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)

Лабораторная работа №1

по курсу «Технологии машинного обучения»

Выполнил

студент группы ИУ5-64Б

Шпак И.Д.

Москва, 2020

1 Исходное задание

- Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь. Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных, например из Scikit-learn.
- Для лабораторных работ не рекомендуется выбирать датасеты большого размера.
- Создать ноутбук, который содержит следующие разделы:
    1. Для лабораторных работ не рекомендуется выбирать датасеты большого размера.
    2. Текстовое описание выбранного Вами набора данных.
    3. Основные характеристики датасета.
    4. Визуальное исследование датасета.
    5. Информация о корреляции признаков.
    6. Сформировать отчет и разместить его в своем репозитории на github.

2 Код программы

Выбранный набор данных - описывает стоимость и параметры жилья в Бостоне. Набор данных взят из базы sklearn, является одним из “игрушечных” наборов данных предназначенных для тренировки, пропуски в данных отсутствуют. Из отчета убран график pairplot библиотеки seaborn для облегчения конвертации в latex.

[2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_boston
%matplotlib inline
sns.set(style="ticks")
```

[13]:
```
boston = load_boston()
df = pd.DataFrame(boston.data, columns=boston.feature_names)
df.dtypes
```

[13]:
```
CRIM       float64
ZN         float64
INDUS      float64
CHAS       float64
NOX        float64
```

```
RM         float64
AGE        float64
DIS        float64
RAD        float64
TAX        float64
PTRATIO    float64
B          float64
LSTAT      float64
dtype: object
```

[4]:
```
df.describe()
```

[4]:

	CRIM	ZN	INDUS	CHAS	NOX	RM
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000

	AGE	DIS	RAD	TAX	PTRATIO	B
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032
std	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864
min	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000
25%	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500
50%	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000
75%	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000
max	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000

	LSTAT
count	506.000000
mean	12.653063
std	7.141062
min	1.730000
25%	6.950000
50%	11.360000

```
75%     16.955000
max     37.970000
```

[5]:
```
df.corr()
```

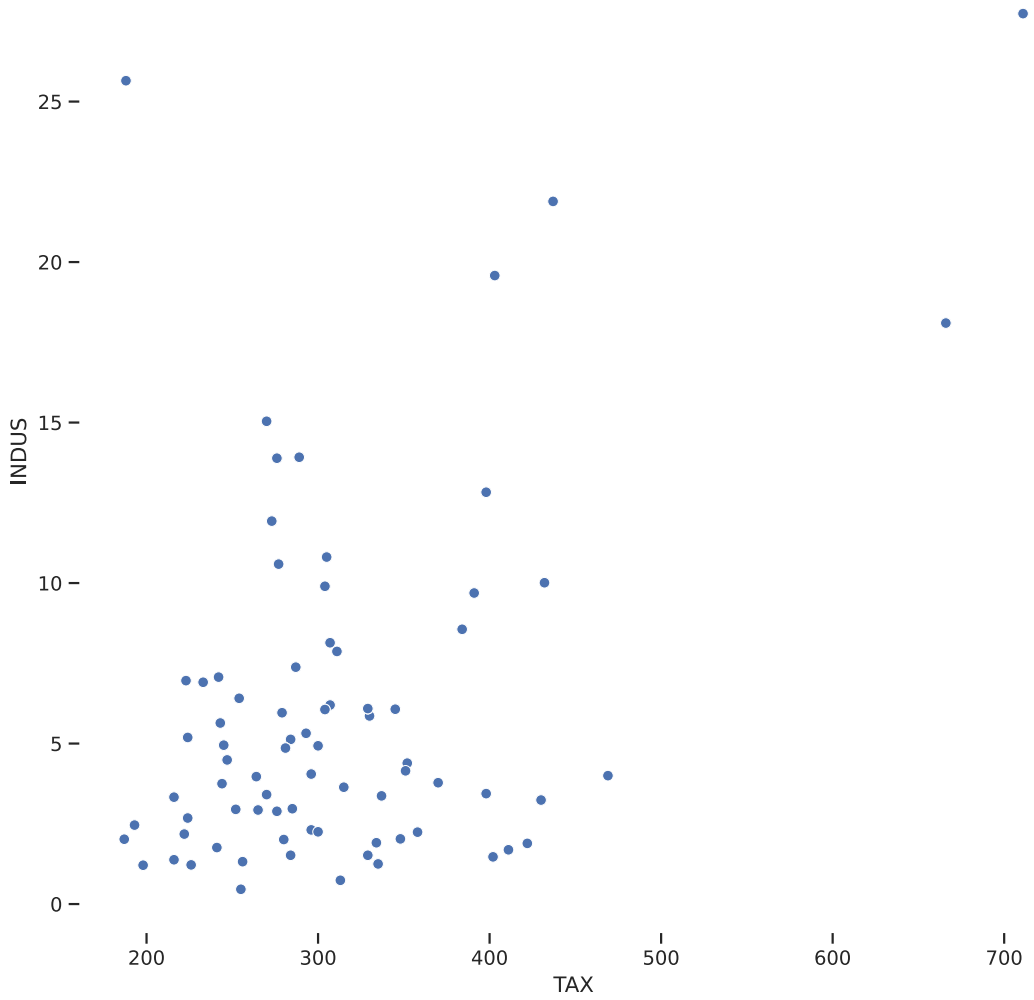
[5]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
CRIM	1.000000	-0.200469	0.406583	-0.055892	0.420972	-0.219247	0.352734
ZN	-0.200469	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537
INDUS	0.406583	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779
CHAS	-0.055892	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518
NOX	0.420972	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470
RM	-0.219247	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265
AGE	0.352734	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000
DIS	-0.379670	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881
RAD	0.625505	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022
TAX	0.582764	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456
PTRATIO	0.289946	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515
B	-0.385064	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534
LSTAT	0.455621	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339

	DIS	RAD	TAX	PTRATIO	B	LSTAT
CRIM	-0.379670	0.625505	0.582764	0.289946	-0.385064	0.455621
ZN	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995
INDUS	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800
CHAS	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929
NOX	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879
RM	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808
AGE	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339
DIS	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996
RAD	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676
TAX	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993
PTRATIO	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044
B	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087
LSTAT	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000

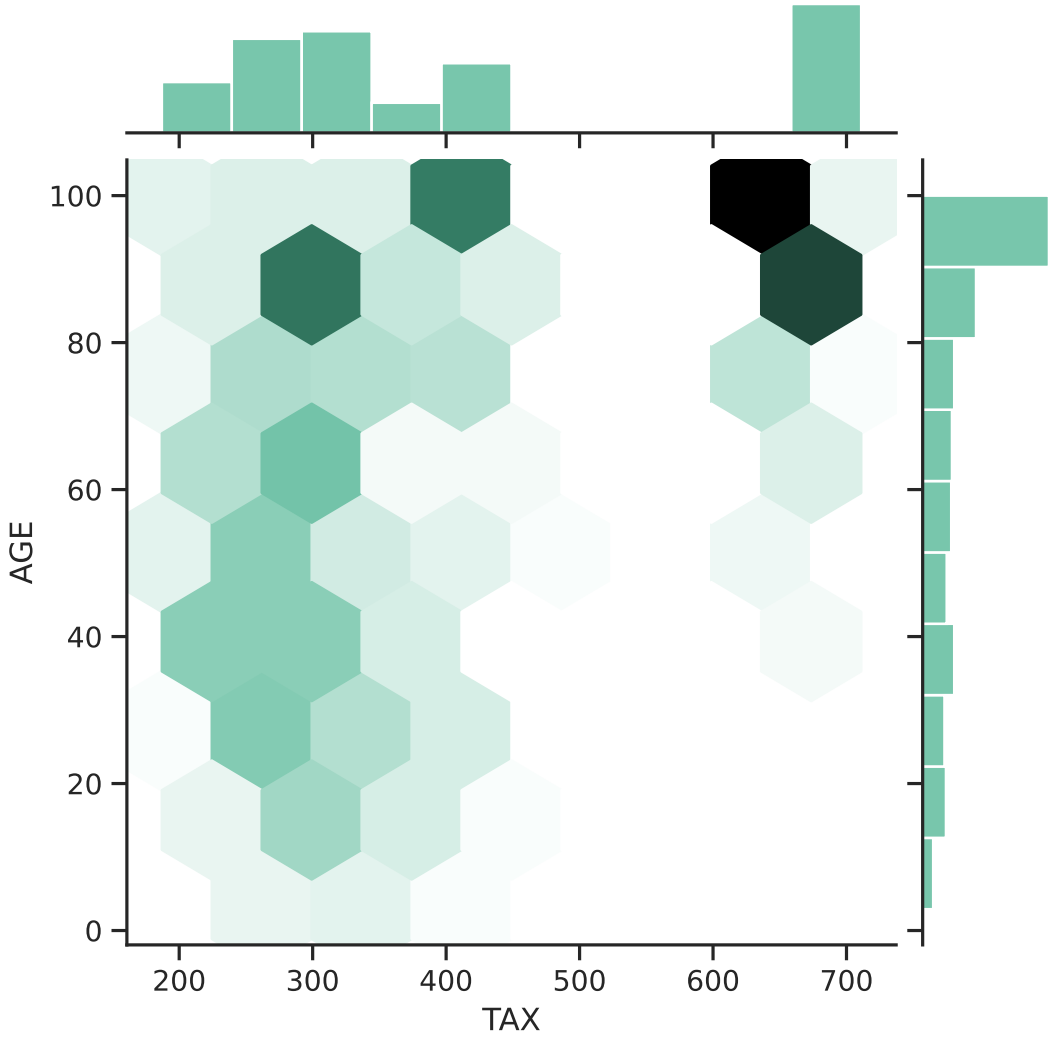
[6]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.despine(fig, left=True, bottom=True)
sns.scatterplot(ax=ax,x='TAX',y='INDUS',data=df)
```

[6]:
```
<AxesSubplot:xlabel='TAX', ylabel='INDUS'>
```



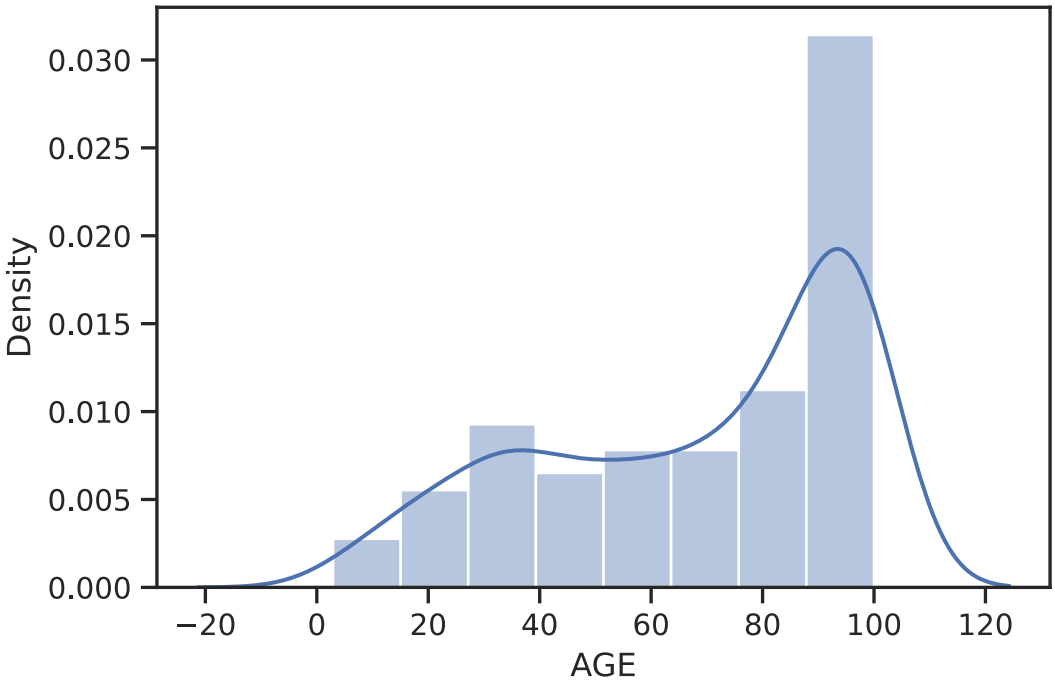
[7]:
```
sns.jointplot(x="TAX", y="AGE",data=df,kind="hex",color="#4CB391")
```

[7]: <seaborn.axisgrid.JointGrid at 0x7f82c471a880>



[8]:
```
sns.distplot(df['AGE'])
```

[8]: <AxesSubplot:xlabel='AGE', ylabel='Density'>


0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
−20
0
20
40
60
80
100
120
AGE