

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



Отчет по лабораторной работе № 1
"Разведочный анализ данных. Исследование и визуализация
данных."

По курсу
«Методы машинного обучения»

Выполнила:
Шаххуд Ф.М.
Студентка группы ИУ5И-12М

Москва, 2020

▼ Цель лабораторной работы

изучая различные графики данных и пытаясь выяснить связь между некоторыми переменными. Мы также установим такие библиотеки, как matplotlib и seaborn.

▼ описание набора данных

В качестве набора данных мы будем использовать набор данных, представленный scikit-learn: стоимость дома в Бостоне. Копия базы данных: <https://archive.ics.uci.edu/ml/machine-learning-databases/housing/>

Этот набор данных имеет характеристики дома и его цену, поэтому прогнозирующая модель может предсказать цену дома в районе Бостона на основе его характеристик.

Атрибуты, которые описывают дом и включены в набор данных как колонки:

- CRIM: уровень преступности на душу населения по городам.
- ZN: доля жилой земли, зонированной для участков более 25 000 кв. Футов.
- INDUS: доля не-розничных бизнес-акров на город.
- CHAS: Фиктивная переменная реки Чарльз (= 1, если тракт ограничивает реку; 0 в противном случае).
- NOX: концентрация оксидов азота (частей на 10 миллионов).
- RM: среднее количество комнат на одно жилище.
- AGE: доля домовладельцев, построенных до 1940 года.
- DIS: взвешивает расстояния до пяти бостонских центров занятости.
- RAD: индекс доступности к радиальным магистралям.
- TAX: ставка налога на полную стоимость имущества за 10 000 долл.
- PTRATIO: Соотношение учеников и учителей по городам.
- B: $1000 (B_k - 0,63)^2$, где B_k - доля чернокожих по городам.
- LSTAT: % ниже статуса населения.
- MEDV: Медианная стоимость домов, занимаемых владельцами, в 1000 долл.

и целевые данные - цена дома, оцененная в [1k долл.]

▼ Импорт библиотек

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

▼ Загрузка данных

мы собираемся загрузить набор данных о ценах на жилье в Бостоне из наборов данных pandas функцию прямой загрузки этого набора данных без внешних файлов.

```
from sklearn.datasets import load_boston
boston=load_boston()
```


▼ Основные характеристики датасета

```
#форма нашего набора данных
boston.data.shape
```

 (506, 13)

Итак, мы видим, что наш набор данных имеет 506 строк данных и 13 объектов или перемен набора данных:

```
for x in boston:
    print(x)
```

 data
target
feature_names
DESCR
filename


Итак, мы видим, что имеем:

1. данные характеристик дома
2. цель, которая является ценой дома
3. Названия элементов, которые являются названиями атрибутов дома.
4. DESCR: описание всего набора данных.
5. имя файла нашего набора данных.

```
#имена элементов столбцов в наборе данных
boston['feature_names']
```

 array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
 'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7')

```
#форма целевого столбца
boston['target'].shape
```

 (506,)

Таким образом, форма цели соответствует форме нашего набора данных. Чтобы увидеть о данных, мы можем отобразить DESCR, который восстановлен внутри набора данных:

```
print(boston['DESCR'])
```



```
.. _boston_dataset:
```

```
Boston house prices dataset
```

```
-----
```

```
**Data Set Characteristics:**
```

```
:Number of Instances: 506
```

```
:Number of Attributes: 13 numeric/categorical predictive. Median Value (at
```

```
:Attribute Information (in order):
```

- CRIM per capita crime rate by town
- ZN proportion of residential land zoned for lots over 25,000 sq
- INDUS proportion of non-retail business acres per town
- CHAS Charles River dummy variable (= 1 if tract bounds river; 0 o
- NOX nitric oxides concentration (parts per 10 million)
- RM average number of rooms per dwelling
- AGE proportion of owner-occupied units built prior to 1940
- DIS weighted distances to five Boston employment centres
- RAD index of accessibility to radial highways
- TAX full-value property-tax rate per \$10,000
- PTRATIO pupil-teacher ratio by town
- B $1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by to
- LSTAT % lower status of the population
- MEDV Median value of owner-occupied homes in \$1000's

```
:Missing Attribute Values: None
```

```
:Creator: Harrison, D. and Rubinfeld, D.L.
```

```
This is a copy of UCI ML housing dataset.
```

```
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
```

```
This dataset was taken from the StatLib library which is maintained at Carnegie
```

```
The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978. Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980. N.B. Various transformations are used in the table on
pages 244-261 of the latter.
```

```
The Boston house-price data has been used in many machine learning papers that
problems.
```

```
.. topic:: References
```

- Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential D
- Quinlan, R. (1993). Combining Instance-Based and Model-Based Learning. In 1

Все атрибуты нашего набора данных записаны, и указано, что в наборе данных нет пропусков. Давайте убедимся, что мы можем видеть:

```
data_tar = pd.DataFrame(data= np.c_[boston['data'], boston['target']],
                        columns= boston['feature_names'].tolist()+['target'])
data=pd.DataFrame(data= np.c_[boston['data']],
                  columns= boston['feature_names'].tolist())
for col in data_tar.columns:
    temp_null_count = data_tar[data_tar[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
CRIM - 0
ZN - 0
INDUS - 0
CHAS - 0
NOX - 0
RM - 0
AGE - 0
DIS - 0
RAD - 0
TAX - 0
PTRATIO - 0
B - 0
LSTAT - 0
target - 0
```

```
# типы данных всех столбцов в наборе данных
data_tar.dtypes
```

```
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
target    float64
dtype: object
```

```
# Первые 5 строк датасета
data_tar.head()
```



	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90

Основные статистические характеристики набора данных
data.describe()



	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
count	506.000000	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	68.574901
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000

Как видно из описания переменных набора данных, стандартное отклонение переменных (посмотрим, имеют ли они небольшое количество значений).

```
for x in data.columns:
    if len(data[x].unique())<3:
        print(x,'The levels are: ')
        print (data[x].unique())
```



CHAS The levels are:
[0. 1.]

Мы видим, что CHAS имеет только два значения, которые равны 0 и 1, мы попытаемся выявить больше

```
for x in data.columns:
    if len(data[x].unique())<10:
        print(x,'The levels are: ')
        print (data[x].unique())
```



```
CHAS The levels are:  
[0. 1.]  
RAD The levels are:  
[ 1.  2.  3.  5.  4.  8.  6.  7. 24.]
```

RAD имеет стандартное значение 8 и имеет только девять значений.

▼ Визуальное исследование датасета

Мы будем использовать две библиотеки, доступные в Python:

- Matplotlib
- Seaborn

```
%matplotlib inline  
sns.set(style="whitegrid")
```

▼ Scatter plot

Сначала мы увидим диаграмму рассеяния для пар данных, чтобы выяснить, существуют ли между ними, используя переменную CHAS в качестве оттенка:

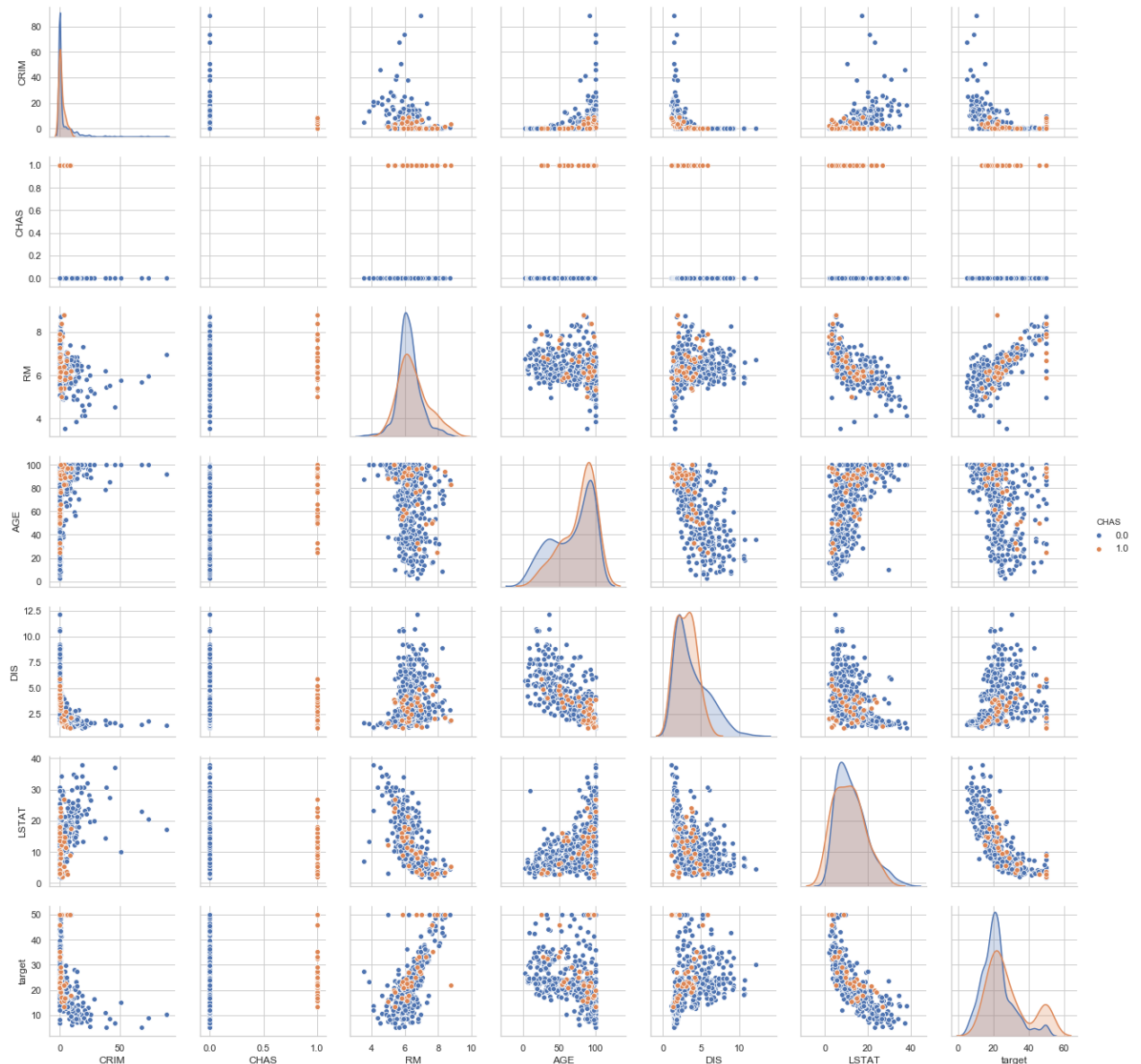
```
sns.pairplot(data_tar[['CRIM', 'CHAS', 'RM', 'AGE', 'DIS', 'LSTAT', 'target']], hue='
```



```

/Users/farahshahhoud/opt/anaconda3/lib/python3.7/site-packages/statsmodels/non
    binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
/Users/farahshahhoud/opt/anaconda3/lib/python3.7/site-packages/statsmodels/non
    FAC1 = 2*(np.pi*bw/RANGE)**2
/Users/farahshahhoud/opt/anaconda3/lib/python3.7/site-packages/statsmodels/non
    binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
/Users/farahshahhoud/opt/anaconda3/lib/python3.7/site-packages/statsmodels/non
    FAC1 = 2*(np.pi*bw/RANGE)**2
<seaborn.axisgrid.PairGrid at 0x120bef190>

```



Из парного графика видно, что существует аппроксимация линейной зависимости между «RM» и столбцом «LSTAT», также аппроксимация экспоненциальной связи между «target» и столбцом «LSTAT». Таким образом, можно посмотреть, сколько данных рядом с ней

```
fig=plt.figure(figsize=(8,8))
ax=fig.gca()
ax.grid()
plt.plot(data_tar['RM'],data_tar['target'],'.r',alpha=0.3)
ax.set_xlabel('RM')
ax.set_ylabel('target')
ax.set_title('relation between "RM" and the house price')
xxx=np.linspace(3.5,9)
yyy=10*xxx-40
plt.plot(xxx,yyy,'b')
```



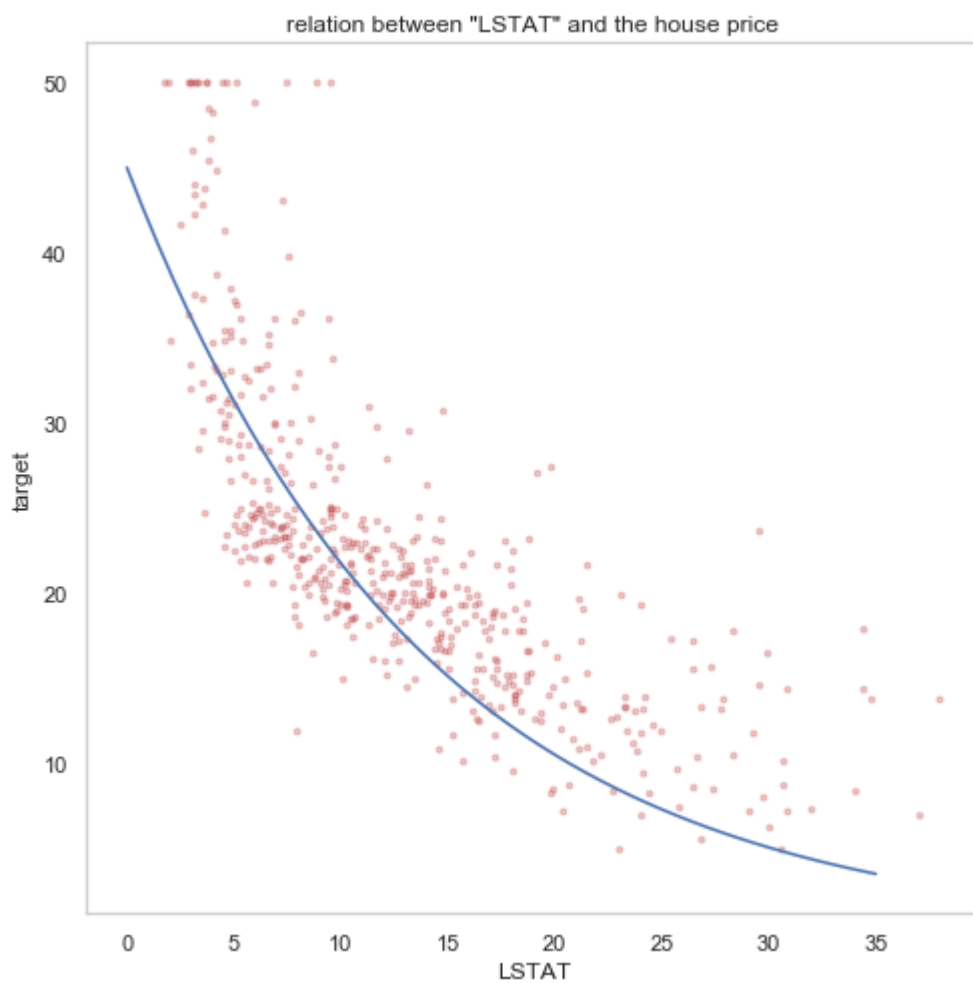
[<matplotlib.lines.Line2D at 0x122c82e10>]



```
fig=plt.figure(figsize=(8,8))
ax=fig.gca()
ax.grid()
plt.plot(data_tar['LSTAT'],data_tar['target'],'.r',alpha=0.3)
ax.set_xlabel('LSTAT')
ax.set_ylabel('target')
ax.set_title('relation between "LSTAT" and the house price')
xxx=np.linspace(0,35)
yyy=45*(0.93**(xxx))
plt.plot(xxx,yyy,'b')
```



[<matplotlib.lines.Line2D at 0x123a384d0>]

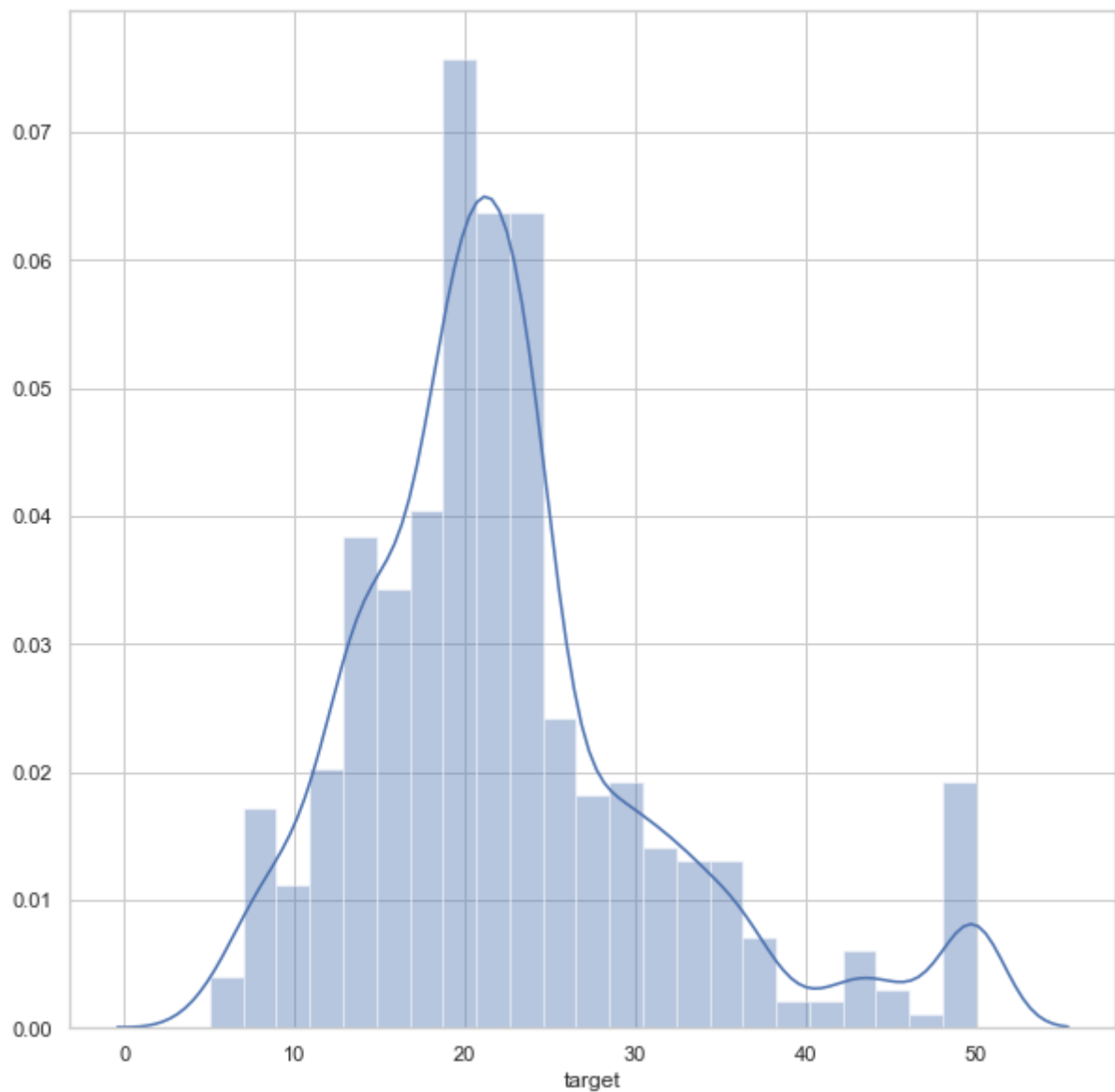


▼ Histogram

```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data_tar['target'])
```



<matplotlib.axes._subplots.AxesSubplot at 0x102375390>



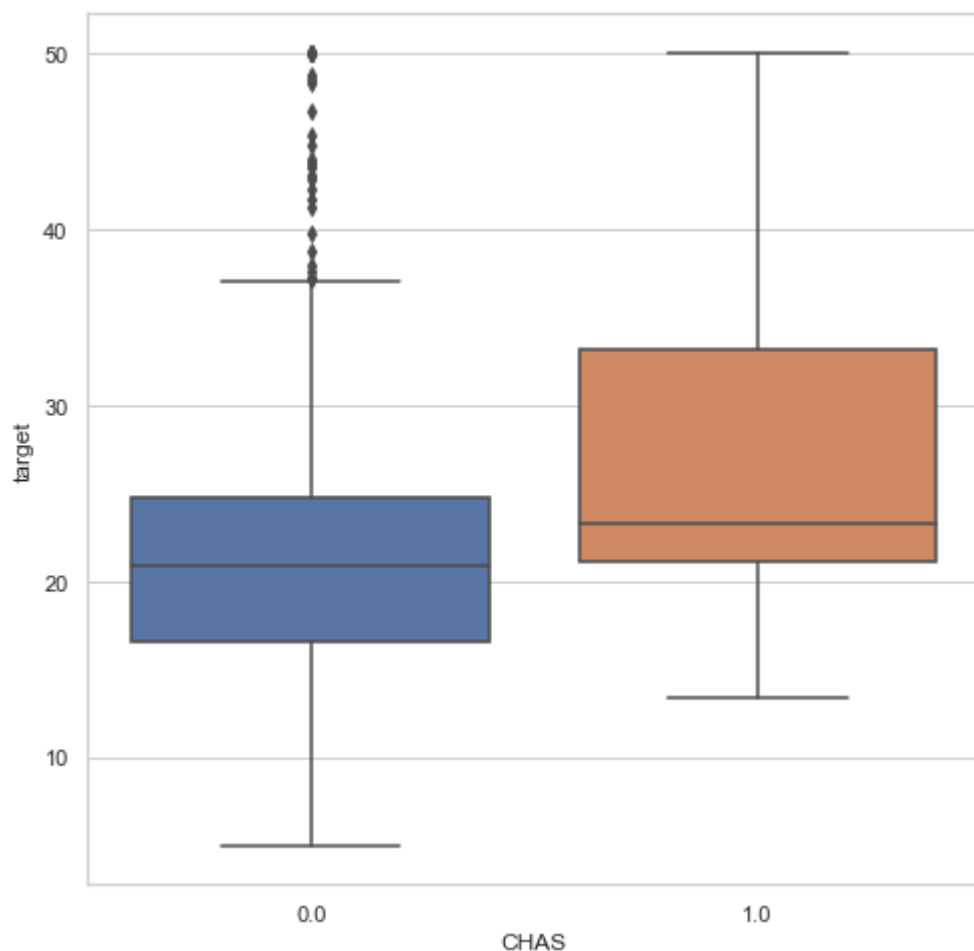
▼ Box Plot

```
fig=plt.figure(figsize=(8,8))
ax=fig.gca()
sns.boxplot(x='CHAS', y='target', data=data, fac_col=0)
```

```
sns.boxplot(x= CHAS ,y= target ,data=data_tar,ax=ax)
```



<matplotlib.axes._subplots.AxesSubplot at 0x123decc10>



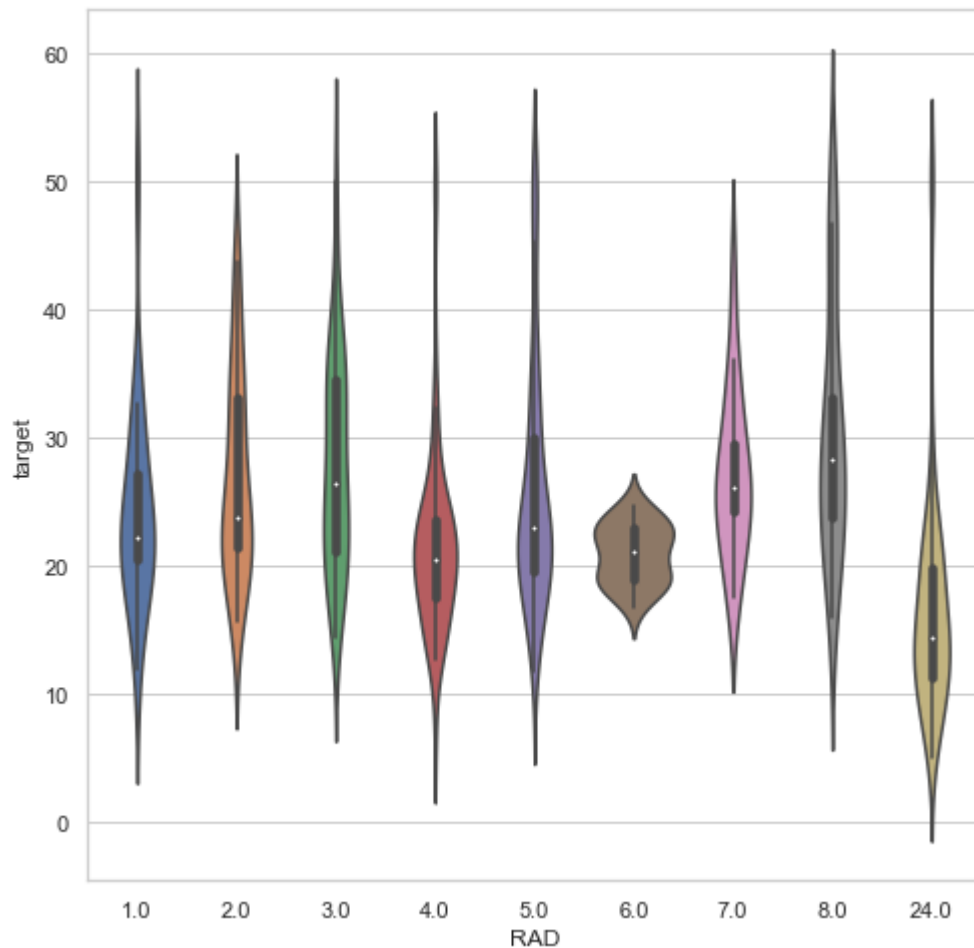
Из коробочного графика мы видим, что средняя цена домов, ограничивающих реку, больше

▼ Violin plot

```
fig=plt.figure(figsize=(8,8))
ax=fig.gca()
sns.violinplot(x='RAD',y='target',data=data_tar,ax=ax)
```



<matplotlib.axes._subplots.AxesSubplot at 0x1241c5410>



Из Violin Plot видно, что цены на дома, которые имеют большой радиус от шоссе, также низ

▼ Информация о корреляции признаков

```
corr_mat=data_tar.corr().round(2)
```

```
corr_mat
```

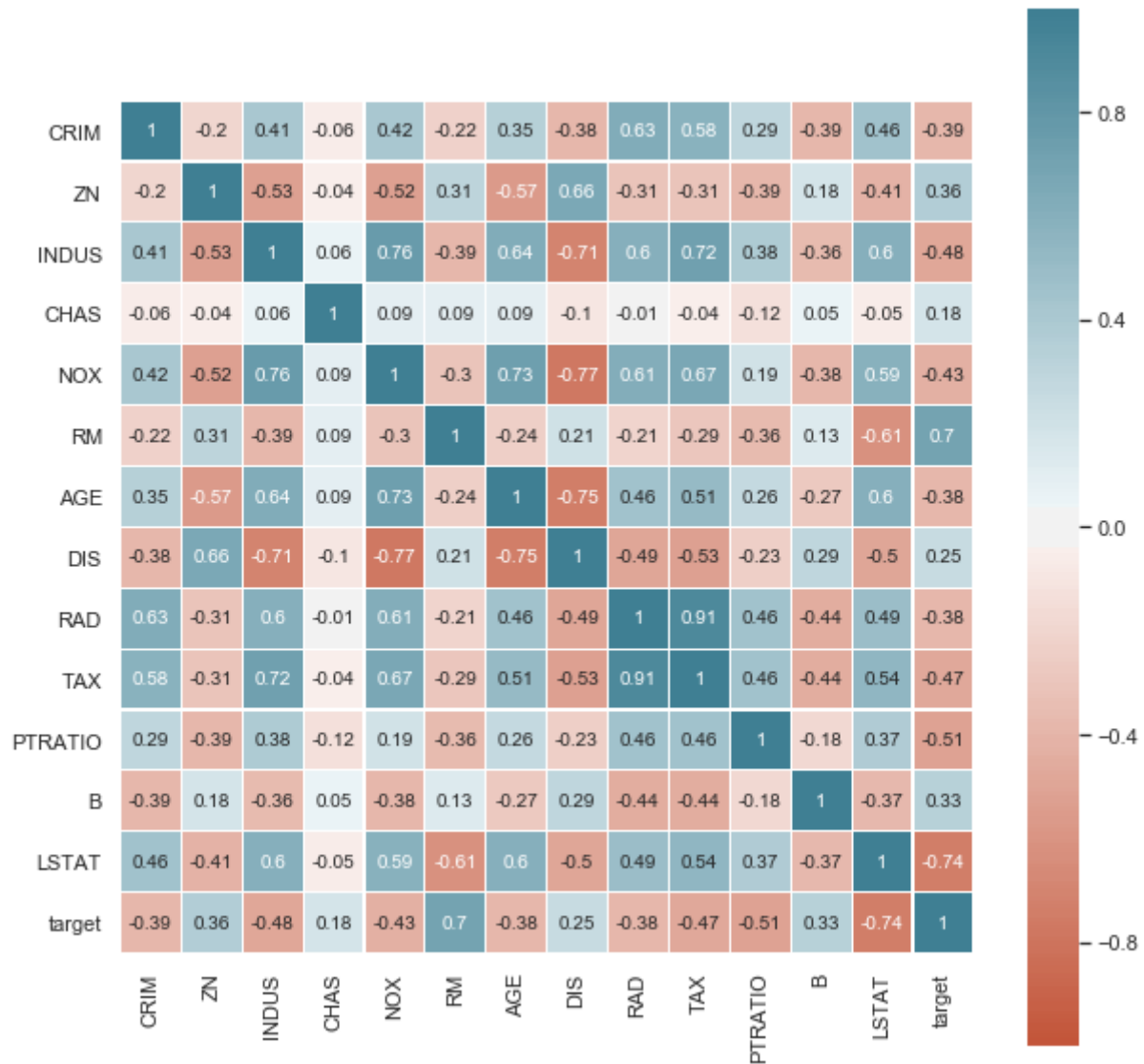


	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
CRIM	1.00	-0.20	0.41	-0.06	0.42	-0.22	0.35	-0.38	0.63	0.58	0.29	-0.39
ZN	-0.20	1.00	-0.53	-0.04	-0.52	0.31	-0.57	0.66	-0.31	-0.31	-0.39	0.18
INDUS	0.41	-0.53	1.00	0.06	0.76	-0.39	0.64	-0.71	0.60	0.72	0.38	-0.36
CHAS	-0.06	-0.04	0.06	1.00	0.09	0.09	0.09	-0.10	-0.01	-0.04	-0.12	0.05
NOX	0.42	-0.52	0.76	0.09	1.00	-0.30	0.73	-0.77	0.61	0.67	0.19	-0.38
RM	-0.22	0.31	-0.39	0.09	-0.30	1.00	-0.24	0.21	-0.21	-0.29	-0.36	0.13
AGE	0.35	-0.57	0.64	0.09	0.73	-0.24	1.00	-0.75	0.46	0.51	0.26	-0.27
DIS	-0.38	0.66	-0.71	-0.10	-0.77	0.21	-0.75	1.00	-0.49	-0.53	-0.23	0.29
RAD	0.63	-0.31	0.60	-0.01	0.61	-0.21	0.46	-0.49	1.00	0.91	0.46	-0.44
TAX	0.58	-0.31	0.72	-0.04	0.67	-0.29	0.51	-0.53	0.91	1.00	0.46	-0.44
PTRATIO	0.29	-0.39	0.38	-0.12	0.19	-0.36	0.26	-0.23	0.46	0.46	1.00	-0.18
B	-0.39	0.18	-0.36	0.05	-0.38	0.13	-0.27	0.29	-0.44	-0.44	-0.18	1.00
LSTAT	0.46	-0.41	0.60	-0.05	0.59	-0.61	0.60	-0.50	0.49	0.54	0.37	-0.37
target	-0.39	0.36	-0.48	0.18	-0.43	0.70	-0.38	0.25	-0.38	-0.47	-0.51	0.36

```
fig, ax=plt.subplots(figsize=(10,10))
sns.heatmap(corr_mat,annot=True,annot_kws={"size": 10},vmin=-1, vmax=1, center=0,
            cmap=sns.diverging_palette(20, 220, n=200),linewidths = 0.1,square=True)
```



<matplotlib.axes._subplots.AxesSubplot at 0x12439be90>



так что видно, что цель имеет хорошую корреляцию с переменными LSTAT и RM.

Также существует высокая корреляция между двумя переменными TAX и RAD, что логично с автомагистралями (центр города), будут облагаться более высокими налогами. Таким образом, чтобы предсказать цены.

Мы также видим, что существует хорошая корреляция между переменной DIS и {NOX, INDUS}.

этого мы можем отказаться от одного из этих вариантов, как это может быть обнаружено с помощью корреляционного анализа. Наконец, мы видим, что, находится ли дом на реке или нет, имеет очень низкую корреляцию с ценой. Поэтому мы можем отказаться от этой переменной при построении сети.

▼ список литературы

1. sklearn.datasets.load_boston, [https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html]
2. The Python Graph Gallery, [<https://python-graph-gallery.com/>]
3. Introduction to Data Visualization in Python, [<https://towardsdatascience.com/introduction-to-data-visualization-in-python-1e1e1e1e1e1e>]
4. Violin plots explained, [<https://towardsdatascience.com/violin-plots-explained-fb1d115e023c>]