МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

Лабораторная работа №2

по курсу «Методы машинного обучения»
на тему «Изучение библиотек обработки данных.»

Выполнил: Сефербеков М.С
группа ИУ5-21М

Москва - 2020

[mlcourse.ai (mlcourse.ai)](mlcourse.ai) – Open Machine Learning Course

Author: [Yury Kashnitskiy (http://yorko.github.io)](http://yorko.github.io)
Translated and edited by [Sergey Isaev (https://www.linkedin.com/in/isvforall/)](https://www.linkedin.com/in/isvforall/), [Artem Trunov (https://www.linkedin.com/in/datamove/)](https://www.linkedin.com/in/datamove/), [Anastasia Manokhina (https://www.linkedin.com/in/anastasiamanokhina/)](https://www.linkedin.com/in/anastasiamanokhina/), and [Yuanyuan Pao (https://www.linkedin.com/in/yuanyuanpao/)](https://www.linkedin.com/in/yuanyuanpao/)

Assignment #1 (demo)

Exploratory data analysis with Pandas

In this task you should use Pandas to answer a few questions about the [Adult (https://archive.ics.uci.edu/ml/datasets/Adult)](https://archive.ics.uci.edu/ml/datasets/Adult) dataset. (You don't have to download the data – it's already here). Choose the answers in the [web-form (https://docs.google.com/forms/d/1uY7MpI2trKx6FLWZte0uVh3ULV4Cm_tDud0VDFGCOKg)](https://docs.google.com/forms/d/1uY7MpI2trKx6FLWZte0uVh3ULV4Cm_tDud0VDFGCOKg). This is a demo version of an assignment, so by submitting the form, you'll see a link to the solution .ipynb file.

Unique values of all features (for more information, please see the links above):

- `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 [65]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
data = pd.read_csv('adult.data.csv')
data.head()
```

Out[3]:

	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

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

In [11]:
```
data['sex'].value_counts()
```

Out[11]:
```
Male      21790
Female    10771
Name: sex, dtype: int64
```

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

In [15]:
```
data[data['sex']=='Female']['age'].mean()
```

Out[15]:
```
36.85823043357163
```

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

In [20]:
```
(data['native-country']=='Germany').sum()/data.shape[0]
```

Out[20]:
```
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 [34]:
```
print('<=50k mean='+str((data[data['salary']=='<=50K']['age'].mean())))
print('<=50k std='+str((data[data['salary']=='<=50K']['age'].std())))
print('>50k  mean='+str((data[data['salary']=='>50K']['age'].mean())))
print('>50k  std='+str((data[data['salary']=='>50K']['age'].std())))
```

```
<=50k mean=36.78373786407767
<=50k std=14.020088490824813
>50k  mean=44.24984058155847
>50k  std=10.51902771985177
```

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)

In [36]:
```
data[data['salary']=='>50K']['education'].value_counts() #False
```

Out[36]:
```
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
```

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.

In [42]:
```
print(data.groupby(['race'])['sex'].describe())
print('maximum age of men of Amer-Indian-Eskimo race:'+str(data[data['race']=='Amer-Indian-Eskimo']['age'].max()))
```

```
                    count unique   top   freq
race
Amer-Indian-Eskimo    311      2  Male    192
Asian-Pac-Islander   1039      2  Male    693
Black                3124      2  Male   1569
Other                 271      2  Male    162
White               27816      2  Male  19174
maximum age of men of Amer-Indian-Eskimo race: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.

In [53]:
```
notsingle=['Married-civ-spouse','Married-spouse-absent','Married-AF-spouse']
single=['Never-married','Separated','Divorced','Widowed']
print(data.loc[(data['sex'] == 'Male') & (data['marital-status'].isin(single)), 'salary'].value_counts())
print(data.loc[(data['sex'] == 'Male') & (data['marital-status'].isin(notsingle)), 'salary'].value_counts())
```

```
<=50K    7552
>50K      697
Name: salary, dtype: int64
<=50K    7576
>50K     5965
Name: salary, dtype: int64
```

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?

In [61]:
```
maxhour = data['hours-per-week'].max()
print('maxhour '+str(maxhour))
hardworkers=data[data['hours-per-week']==maxhour].shape[0]
print('hardworkers  '+str(hardworkers))
peoplewithmoney = (data[(data['hours-per-week'] == maxhour) & (data['salary'] == '>50K')].shape[0]) / hardworkers
print('peoplewithmoney '+ str(peoplewithmoney))
```

```
maxhour 99
hardworkers  85
peoplewithmoney 0.29411764705882354
```

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?

In [67]:
```
pd.crosstab(data['native-country'], data['salary'],
           values=data['hours-per-week'], aggfunc=np.mean)
```

Out[67]:

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

In [20]:
```
import pandasql as ps
user_usage = pd.read_csv('user_usage.csv')
user_device = pd.read_csv('user_device.csv')
pysqldf = lambda q: sqldf(q, globals())
```

In [21]:
```
result = pd.merge(user_usage,
                  user_device[['use_id', 'platform', 'device']],
                  on='use_id')
result.head()
```

Out[21]:

	outgoing_mins_per_month	outgoing_sms_per_month	monthly_mb	use_id	platform	device
0	21.97	4.82	1557.33	22787	android	GT-I9505
1	1710.08	136.88	7267.55	22788	android	SM-G930F
2	1710.08	136.88	7267.55	22789	android	SM-G930F
3	94.46	35.17	519.12	22790	android	D2303
4	71.59	79.26	1557.33	22792	android	SM-G361F

In [30]:
```
q = """SELECT
        user_device.use_id, user_device.platform, user_device.device
     FROM
        user_device
    JOIN user_usage
           ON user_usage.use_id = user_device.use_id;"""
joined=ps.sqldf(q, locals())
```

In [31]:
```
print(joined)
```

```
     use_id platform                 device
0     22787  android               GT-I9505
1     22788  android               SM-G930F
2     22789  android               SM-G930F
3     22790  android                  D2303
4     22792  android               SM-G361F
..      ...      ...                    ...
154   23043  android               SM-G900F
155   23044  android               SM-G900F
156   23046  android            Moto G (4)
157   23049  android               SM-G900F
158   23053  android  Vodafone Smart ultra 6

[159 rows x 3 columns]
```

In [34]:
```
result.groupby("platform").agg({
        "use_id": "count"})
```

Out[34]:

platform	use_id
android	157
ios	2

In [42]:
```
 q = "select platform ,sum(use_id) as use_id FROM result GROUP BY platform;"
group=ps.sqldf(q, locals())
```

In [43]:
```
print(group)
```

```
  platform   use_id
0  android  3598809
1      ios    45841
```

In [ ]: