Лабораторная работа

Обработка больших данных на Python

Задания:

1. Откройте файл данных, используйте DataFrame для копирования данных.
2. Выполните статистический анализ данных california_housing_train.csv.
3. Выполните операции доступа к отдельным строкам и столбцам данных california_housing_train.csv.
4. Постройте гистограмму распределения данных california_housing_train.csv.
5. Выполните агрегирование данных california_housing_train.csv.
6. Самостоятельно загрузите файл csv с сайта kaggle.com. Задайте названия столбцам на русском языке.
7. Обязательно приведите комментарии к программному коду.
8. Для сдачи работы нужно присдать програмнный код и исходный файл с сайта kaggle.com

Порядок выполнения заданий

1. Откройте файл данных, используйте DataFrame для копирования данных.

```
#1
from __future__ import print_function
import math
from IPython import display
```

```
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics
import tensorflow as tf
from tensorflow.python.data import Dataset

tf.logging.set_verbosity(tf.logging.ERROR)
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.1f}'.format
california_housing_dataframe = pd.read_csv("https://download.mlcc.google.com/mledu-
datasets/california_housing_train.csv", sep=",")
california_housing_dataframe= california_housing_dataframe.reindex(np.random.permutation(california_housing_dataframe.ind
ex))
california_housing_dataframe
```

```
#1
from __future__ import print_function

import math

from IPython import display
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
from sklearn import metrics
%tensorflow_version 1.x
import tensorflow as tf
from tensorflow.python.data import Dataset

#tf.logging.set_verbosity(tf.logging.ERROR)
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.1f}'.format

california_housing_dataframe = pd.read_csv("https://download.mlcc.google.com/mledu-datasets/california_housing_train.csv", sep=",")

california_housing_dataframe = california_housing_dataframe.reindex(
    np.random.permutation(california_housing_dataframe.index))
```

```
TensorFlow is already loaded. Please restart the runtime to change versions.
```

```
[ ] #2
california_housing_dataframe.describe()
```

2. Выполните статистический анализ данных.

```
#2
california_housing_dataframe.describe()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	17000.0	17000.0	17000.0	17000.0	17000.0	17000.0	17000.0	17000.0	17000.0
mean	-119.6	35.6	28.6	2643.7	539.4	1429.6	501.2	3.9	207300.9
std	2.0	2.1	12.6	2179.9	421.5	1147.9	384.5	1.9	115983.8
min	-124.3	32.5	1.0	2.0	1.0	3.0	1.0	0.5	14999.0
25%	-121.8	33.9	18.0	1462.0	297.0	790.0	282.0	2.6	119400.0
50%	-118.5	34.2	29.0	2127.0	434.0	1167.0	409.0	3.5	180400.0
75%	-118.0	37.7	37.0	3151.2	648.2	1721.0	605.2	4.8	265000.0
max	-114.3	42.0	52.0	37937.0	6445.0	35682.0	6082.0	15.0	500001.0

```
#3 прочитать определенные столбцы и вывести на экран
california_housing_dataframe[['latitude', 'total_bedrooms']]
```

	latitude	total_bedrooms
9928	36.8	470.0
5421	34.7	900.0
13385	38.4	211.0
3103	33.8	345.0
12648	36.9	695.0
...	...	...
7248	34.2	301.0
9665	36.5	343.0
3422	34.1	450.0
14305	37.1	136.0
16250	38.3	597.0

17000 rows × 2 columns

3Выполните операции доступа к отдельным строкам и столбцам данных.

```
#4 скопировать набор данных в новый и отобразить
california_housing_dataframe['COPYstolbec']=california_housing_dataframe['median_income'].copy()
california_housing_dataframe
```

```
#5
califDataCopy=california_housing_dataframe.copy()
califDataCopy.head(10).describe()
```

```
#6 вывести первые 15 записей
california_housing_dataframe.head(15)
```

```
#7 выыести первые 10 записей отсортированных по возрастанию по критерию популяции
california_housing_dataframe.head(10).sort_values(by='population')
```

```
#8 california_housing_dataframe.head(15)
#построить график
california_housing_dataframe.head(15).plot.bar(x='total_rooms', y='total_bedrooms')
```

```
#9 создаем срез первых двух строк
california_housing_dataframe.iloc[:2]
```

```
#10 последних двух строк
california_housing_dataframe.iloc[-2:]
```

```
#11 создание среза данных
california_housing_dataframe[california_housing_dataframe["population"] > 8000]
```

```
[ ]  #9 создаем срез первых двух строк
     california_housing_dataframe.iloc[:2]
```

```
[ ]  #10 последних двух строк
     california_housing_dataframe.iloc[-2:]
```

```
[ ]  #11 создание среза данных
     california_housing_dataframe[california_housing_dataframe["population"] > 8000]
```

```
[ ]  #12 срез с операцией И
     california_housing_dataframe[ (california_housing_dataframe["population"] > 13000) & (california_housing_dataframe["households"] > 5051) ]
```

```
[ ]  #13 срез с операцией XOR
     california_housing_dataframe[ (california_housing_dataframe["population"] > 13000) ^ (california_housing_dataframe["households"] > 5051) ]
```

```
     #14-удваиваем значение столбца
     california_housing_dataframe[["COPYstolbec"]].apply(lambda value: value*2)
```

Пример чтения файлов с диска и компьютера

Пример чтения файлов с компьютера

```
[ ] from google.colab import files
    uploaded = files.upload()

    # To store dataset in a Pandas Dataframe
    import io
    df2 = pd.read_csv(io.BytesIO(uploaded['diabetes.csv']))
```

Выбрать файлы Файл не выбран Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enal

```
Saving diabetes.csv to diabetes (2).csv
```

```
df2.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000

```
[ ] from google.colab import drive
    drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

```
import pandas as pd
data=pd.read_csv("/content/gdrive/My Drive/Colab Notebooks/Data/diabetes.csv")
data
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1