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

In [ ]:
```
df = pd.read_csv('data/cars.csv')
```

Анализ датасета

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 539 entries, 0 to 538
Data columns (total 22 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Car_Image       539 non-null    object
 1   Name_and_model  539 non-null    object
 2   Model_type      539 non-null    object
 3   In_Game_Price   539 non-null    object
 4   car_source      539 non-null    object
 5   stock_specs     539 non-null    object
 6   Stock_Rating    539 non-null    object
 7   Drive_Type      539 non-null    object
 8   speed           539 non-null    object
 9   handling        539 non-null    object
 10  acceleration    539 non-null    object
 11  launch          539 non-null    object
 12  braking         539 non-null    object
 13  Offroad         539 non-null    object
 14  Top_Speed       539 non-null    object
 15  0-60_Mph        539 non-null    object
 16  0-100_Mph       539 non-null    object
 17  g-force         539 non-null    object
 18  car_source_1    539 non-null    object
 19  car_source_2    539 non-null    object
 20  Horse_Power     539 non-null    object
 21  Weight_lbs      539 non-null    object
dtypes: object(22)
memory usage: 92.8+ KB
```

In [ ]:
```
df.head()
```

Out[ ]:

	Car_Image	Name_and_model	Model_type	In_Game_
0	https://www.kudosprime.com/fh5/images/cars/sid...	2001 Acura Integra Type R	RETRO HOT HATCH	25
1	https://www.kudosprime.com/fh5/images/cars/sid...	2002 Acura RSX Type S	RETRO HOT HATCH	25
2	https://www.kudosprime.com/fh5/images/cars/sid...	2017 Acura NSX	MODERN SUPERCARS	170
3	https://www.kudosprime.com/fh5/images/cars/sid...	1973 Alpine A110 1600s	CLASSIC RALLY	98
4	https://www.kudosprime.com/fh5/images/cars/sid...	2017 Alpine A110	MODERN SPORTS CARS	6

5 rows × 22 columns

In [ ]:
```
from utils import get_df_info

get_df_info(df)
```

```
Столбец Car_Image (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 0)

Столбец Name_and_model (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс
1)

Столбец Model_type (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 2)

Столбец In_Game_Price (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс
3)

Столбец car_source (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 4)

Столбец stock_specs (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 5)

Столбец Stock_Rating (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс
6)

Столбец Drive_Type (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 7)

Столбец speed (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 8)

Столбец handling (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 9)

Столбец acceleration (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 1
0)

Столбец launch (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 11)

Столбец braking (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 12)

Столбец Offroad (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 13)

Столбец Top_Speed (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 14)

Столбец 0-60_Mph (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 15)

Столбец 0-100_Mph (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 16)

Столбец g-force (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 17)

Столбец car_source_1 (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 1
8)

Столбец car_source_2 (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 1
9)

Столбец Horse_Power (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 2
0)

Столбец Weight_lbs (тип object) имеет 0 пропусков из 539 значений, 0.0% (индекс 21)
```

Преобразование категориальных признаков в числовые

In [ ]:
```
mt = 'Model_type'

mt_enc = pd.DataFrame({'Model_type':df[mt]})
np.unique(mt_enc)
```

Out[ ]:
```
array(['BUGGIES', 'CLASSIC MUSCLE', 'CLASSIC RACERS', 'CLASSIC RALLY',
       'CLASSIC SPORTS CARS', 'CULT CARS', 'CULT CLASSICS', 'DRIFT CARS',
       'EXTREME TRACK TOYS', 'GT CARS', 'HOT HATCH', 'HYPERCARS',
       'MODERN MUSCLE', 'MODERN RALLY', 'MODERN SPORTS CARS',
       'MODERN SUPERCARS', 'OFFROAD', "PICK-UP & 4X4'S", 'RALLY MONSTERS',
       'RARE CLASSICS', 'RETRO HOT HATCH', 'RETRO MUSCLE', 'RETRO RALLY',
       'RETRO SALOONS', 'RETRO SPORTS CARS', 'RETRO SUPERCARS',
       'RODS AND CUSTOMS', 'SPORTS UTILITY HEROES', 'SUPER GT',
       'SUPER HOT HATCH', 'SUPER SALOONS', 'TRACK TOYS', 'TRUCKS',
       'UNLIMITED BUGGIES', 'UNLIMITED OFFROAD', "UTV'S",
       'VANS AND UTILITY', 'VINTAGE RACERS', 'info_not_found'],
      dtype=object)
```

In [ ]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

Кодирование категорий целочисленными значениями

In [ ]:
```
le = LabelEncoder()
mt_le = le.fit_transform(mt_enc[mt])
```

In [ ]:
```
mt_unq = np.unique(mt_le)
mt_unq
```

Out[ ]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38])
```

In [ ]:
```
le.inverse_transform(np.unique(mt_unq))
```

Out[ ]:
```
array(['BUGGIES', 'CLASSIC MUSCLE', 'CLASSIC RACERS', 'CLASSIC RALLY',
       'CLASSIC SPORTS CARS', 'CULT CARS', 'CULT CLASSICS', 'DRIFT CARS',
       'EXTREME TRACK TOYS', 'GT CARS', 'HOT HATCH', 'HYPERCARS',
       'MODERN MUSCLE', 'MODERN RALLY', 'MODERN SPORTS CARS',
       'MODERN SUPERCARS', 'OFFROAD', "PICK-UP & 4X4'S", 'RALLY MONSTERS',
       'RARE CLASSICS', 'RETRO HOT HATCH', 'RETRO MUSCLE', 'RETRO RALLY',
       'RETRO SALOONS', 'RETRO SPORTS CARS', 'RETRO SUPERCARS',
       'RODS AND CUSTOMS', 'SPORTS UTILITY HEROES', 'SUPER GT',
       'SUPER HOT HATCH', 'SUPER SALOONS', 'TRACK TOYS', 'TRUCKS',
       'UNLIMITED BUGGIES', 'UNLIMITED OFFROAD', "UTV'S",
       'VANS AND UTILITY', 'VINTAGE RACERS', 'info_not_found'],
      dtype=object)
```

Кодирование категорий наборами бинарных значений

In [ ]:
```
ohe = OneHotEncoder()

mt_ohe = ohe.fit_transform(mt_enc[[mt]])
```

In [ ]:
```
mt_ohe.todense()[0:10]
```

Out[ ]:
```
matrix([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 1., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 1., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0.],
        [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
         0., 0., 0., 0., 0., 0., 0.]])
```

In [ ]:
```
mt_enc.head()
```

Out[ ]:

	Model_type
0	RETRO HOT HATCH
1	RETRO HOT HATCH
2	MODERN SUPERCARS
3	CLASSIC RALLY
4	MODERN SPORTS CARS

Pandas get_dummies - быстрый вариант one-hot кодирования

In [ ]:
```
pd.get_dummies(mt_enc).head()
```

Out[ ]:

	Model_type_BUGGIES	Model_type_CLASSIC MUSCLE	Model_type_CLASSIC RACERS	Model_type_CLASSIC RALLY
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	1
4	0	0	0	0

5 rows × 39 columns