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

Лабораторная работа ☒1
по дисциплине
«Методы машинного обучения»
на тему
«Разведочный анализ данных. Исследование и визуализация данных»

Выполнил:
студент группы ИУ5-64Б
Турусов В. И.

Москва — 2019 г.

1. Цель лабораторной работы

Изучить различные методы визуализации данных

2. Задание

Требуется выполнить следующие действия * Выбрать набор данных * Создать ноутбук, который содержить следующие разделы: 1. Текстовое описание выбранного наборы данных 2. Основные характеристики датасета 3. Визуальное исследование датасета 4. Информация о корелляции признаков * Сформировать отчет и разместить его на своем репозитории GitHub

3. Ход выполнения лабораторной работы

3.1. Текстовое описание набора данных

В качестве набора данных используются данные о стоимости домов в Бостоне. Данный датасет содержит следующие колонки: * CRIM уровень преступности на душу населения по городам * ZN доля жилой земли, зонированной на участки свыше 25 000 кв.фут. * INDUS доля акров не-розничного бизнеса в городе * CHAS Фиктивная переменная Чарльза (= 1, Если тракт ограничивает реку; 0 в противном случае) * NOX концентрация оксидов азота (частей на 10 млн.) * RM среднее количество комнат в одном жилом помещении * AGE доля единиц, занятых владельцами, построенных до 1940 года * DIS взвешенные расстояния до пяти бостонских центров занятости * RAD индекс доступности до радиальных магистралей * TAX ставка налога на имущество на полную стоимость за 10 000 долл. США * PTRATIO Соотношение учеников и учителей по городам * B 1000 (Bk - 0,63) ^ 2, где Bk - доля чернокожих по городам * LSTAT % более низкий статус населения * MEDV Средняя стоимость домов, занимаемых владельцами, в 1000 долларов

3.2. Основные характеристики датасета

Подключим необходимые библиотеки

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.display import set_matplotlib_formats
from sklearn.datasets import load_boston
```

Настроим отображение графиков

[2]:
```
%matplotlib inline
sns.set(style='ticks')
set_matplotlib_formats('retina')
```

Зададим ширину текстового представления данных, чтобы в дальнейшем текст в отчете влеза на A4

[3]:
```
pd.set_option('display.width', 70)
```

Загрузим данные и преобразуем их Pandas Dataframe

[4]:
```
def make_dataframe(ds_function):
    ds = ds_function()
    df = pd.DataFrame(data= np.c_[ds['data'], ds['target']],
                      columns= list(ds['feature_names']) + ['target'])
    return df
```

[5]:
```
data = make_dataframe(load_boston)
```

[6]:
```
data.shape
```

[6]:
```
(506, 14)
```

[7]:
```
# Список колонок
data.columns
```

[7]:
```
Index(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS',
       'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT', 'target'],
      dtype='object')
```

[8]:
```
# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[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
```

[9]:
```
data.head()
```

[9]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	\
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	

```
    TAX  PTRATIO       B  LSTAT  target
0  296.0     15.3  396.90   4.98    24.0
1  242.0     17.8  396.90   9.14    21.6
2  242.0     17.8  392.83   4.03    34.7
3  222.0     18.7  394.63   2.94    33.4
4  222.0     18.7  396.90   5.33    36.2
```

[10]:
```
data.dtypes
```

[10]:
```
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
```

Проверим основные статистические характеристики набора данных

[11]:
```
data.describe()
```

[11]:
```
              CRIM          ZN       INDUS        CHAS         NOX  \
count  506.000000  506.000000  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779    0.069170    0.554695
std      8.601545   23.322453    6.860353    0.253994    0.115878
min      0.006320    0.000000    0.460000    0.000000    0.385000
25%      0.082045    0.000000    5.190000    0.000000    0.449000
50%      0.256510    0.000000    9.690000    0.000000    0.538000
75%      3.677083   12.500000   18.100000    0.000000    0.624000
max     88.976200  100.000000   27.740000    1.000000    0.871000

               RM         AGE         DIS         RAD         TAX  \
count  506.000000  506.000000  506.000000  506.000000  506.000000
mean     6.284634   68.574901    3.795043    9.549407  408.237154
std      0.702617   28.148861    2.105710    8.707259  168.537116
min      3.561000    2.900000    1.129600    1.000000  187.000000
25%      5.885500   45.025000    2.100175    4.000000  279.000000
50%      6.208500   77.500000    3.207450    5.000000  330.000000
75%      6.623500   94.075000    5.188425   24.000000  666.000000
max      8.780000  100.000000   12.126500   24.000000  711.000000

          PTRATIO           B       LSTAT      target
count  506.000000  506.000000  506.000000  506.000000
```

mean	18.455534	356.674032	12.653063	22.532806
std	2.164946	91.294864	7.141062	9.197104
min	12.600000	0.320000	1.730000	5.000000
25%	17.400000	375.377500	6.950000	17.025000
50%	19.050000	391.440000	11.360000	21.200000
75%	20.200000	396.225000	16.955000	25.000000
max	22.000000	396.900000	37.970000	50.000000

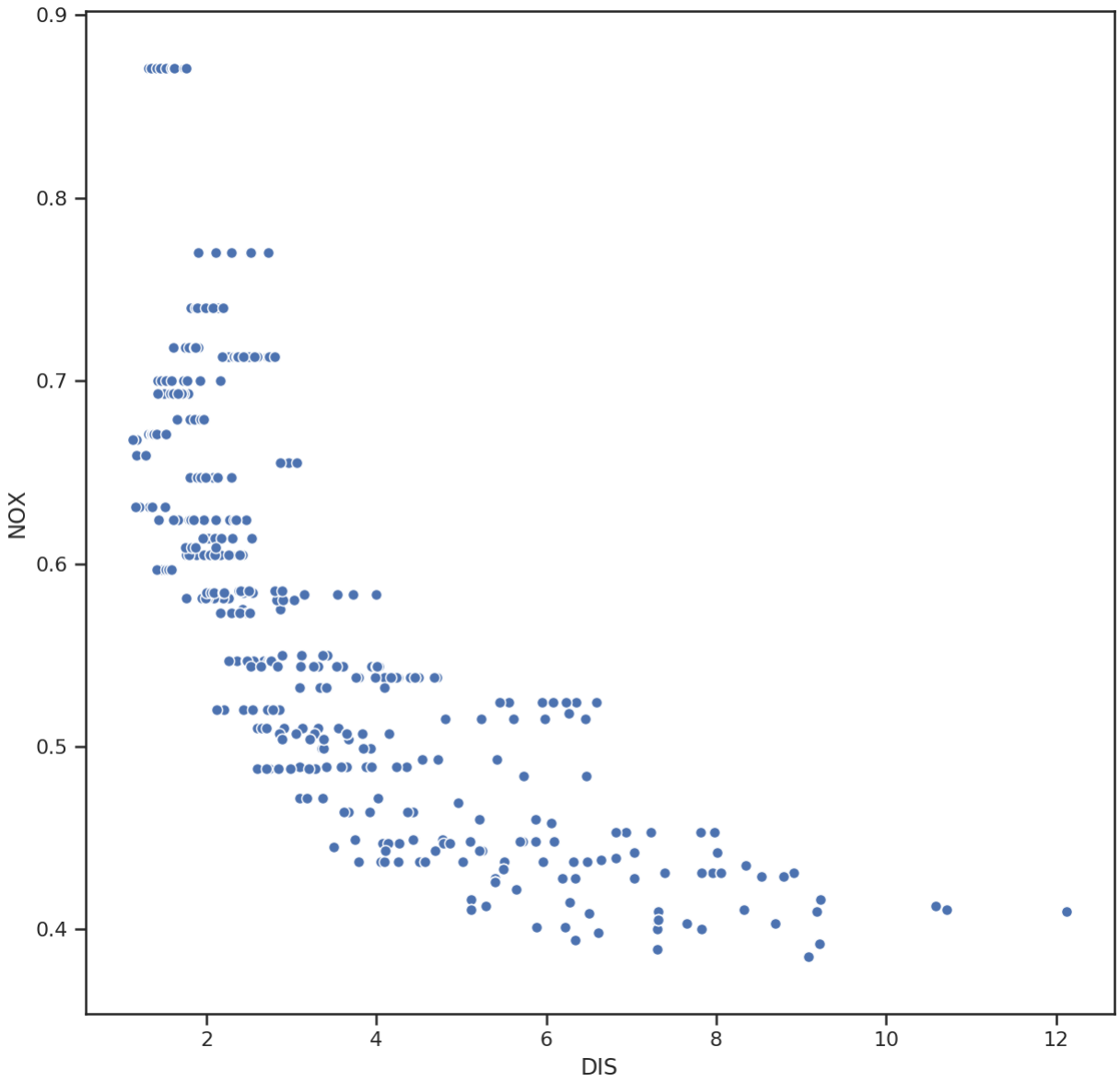
3.3. Визуальное исследование датасета

3.3.1. Диаграмма рассеяния

Позволяет построить распределение двух колонок данных и визуально обнаружить наличие зависимости.

[12]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='DIS', y='NOX', data=data)
```

[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb482897a20>



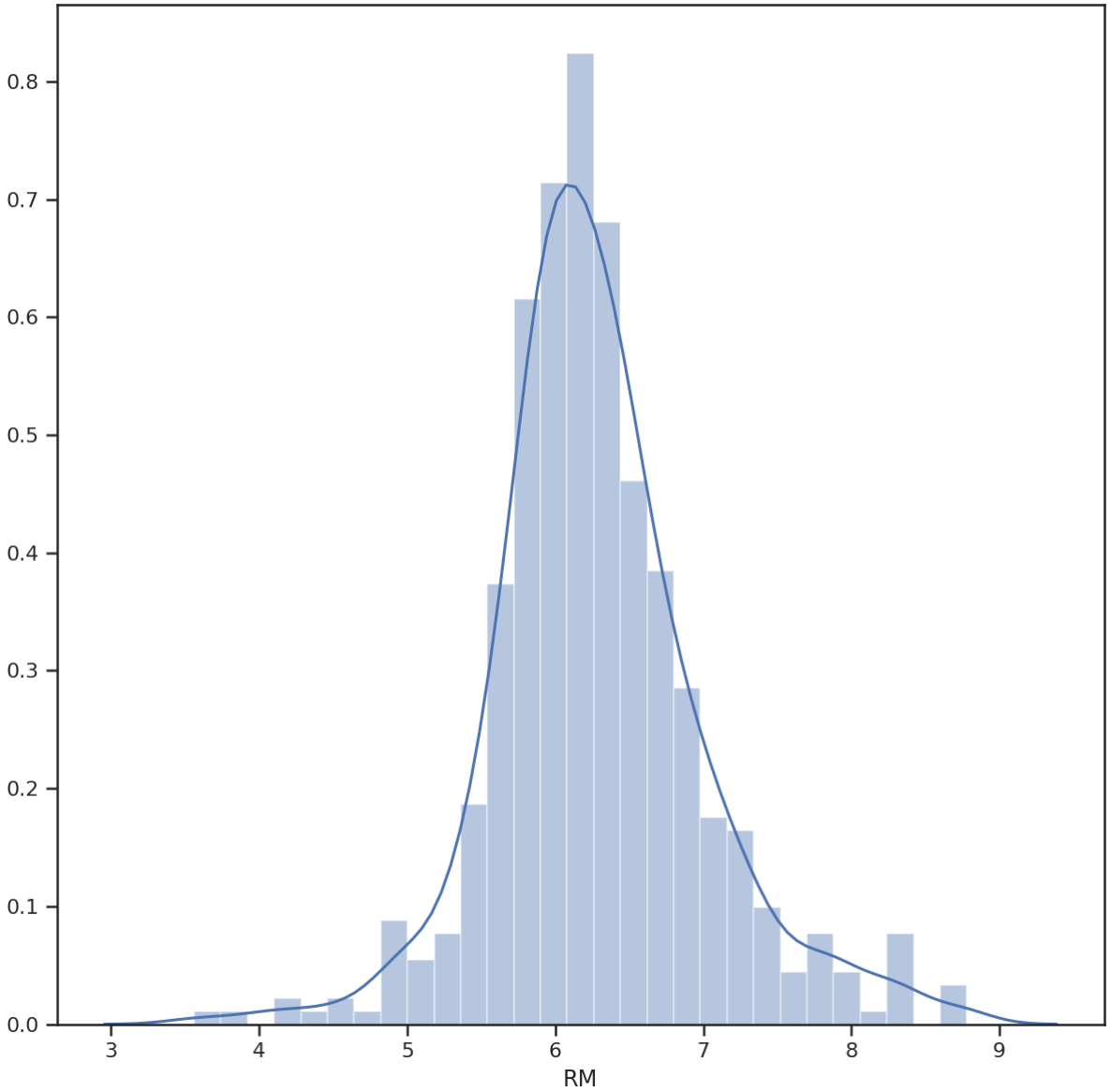
Как видим, чем ближе к бостонским центрам занятости, тем выше концетрация оксидов озота в воздухе

3.3.2. Гистограмма

Позволяет оценить плотность вероятности распределения данных.

[13]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['RM'])
```

[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb480797518>



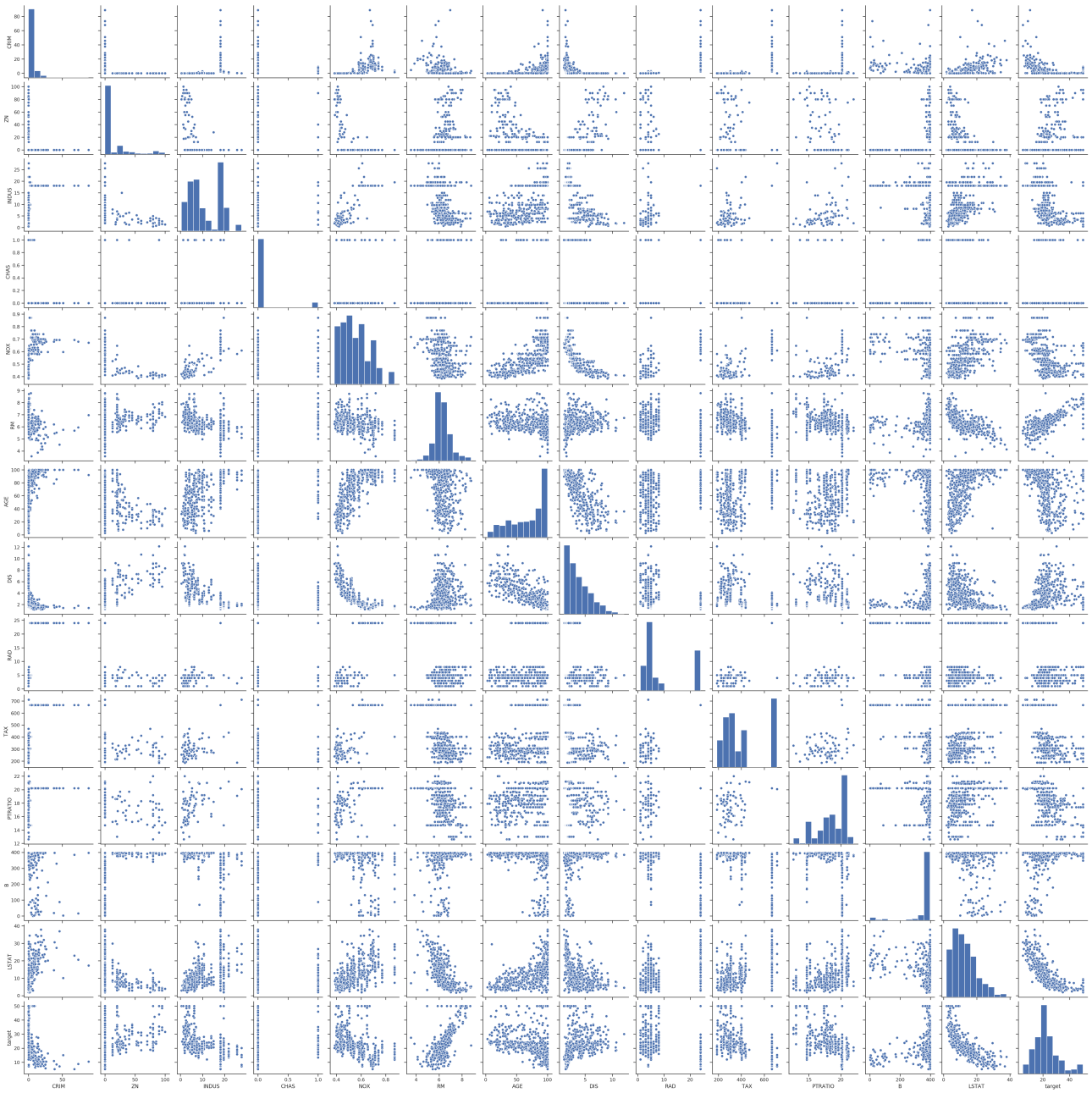
Как видим, что среднее количество комнат в одном жилом помещении в большинстве случаем равно 6

3.3.3. Парные диаграммы

Комбинация гистограмм и диаграмм рассеивания для всего набора данных. Выводится матрица графиков. На пересечении строки и столбца, которые соответствуют двум показателям, строится диаграмма рассеивания. В главной диагонали матрицы строятся гистограммы распределения соответствующих показателей.

[14]:
```
sns.pairplot(data)
```

[14]: <seaborn.axisgrid.PairGrid at 0x7fb4807e3208>



3.4. Информация о корелляции признаков

Построим корреляционную матрицу по всему набору данных
Проверка корреляции признаков позволяет решить две задачи:

- Понять какие признаки (колонки датасета) наиболее сильно коррелируют с целевым признаком (в нашем примере это колонка “RM”).Именно эти признаки будут наиболее информативными для моделей машинного обучения. Признаки, которые слабо коррелируют с целевым признаком, можно попробовать исключить из построения модели, иногда это повышает качество модели. Нужно отметить, что некоторые алгоритмы машинного обучения автоматически определяют ценность того или иного признака для построения модели.

- Понять какие нецелевые признаки линейно зависимы между собой. Линейно зависимые признаки, как правило, очень плохо влияют на качество моделей. Поэтому если несколько признаков линейно зависимы, то для построения модели из них выбирают какой-то один признак.

[15]:
```
data.corr()
```

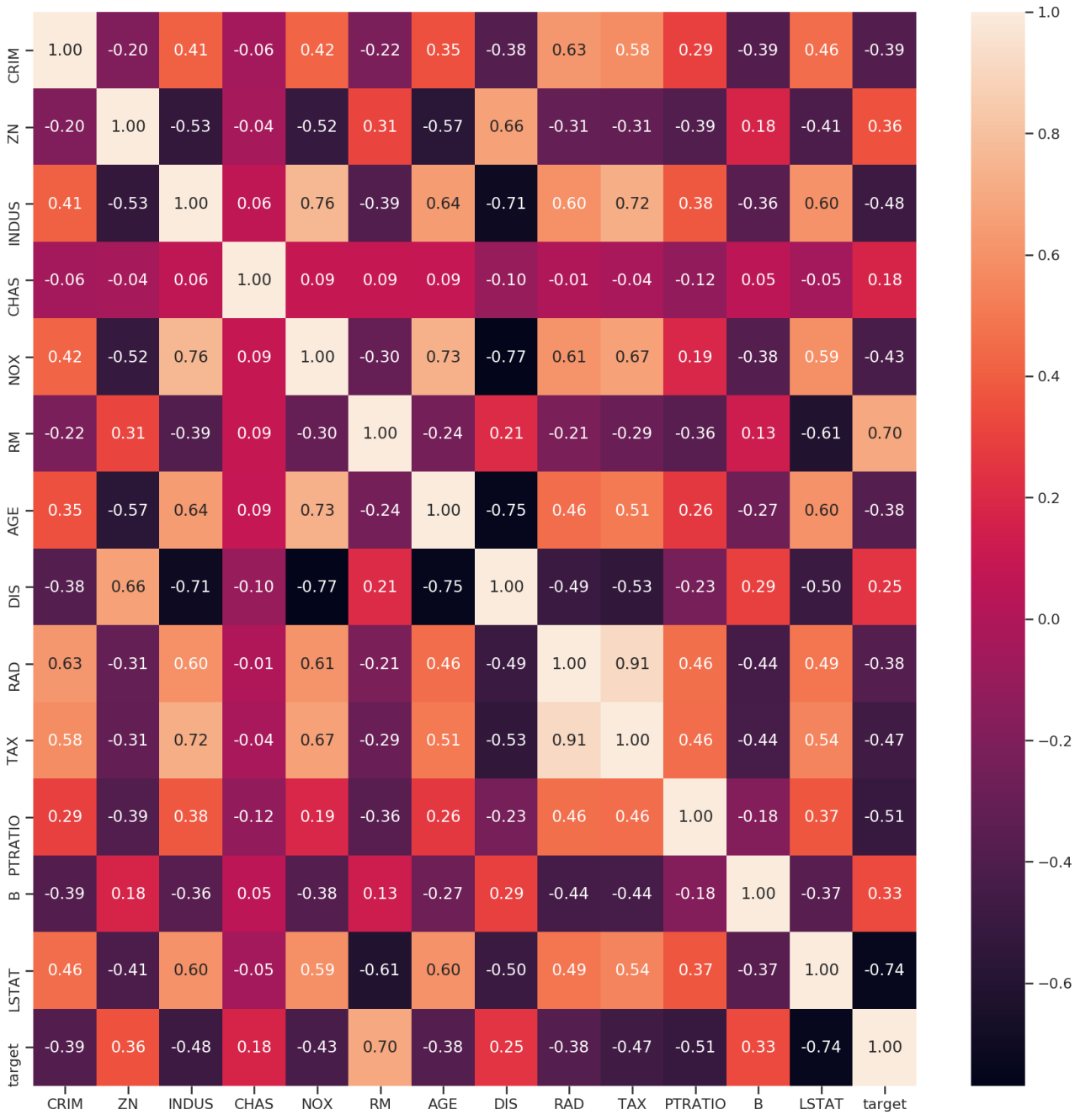
[15]:
```
             CRIM       ZN     INDUS      CHAS       NOX        RM  \
CRIM     1.000000 -0.200469  0.406583 -0.055892  0.420972 -0.219247
ZN      -0.200469  1.000000 -0.533828 -0.042697 -0.516604  0.311991
INDUS    0.406583 -0.533828  1.000000  0.062938  0.763651 -0.391676
CHAS    -0.055892 -0.042697  0.062938  1.000000  0.091203  0.091251
NOX      0.420972 -0.516604  0.763651  0.091203  1.000000 -0.302188
RM      -0.219247  0.311991 -0.391676  0.091251 -0.302188  1.000000
AGE      0.352734 -0.569537  0.644779  0.086518  0.731470 -0.240265
DIS     -0.379670  0.664408 -0.708027 -0.099176 -0.769230  0.205246
RAD      0.625505 -0.311948  0.595129 -0.007368  0.611441 -0.209847
TAX      0.582764 -0.314563  0.720760 -0.035587  0.668023 -0.292048
PTRATIO  0.289946 -0.391679  0.383248 -0.121515  0.188933 -0.355501
B       -0.385064  0.175520 -0.356977  0.048788 -0.380051  0.128069
LSTAT    0.455621 -0.412995  0.603800 -0.053929  0.590879 -0.613808
target  -0.388305  0.360445 -0.483725  0.175260 -0.427321  0.695360

              AGE       DIS       RAD       TAX   PTRATIO         B  \
CRIM     0.352734 -0.379670  0.625505  0.582764  0.289946 -0.385064
ZN      -0.569537  0.664408 -0.311948 -0.314563 -0.391679  0.175520
INDUS    0.644779 -0.708027  0.595129  0.720760  0.383248 -0.356977
CHAS     0.086518 -0.099176 -0.007368 -0.035587 -0.121515  0.048788
NOX      0.731470 -0.769230  0.611441  0.668023  0.188933 -0.380051
RM      -0.240265  0.205246 -0.209847 -0.292048 -0.355501  0.128069
AGE      1.000000 -0.747881  0.456022  0.506456  0.261515 -0.273534
DIS     -0.747881  1.000000 -0.494588 -0.534432 -0.232471  0.291512
RAD      0.456022 -0.494588  1.000000  0.910228  0.464741 -0.444413
TAX      0.506456 -0.534432  0.910228  1.000000  0.460853 -0.441808
PTRATIO  0.261515 -0.232471  0.464741  0.460853  1.000000 -0.177383
B       -0.273534  0.291512 -0.444413 -0.441808 -0.177383  1.000000
LSTAT    0.602339 -0.496996  0.488676  0.543993  0.374044 -0.366087
target  -0.376955  0.249929 -0.381626 -0.468536 -0.507787  0.333461
```

```
             LSTAT    target
CRIM      0.455621 -0.388305
ZN       -0.412995  0.360445
INDUS     0.603800 -0.483725
CHAS     -0.053929  0.175260
NOX       0.590879 -0.427321
RM       -0.613808  0.695360
AGE       0.602339 -0.376955
DIS      -0.496996  0.249929
RAD       0.488676 -0.381626
TAX       0.543993 -0.468536
PTRATIO   0.374044 -0.507787
B        -0.366087  0.333461
LSTAT     1.000000 -0.737663
target   -0.737663  1.000000
```

[16]:
```
# Визуализируем корреляционную матрицу с помощью тепловой карты
fig, ax = plt.subplots(figsize=(15,15))
sns.heatmap(data.corr(), annot=True, fmt='.2f')
```

[16]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb46e9500f0>


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 0.46 -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 -0.41 0.36
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 0.60 -0.48
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 -0.05 0.18
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 0.59 -0.43
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 -0.61 0.70
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 0.60 -0.38
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 -0.50 0.25
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 0.49 -0.38
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 0.54 -0.47
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 0.37 -0.51
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 -0.37 0.33
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 1.00 -0.74
target -0.39 0.36 -0.48 0.18 -0.43 0.70 -0.38 0.25 -0.38 -0.47 -0.51 0.33 -0.74 1.00
CRIM ZN INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO B LSTAT target
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6