  columns=['count people', '%'], index=['>50K', '<=50K'])
df

```

	count people	%
>50K	25	29.411765
<=50K	60	70.588235

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?

```

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

```

native-country	?	Cambodia	Canada	China	Columbia	Cuba	Dominican-Republic	Ecuador	El-Salvador	England	...	Portugal	Puerto-Rico	Scotland
salary														
<=50K	40.164760	41.416667	37.914634	37.381818	38.684211	37.985714	42.338235	38.041667	36.030928	40.483333	...	41.939394	38.470588	39.444444
>50K	45.547945	40.000000	45.641026	38.900000	50.000000	42.440000	47.000000	48.750000	45.000000	44.533333	...	41.500000	39.416667	46.666667

2 rows x 42 columns

```
df['Japan']
```

```

salary
<=50K    41.000000
>50K     47.958333
Name: Japan, dtype: float64

```

Вывод:

Познакомилась с библиотекой Pandas.