

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



Отчёт

“Методы машинного обучения”

Лабораторная работа № 1

**“Разведочный анализ данных. Исследование и
визуализация данных”**

ИСПОЛНИТЕЛЬ:

Студент группы ИУ5-21М

Коростелёв В.М. _____

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е. _____

Москва – 2019

Цель работы

Изучить различные методы визуализации данных.

Задание

Требуется выполнить следующие действия:

- 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.

Ход выполнения работы

Текстовое описание набора данных

В качестве набора данных используется датасет с информацией об автомобилях от 1985-го года. Набор данных доступен по ссылке: kaggle.com/fazilbtopal/auto85.

Набор данных состоит из одного файла `auto.csv`, содержащего все данные датасета. Данный файл содержит следующие колонки:

- Марка авто
- Тип топлива
- Тип двигателя
- Количество дверей
- Тип кузова
- Тип привода
- Расположение двигателя
- Расположение колесной базы
- Длина кузова
- Ширина кузова
- Высота кузова
- Топливная система
- Размер двигателя
- Количество цилиндров
- Компрессия
- Цена
- Скорость
- Количество лошадиных сил

Основные характеристики набора данных

Подключим все необходимые библиотеки:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

Загрузим данные:

```
data = pd.read_csv('auto.csv', sep=",")
```

Первые 5 строк датасета:

```
In [5]: data.head()
```

Out[5]:

	symboling	normalized-losses	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg
0	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	111.0	5000.0	21	27
1	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	111.0	5000.0	21	27
2	1	122	alfa-romero	std	two	hatchback	rwd	front	94.5	0.822681	...	9.0	154.0	5000.0	19	26
3	2	164	audi	std	four	sedan	fwd	front	99.8	0.848630	...	10.0	102.0	5500.0	24	30
4	2	164	audi	std	four	sedan	4wd	front	99.4	0.848630	...	8.0	115.0	5500.0	18	22

5 rows × 29 columns

Размер датасета (строк, столбцов):

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

Out[6]: (201, 29)

Список колонок с типами данных:

```
In [9]: data.dtypes
```

```
Out[9]: symboling          int64
normalized-losses      int64
make                   object
aspiration             object
num-of-doors          object
body-style             object
drive-wheels          object
engine-location        object
wheel-base           float64
length                float64
width                 float64
height                float64
curb-weight           int64
engine-type           object
num-of-cylinders       object
engine-size           int64
fuel-system           object
bore                  float64
stroke                float64
compression-ratio      float64
horsepower            float64
peak-rpm              float64
city-mpg              int64
highway-mpg           int64
price                 float64
city-L/100km          float64
horsepower-binned     object
diesel                int64
gas                   int64
dtype: object
```

Основные статистические характеристики набора данных:

```
In [10]: data.describe()
```

```
Out[10]:
```

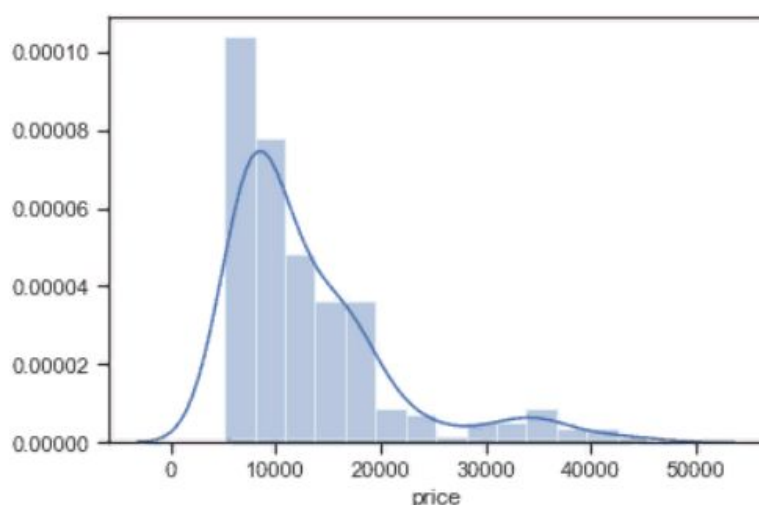
	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower
count	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	197.000000	201.000000	201.000000
mean	0.840796	122.000000	98.797015	0.837102	0.915126	53.766667	2555.666667	126.875622	3.330692	3.256904	10.164279	103.405534
std	1.254802	31.99625	6.066366	0.059213	0.029187	2.447822	517.296727	41.546834	0.268072	0.319256	4.004965	37.365700
min	-2.000000	65.000000	86.600000	0.678039	0.837500	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000000	48.000000
25%	0.000000	101.000000	94.500000	0.801538	0.890278	52.000000	2169.000000	98.000000	3.150000	3.110000	8.600000	70.000000
50%	1.000000	122.000000	97.000000	0.832292	0.909722	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000000	95.000000
75%	2.000000	137.000000	102.400000	0.881788	0.925000	55.500000	2926.000000	141.000000	3.580000	3.410000	9.400000	116.000000
max	3.000000	256.000000	120.900000	1.000000	1.000000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000000	262.000000

Визуальное исследование датасета

Оценим распределение целевого признака - стоимость:

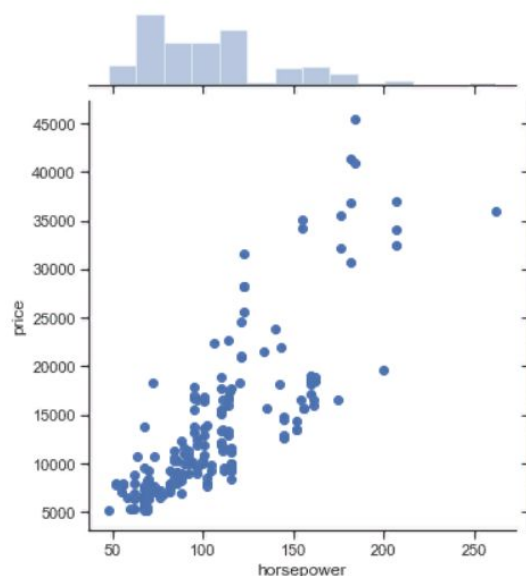
```
In [13]: sns.distplot(data['price'])
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x19852ea9860>
```



Видно, что имеется большой перевес в сторону автомобилей стоимостью 5-10 тысяч долларов. Оценим, насколько цена зависит от количества лошадиных сил:

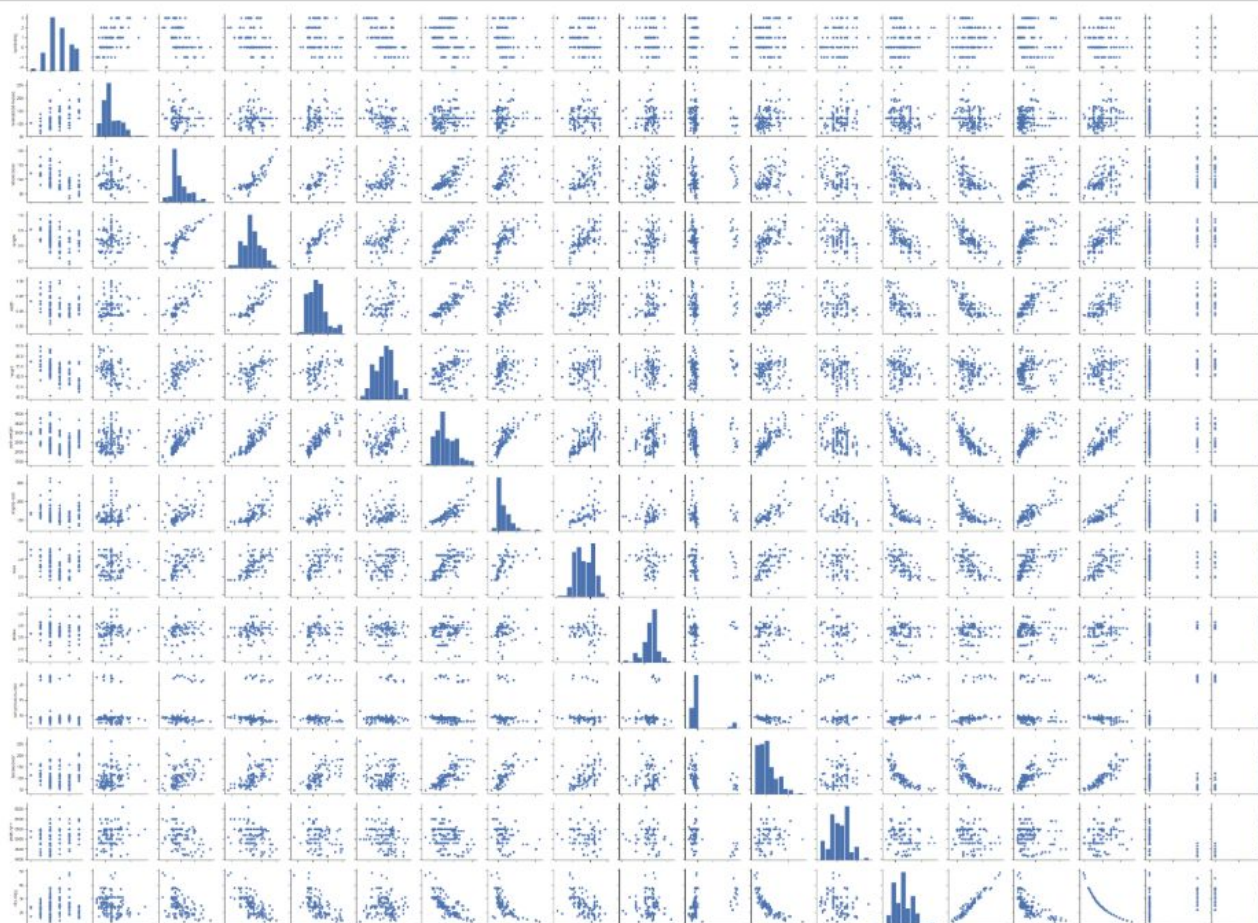
```
In [14]: sns.jointplot(x='horsepower', y='price', data=data)
Out[14]: <seaborn.axisgrid.JointGrid at 0x19852c89160>
```



Видно, что цена и количество лошадей связаны. Чем больше лошадей, тем выше стоимость автомобиля.

Построим парные диаграммы по всем наборам данным:

```
In [20]: sns.pairplot(data, plot_kws=dict(linewidth=0));
```



Информация о корреляции признаков

Построим корреляционную матрицу по всему набору данных:

In [13]: `data.corr()`

Out[13]:

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower	peak-rpm
symboling	1.000000	0.466264	-0.535987	-0.365404	-0.242423	-0.550160	-0.233118	-0.110581	-0.140019	-0.008245	-0.182196	0.075819	0.279740
normalized-losses	0.466264	1.000000	-0.056661	0.019424	0.086802	-0.373737	0.099404	0.112360	-0.029862	0.055563	-0.114713	0.217299	0.239543
wheel-base	-0.535987	-0.056661	1.000000	0.876024	0.814507	0.590742	0.782097	0.572027	0.493244	0.158502	0.250313	0.371147	-0.360305
length	-0.365404	0.019424	0.876024	1.000000	0.857170	0.492063	0.880665	0.685025	0.608971	0.124139	0.159733	0.579821	-0.285970
width	-0.242423	0.086802	0.814507	0.857170	1.000000	0.306002	0.866201	0.729436	0.544885	0.188829	0.189867	0.615077	-0.245118
height	-0.550160	-0.373737	0.590742	0.492063	0.306002	1.000000	0.307581	0.074694	0.180449	-0.062704	0.259737	-0.087027	-0.309974
curb-weight	-0.233118	0.099404	0.782097	0.880665	0.866201	0.307581	1.000000	0.849072	0.644060	0.167562	0.156433	0.757976	-0.279740
engine-size	-0.110581	0.112360	0.572027	0.685025	0.729436	0.074694	0.849072	1.000000	0.572609	0.209523	0.028889	0.822676	-0.256733
bore	-0.140019	-0.029862	0.493244	0.608971	0.544885	0.180449	0.644060	0.572609	1.000000	-0.055390	0.001263	0.566936	-0.267392
stroke	-0.008245	0.055563	0.158502	0.124139	0.188829	-0.062704	0.167562	0.209523	-0.055390	1.000000	0.187923	0.098462	-0.065713
compression-ratio	-0.182196	-0.114713	0.250313	0.159733	0.189867	0.259737	0.156433	0.028889	0.001263	0.187923	1.000000	-0.214514	-0.435780
horsepower	0.075819	0.217299	0.371147	0.579821	0.615077	-0.087027	0.757976	0.822676	0.566936	0.098462	-0.214514	1.000000	0.107885
peak-rpm	0.279740	0.239543	-0.360305	-0.285970	-0.245800	-0.309974	-0.279361	-0.256733	-0.267392	-0.065713	-0.435780	0.107885	1.000000
city-mpg	-0.035527	-0.225016	-0.470606	-0.665192	-0.633531	-0.049800	-0.749543	-0.650546	-0.582027	-0.034696	0.331425	-0.822214	-0.115077
highway-mpg	0.036233	-0.181877	-0.543304	-0.698142	-0.680635	-0.104812	-0.794889	-0.679571	-0.591309	-0.035201	0.268465	-0.804575	-0.058143
price	-0.082391	0.133999	0.584642	0.690628	0.751265	0.135486	0.834415	0.872335	0.543155	0.082310	0.071107	0.809575	-0.101118
city-L/100km	0.066171	0.238567	0.476153	0.657373	0.673363	0.003811	0.785353	0.745059	0.554610	0.037300	-0.299372	0.889488	0.115077
diesel	-0.196735	-0.101546	0.307237	0.211187	0.244356	0.281578	0.221046	0.070779	0.054458	0.241303	0.985231	-0.169053	-0.475118

Визуализируем корреляционную матрицу матрицу с помощью тепловой карты:

In [16]: `sns.heatmap(data.corr())`

Out[16]: `<matplotlib.axes._subplots.AxesSubplot at 0x201c55d2160>`

