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 ИУ5-23М
Богомолов Д.Н.

Москва — 2020 г.

1. Цель лабораторной работы

Изучить различные методы визуализации данных [1].

2. Задание

Требуется выполнить следующие действия [1]:

- Выбрать набор данных (датасет).
- Создать ноутбук, который содержит следующие разделы:
  1. Текстовое описание выбранного набора данных.
  2. Основные характеристики датасета.
  3. Визуальное исследование датасета.
  4. Информация о корреляции признаков.
- Сформировать отчет и разместить его в своем репозитории на GitHub.

3. Ход выполнения работы

3.1. Текстовое описание набора данных

В качестве набора данных используется список с предсказанием цены квартиры в зависимости от ее района. Данный набор данных доступен по следующему адресу: `https://scikit-learn.org/0.20/modules/generated/sklearn.datasets.load_boston.html`.

Данный файл содержит следующие колонки:

- **CRIM** per capita crime rate by town
- **ZN** proportion of residential land zoned for lots over 25,000 sq.ft.
- **INDUS** proportion of non-retail business acres per town
- **CHAS** Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- **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(Bk - 0.63)^2
- **LSTAT** % lower status of the population
- **MEDV** Median value of owner-occupied homes in $1000's

3.2. Основные характеристики набора данных

Подключим все необходимые библиотеки:

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from sklearn.datasets import *
data = load_boston()
%matplotlib inline
sns.set(style="ticks")
```

Выведем размер датасета:

```
In [2]: X, y = load_boston(return_X_y=True)
print(X.shape)
```

Создание Pandas Dataframe

```
In [3]: 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
```

Проверим полученные типы:

```
In [6]: data.dtypes #Типы данных каждого атрибута
Out[6]: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
```

Посмотрим на данные в данном наборе данных:

```
In [7]: data.head()
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	tar
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	4.98	
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	9.14	
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	4.03	
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	2.94	
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	5.33	

Проверим датасет на пустые значения:

```
In [8]: for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

Вывод описательных статистик по каждому атрибуту:

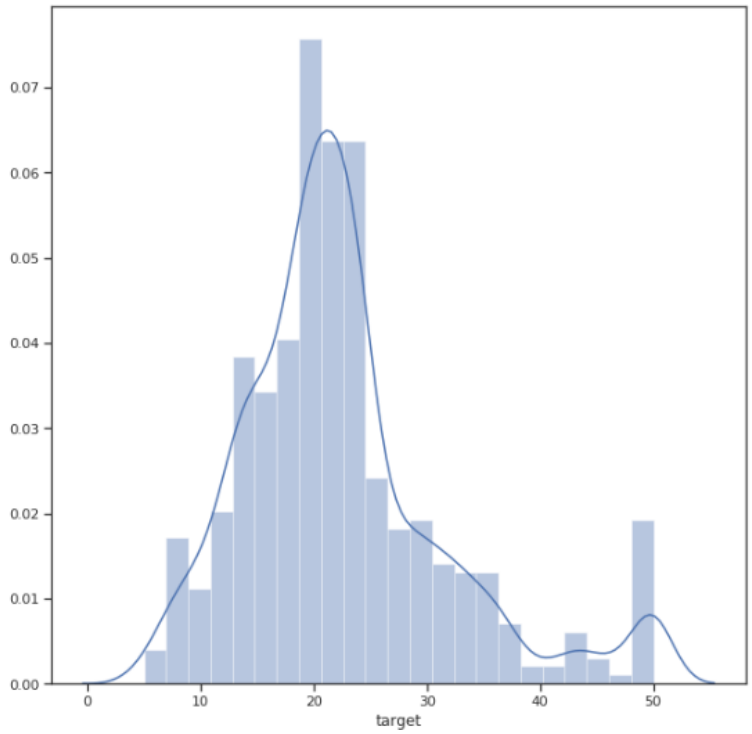
```
In [9]: data.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	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	3.795043	9.549407	408.237154	18.455534	356.674032	12.653063
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	7.141062
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	6.950000
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	11.360000
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	16.955000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	37.970000

3.3. Визуальное исследование датасета

Оценим распределение целевого признака — мощности солнечного излучения:

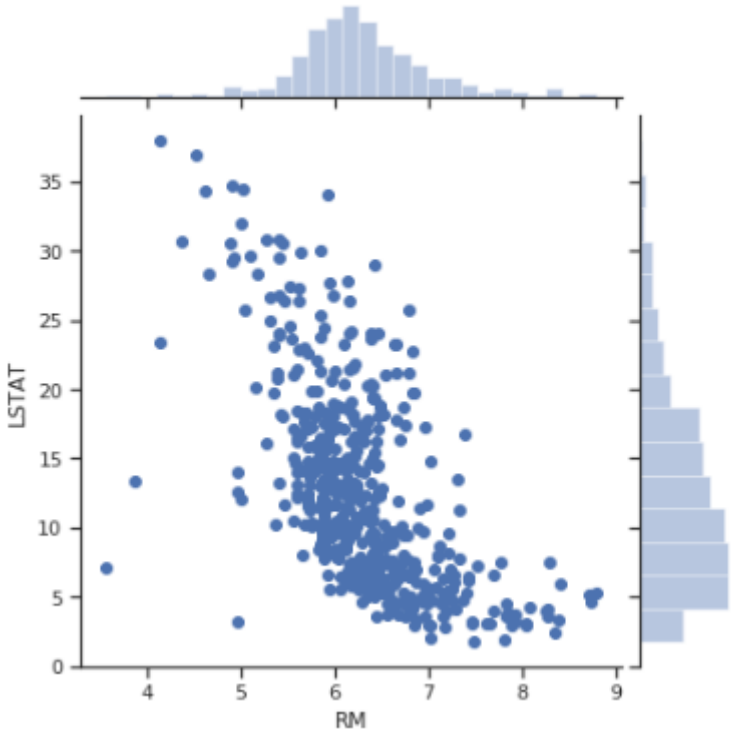
```
In [13]: fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['target'])
```



Посмотрим зависимость параметра LSTAT от параметра RM:

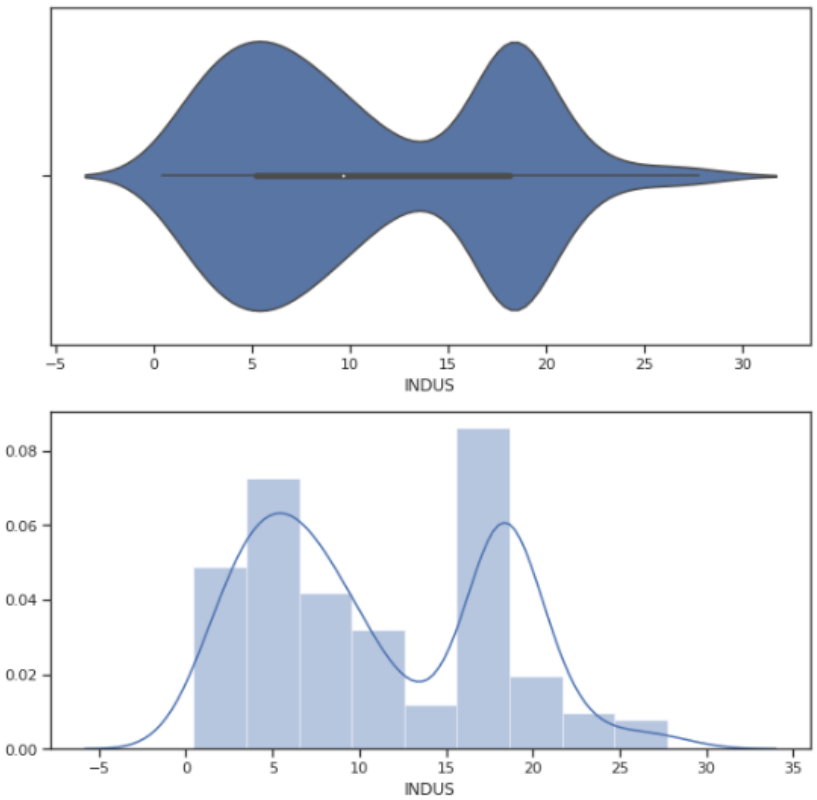
```
In [14]: sns.jointplot(x='RM', y='LSTAT', data=data)
```

```
<seaborn.axisgrid.JointGrid at 0x7f39eb019320>
```



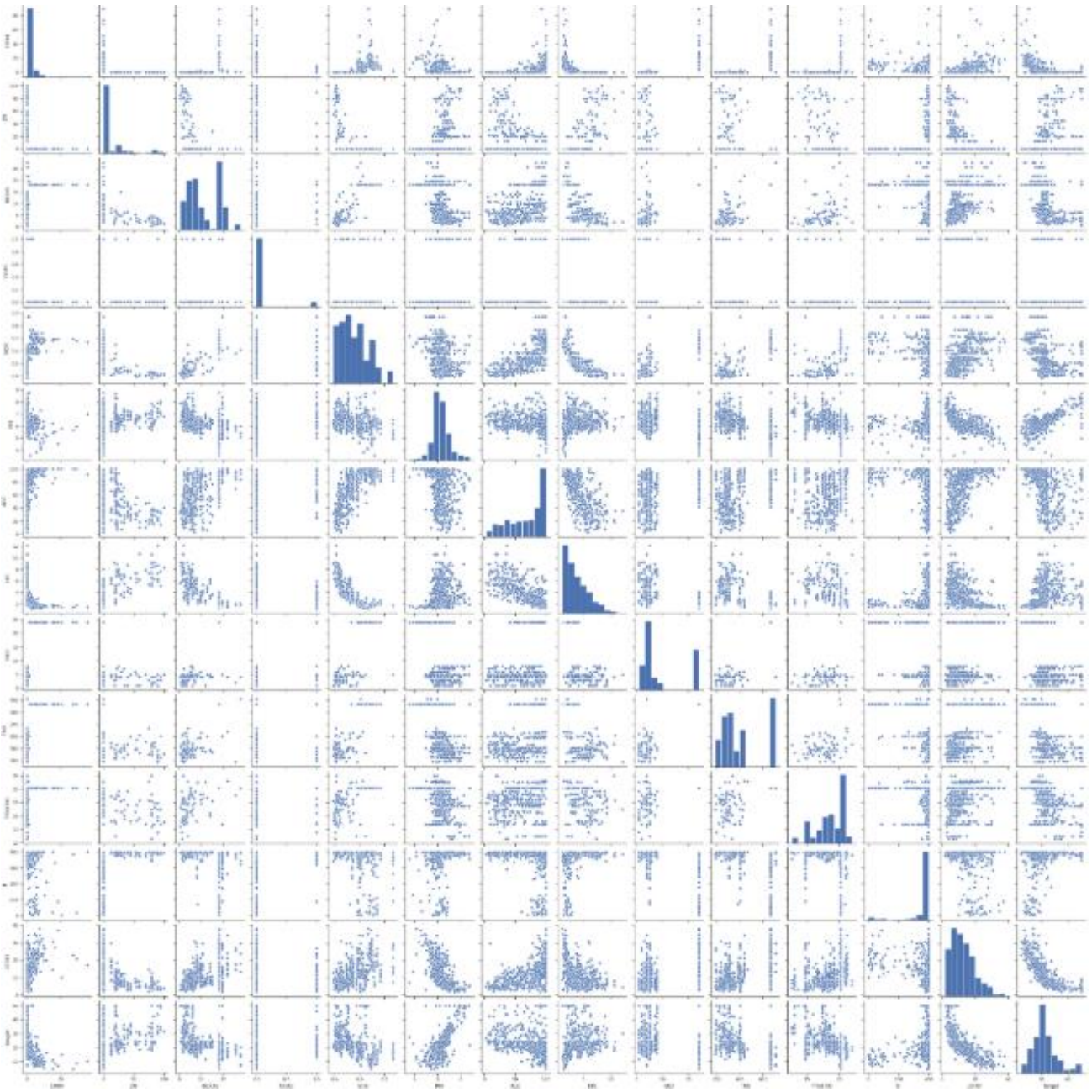
Посмотрим на распределение мощности излучения в течение дня:

```
In [15]: fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['INDUS'])
sns.distplot(data['INDUS'], ax=ax[1])
```



Построим парные диаграммы по всем показателям по исходному набору данных:

```
In [16]: sns.pairplot(data)
```

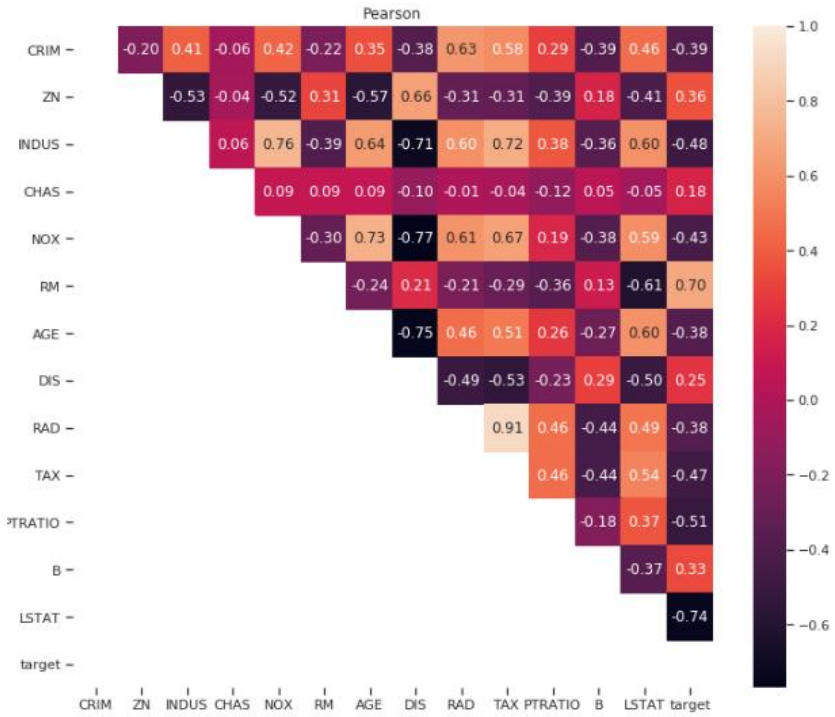


3.4. Информация о корреляции признаков

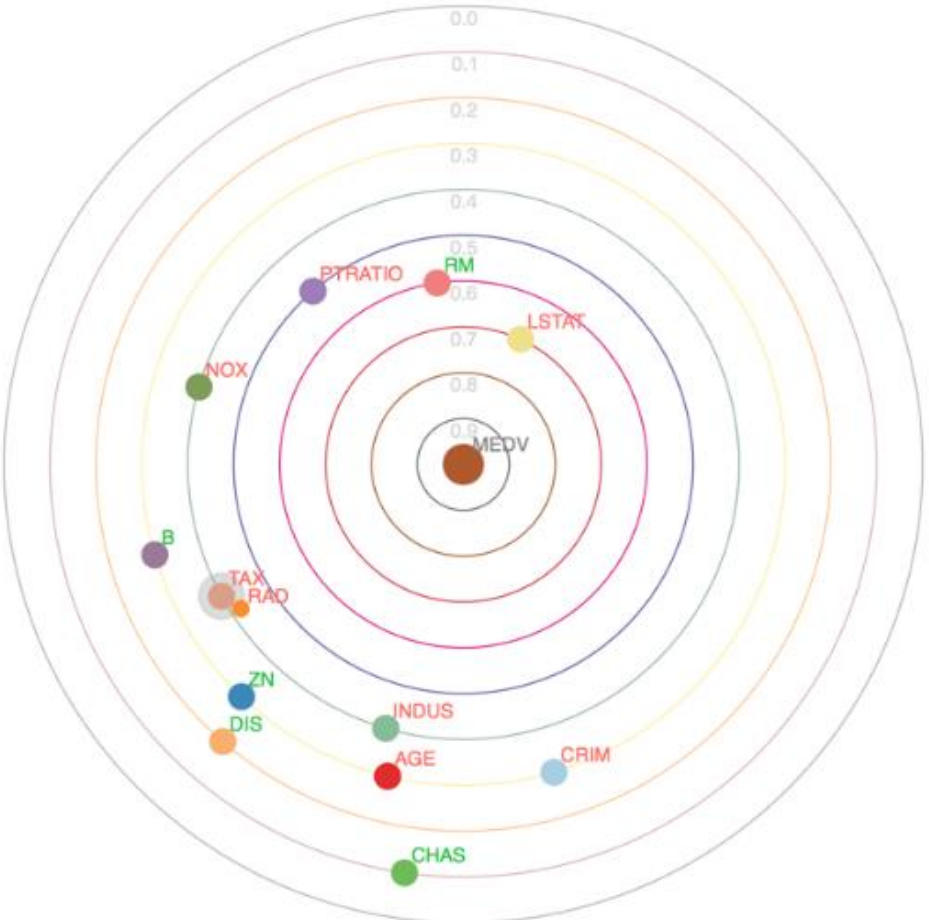
Проведём корреляционный анализ тремя методами:

```
In [17]: mask = np.zeros_like(data.corr(), dtype=np.bool)
mask[np.tril_indices_from(mask)] = True

fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(40,10))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], mask=mask, annot=True, f
mt='.2f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], mask=mask, annot=True, f
mt='.2f')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], mask=mask, annot=True,
fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```



Для более наглядного отображения корреляции воспользуемся библиотекой Solar correlation map :



Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Разведочный анализ данных. Исследование и визуализация данных» [Электронный ресурс] // GitHub. – 2019. – Режим доступа: https://github.com/ugapanyuk/ml_course/wiki/LAB_EDA_VISUALIZATION (дата обращения: 15.02.2020).

[2] Scikit-learn. Boston house-prices dataset [Electronic resource] // Scikit-learn. — 2018. — Access mode: https://scikit-learn.org/0.20/modules/generated/sklearn.datasets.load_boston.html (online; accessed: 18.02.2020).

[3] Team The IPython Development. IPython 7.3.0 Documentation [Electronic resource] // Read the Docs. — 2019. — Access mode: https://ipython.readthedocs.io/en/stable/ (online; accessed: 20.02.2020).

[4] Waskom M. seaborn 0.9.0 documentation [Electronic resource] // PyData. — 2018. — Access mode: https://seaborn.pydata.org/ (online; accessed: 20.02.2020).

[5] pandas 0.24.1 documentation [Electronic resource] // PyData. — 2019. — Access mode: http://pandas.pydata.org/pandas-docs/stable/ (online; accessed: 20.02.2020).