



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

## ОТЧЕТ

### *ПО РУБЕЖНЫЙ КОНТРОЛЬ №1*

### *ПО ДИСЦИПЛИНЕ «МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ»*

### *ВАРИАНТ 17*

Студент ИУ5И-25М  
(Группа)

\_\_\_\_\_  
(Подпись, дата) Ши Чжань  
(И.О.Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата) Ю.Е.Гапанюк  
(И.О.Фамилия)

2025 г.

## ВВЕДЕНИЕ

Для студентов групп ИУ5-21М, ИУ5-22М, ИУ5-23М, ИУ5-24М, ИУ5-25М  
номер варианта = номер в списке группы.

Для студентов групп ИУ5И-21М, ИУ5И-22М, ИУ5И-23М, ИУ5И-24М,  
ИУ5И-25М номер варианта = 15 + номер в списке группы.

Для студентов групп ИУ5-25МВ номер варианта = 20 + номер в списке  
группы.

Дополнительные требования по группам:

- Для студентов групп ИУ5-21М, ИУ5И-21М - для пары произвольных колонок данных построить график "Диаграмма рассеяния".
- Для студентов групп ИУ5-22М, ИУ5И-22М - для произвольной колонки данных построить гистограмму.
- Для студентов групп ИУ5-23М, ИУ5И-23М - для произвольной колонки данных построить график "Ящик с усами (boxplot)".
- Для студентов группы ИУ5-24М, ИУ5И-24М - для произвольной колонки данных построить график "Скрипичная диаграмма (violin plot)".
- Для студентов группы ИУ5-25М, ИУ5И-25М, ИУ5-25МВ - для произвольной колонки данных построить парные диаграммы (pairplot).

Каждая задача предполагает использование набора данных. Набор данных выбирается Вами произвольно с учетом следующих условий:

- Вы можете использовать один набор данных для решения всех задач, или решать каждую задачу на своем наборе данных.
- Набор данных должен отличаться от набора данных, который использовался в лекции для решения рассматриваемой задачи.

- Вы можете выбрать произвольный набор данных (например тот, который Вы использовали в лабораторных работах) или создать собственный набор данных (что актуально для некоторых задач, например, для задач удаления псевдоконстантных или повторяющихся признаков).
- Выбранный или созданный Вами набор данных должен удовлетворять условиям поставленной задачи. Например, если решается задача устранения пропусков, то набор данных должен содержать пропуски.

Полученные варианты:

- Номер варианта =  $15 + 2 = 17$
- Номер задачи №1: 17  
Задача №17 - Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием преобразования Йео-Джонсона (Yeo-Johnson transformation).
- Номер задачи №2: 37  
Задача №37 - Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectPercentile для 5% лучших признаков, и метод, основанный на взаимной информации.

Дополнительные требования по группам:

- Для студентов групп ИУ5-22М, ИУ5И-22М - Для произвольной колонки данных построить гистограмму.

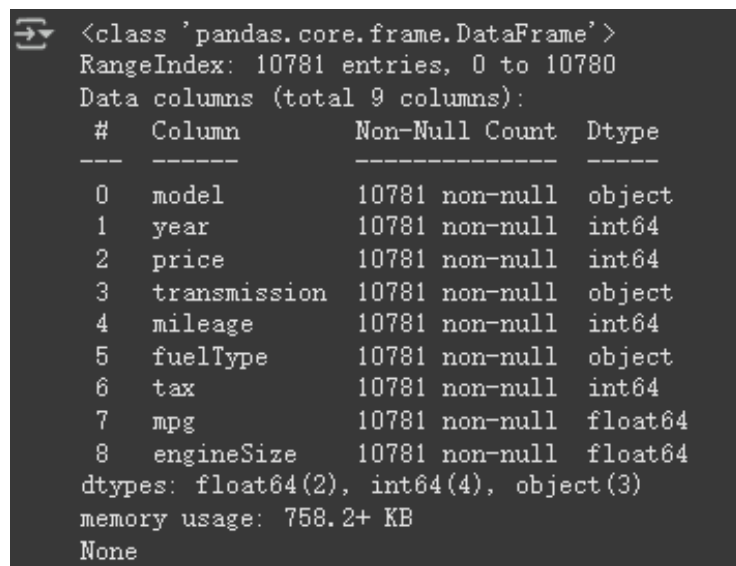
# ХОД ВЫПОЛНЕНИЯ РАБОТЫ

## Часть 1. Текстовое описание набора данных

Набор данных № 1: bmw.csv

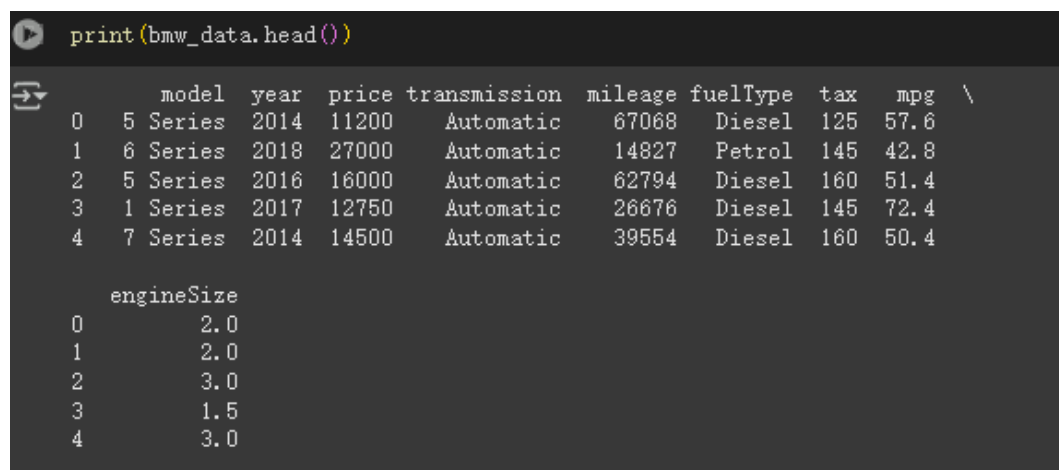
Этот набор данных собирает цены на подержанные автомобили BMW в Великобритании и используется для анализа влияния различных факторов на цены на подержанные автомобили.

Набор данных содержит информацию о цене, трансмиссии, пробеге, типе топлива, дорожном налоге, расходе миль на галлон (mpg) и объеме двигателя.



```
>>> <class 'pandas.core.frame.DataFrame'>
RangeIndex: 10781 entries, 0 to 10780
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   model           10781 non-null  object
1   year            10781 non-null  int64
2   price           10781 non-null  int64
3   transmission     10781 non-null  object
4   mileage         10781 non-null  int64
5   fuelType        10781 non-null  object
6   tax             10781 non-null  int64
7   mpg             10781 non-null  float64
8   engineSize      10781 non-null  float64
dtypes: float64(2), int64(4), object(3)
memory usage: 758.2+ KB
None
```

Рисунок 1: Информация о наборе данных (bmw.csv)



```
>>> print(bmw_data.head())
```

	model	year	price	transmission	mileage	fuelType	tax	mpg	\
0	5 Series	2014	11200	Automatic	67068	Diesel	125	57.6	
1	6 Series	2018	27000	Automatic	14827	Petrol	145	42.8	
2	5 Series	2016	16000	Automatic	62794	Diesel	160	51.4	
3	1 Series	2017	12750	Automatic	26676	Diesel	145	72.4	
4	7 Series	2014	14500	Automatic	39554	Diesel	160	50.4	

	engineSize
0	2.0
1	2.0
2	3.0
3	1.5
4	3.0

Рисунок 2: Первые 5 строк набора данных (bmw.csv)

## Набор данных № 2: Car\_Features.csv

Набор данных об автомобилях с такими характеристиками, как марка, модель, год выпуска, двигатель и другие свойства автомобиля, используемые для прогнозирования его цены.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11914 entries, 0 to 11913
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Make                  11914 non-null  object
1   Model                 11914 non-null  object
2   Year                  11914 non-null  int64
3   Engine Fuel Type      11911 non-null  object
4   Engine HP             11845 non-null  float64
5   Engine Cylinders      11884 non-null  float64
6   Transmission Type     11914 non-null  object
7   Driven_Wheels         11914 non-null  object
8   Number of Doors       11908 non-null  float64
9   Market Category       8172 non-null   object
10  Vehicle Size          11914 non-null  object
11  Vehicle Style         11914 non-null  object
12  highway MPG           11914 non-null  int64
13  city mpg              11914 non-null  int64
14  Popularity            11914 non-null  int64
15  MSRP                  11914 non-null  int64
dtypes: float64(3), int64(5), object(8)
memory usage: 1.5+ MB
None
```

Рисунок 3: Информация о наборе данных (Car\_Features.csv)

	Make	Model	Year	Engine	Fuel Type	Engine HP	\
0	BMW	1 Series M	2011	premium unleaded	(required)	335.0	
1	BMW	1 Series	2011	premium unleaded	(required)	300.0	
2	BMW	1 Series	2011	premium unleaded	(required)	300.0	
3	BMW	1 Series	2011	premium unleaded	(required)	230.0	
4	BMW	1 Series	2011	premium unleaded	(required)	230.0	
	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	\		
0	6.0	MANUAL	rear wheel drive	2.0			
1	6.0	MANUAL	rear wheel drive	2.0			
2	6.0	MANUAL	rear wheel drive	2.0			
3	6.0	MANUAL	rear wheel drive	2.0			
4	6.0	MANUAL	rear wheel drive	2.0			
	Market Category	Vehicle Size	Vehicle Style	\			
0	Factory Tuner,Luxury,High-Performance	Compact	Coupe				
1	Luxury,Performance	Compact	Convertible				
2	Luxury,High-Performance	Compact	Coupe				
3	Luxury,Performance	Compact	Coupe				
4	Luxury	Compact	Convertible				
	highway MPG	city mpg	Popularity	MSRP			
0	26	19	3916	46135			
1	28	19	3916	40650			
2	28	20	3916	36350			
3	28	18	3916	29450			
4	28	18	3916	34500			

Рисунок 4: Первые 5 строк набора данных (Car\_Features.csv)

## Часть 2. Задача №16

Задача №16 - Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием преобразования Бокса-Кокса (Box-Cox transformation).

Используя набор данных № 1: `bmw.csv`

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

# 加载数据集
data = pd.read_csv('bmw.csv')

# 定义一个函数来绘制诊断图（直方图和 Q-Q 图）
def diagnostic_plots(df, variable):
    plt.figure(figsize=(15, 6))

    # 直方图
    plt.subplot(1, 2, 1)
    df[variable].hist(bins=30, edgecolor='black', alpha=0.7)
    plt.title(f'Histogram of {variable}')

    # Q-Q 图
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot=plt)
    plt.title(f'Q-Q Plot of {variable}')

    plt.show()

# 对原始的 price 列进行诊断
diagnostic_plots(data, 'price')

# 应用 Box-Cox 变换
data['price_boxcox'], param = stats.boxcox(data['price'])

print(f'Optimal  $\lambda$  value for Box-Cox transformation: {param}')

# 对变换后的 price_boxcox 列进行诊断
diagnostic_plots(data, 'price_boxcox')
```

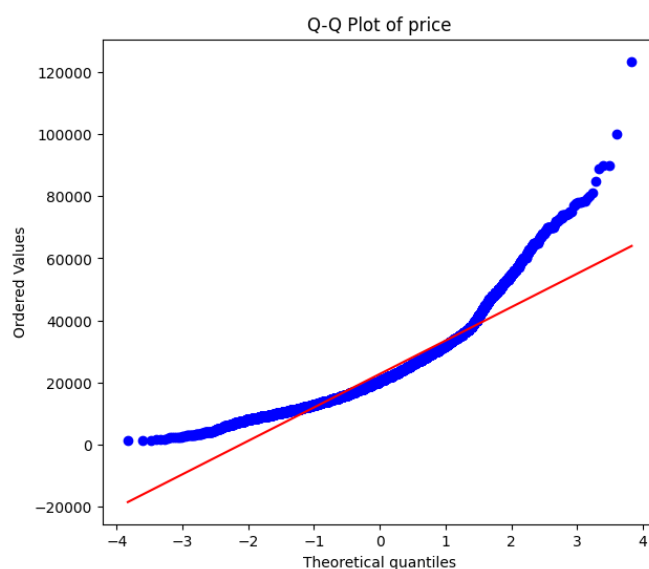
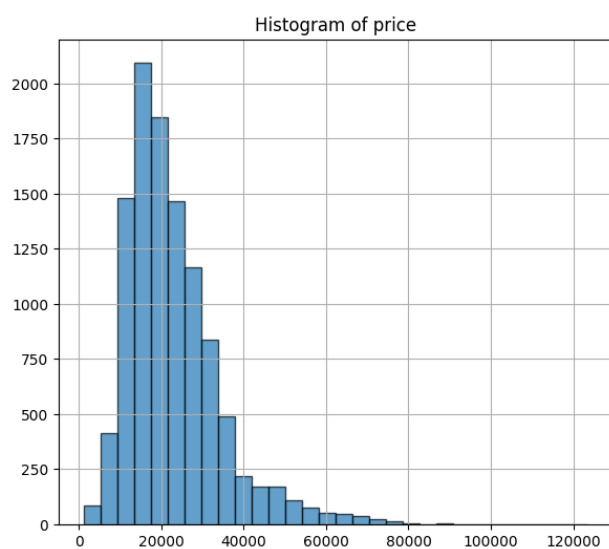


Рисунок 5: Гистограмма и график Q-Q перед преобразованием данных

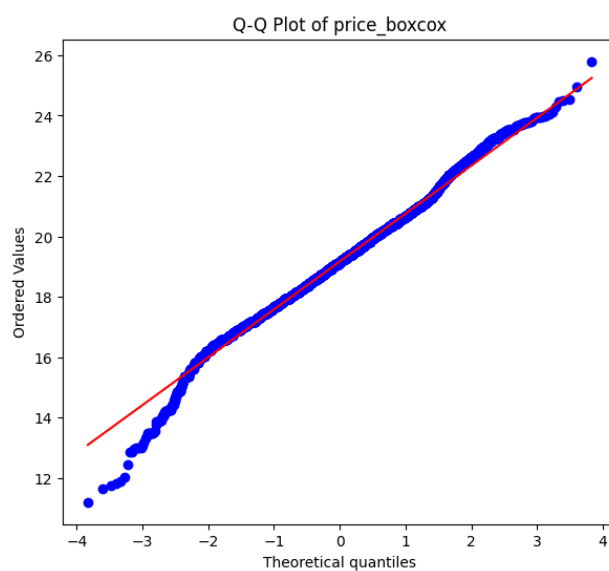
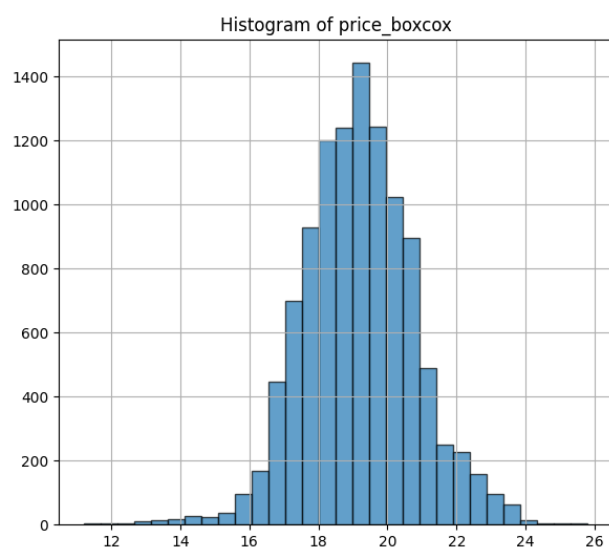


Рисунок 6: Гистограмма и график Q-Q после преобразования данных



### Часть 3. Задача №36

Задача №36 - Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectKBest для 5 лучших признаков, и метод, основанный на взаимной информации.

Используя набор данных № 2: Car\_Features.csv

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.feature_selection import SelectKBest, mutual_info_regression,
f_regression
from sklearn.impute import SimpleImputer

# 加载数据集
Car_Features_data = pd.read_csv('Car_Features.csv')
df = pd.read_csv('Car_Features.csv')

# 查看数据集的结构
print(df.info())

# 检查缺失值
print(df.isnull().sum())

# 填充缺失值
imputer = SimpleImputer(strategy='median')
numeric_columns = df.select_dtypes(include=[np.number]).columns.tolist()
df[numeric_columns] = imputer.fit_transform(df[numeric_columns])

# 假设我们要预测的目标变量是 'MSRP', 其他数值型列为特征
X = df[numeric_columns].drop('MSRP', axis=1)
y = df['MSRP']

# 使用互信息方法选择 5 个最佳特征
selector_mutual_info = SelectKBest(score_func=mutual_info_regression, k=5)
X_new_mutual_info = selector_mutual_info.fit_transform(X, y)

# 获取选中的特征名称
selected_features_mutual_info = X.columns[selector_mutual_info.get_support()]

print("\nSelected features using mutual information:")
print(selected_features_mutual_info.tolist())
```

```
# 可视化特征分数
def plot_feature_scores(selector, title):
    scores = selector.scores_
    features = X.columns
    plt.figure(figsize=(10, 6))
    plt.barh(features, scores)
    plt.xlabel('Feature Scores')
    plt.title(title)
    plt.gca().invert_yaxis()

plot_feature_scores(selector_mutual_info, 'Feature Scores using Mutual Information')
```

OUTPUT:

Selected features using mutual information:

['Year', 'Engine HP', 'highway MPG', 'city mpg', 'Popularity']

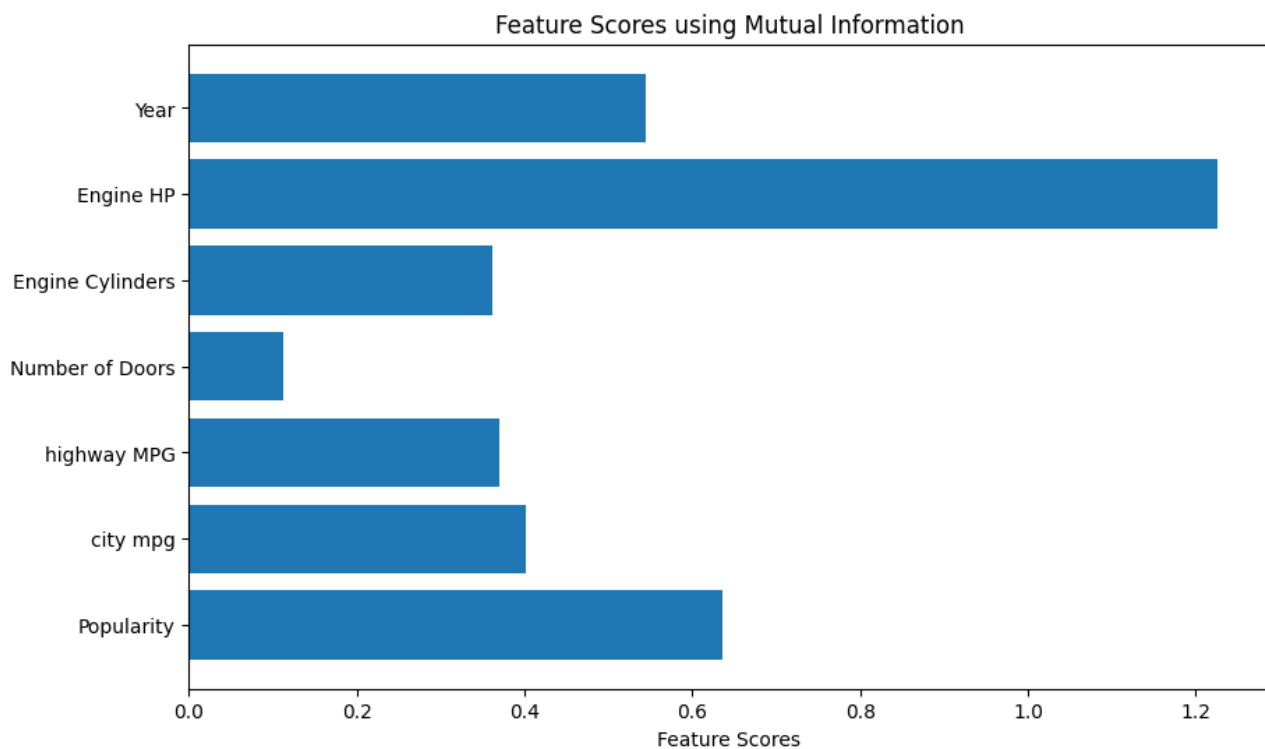


Рисунок 7: Результаты с использованием методов А и В

## Часть 4. Дополнительные требования

Для произвольной колонки данных построить гистограмму.

Используя набор данных № 1: `bmw.csv`

```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('bmw.csv')

data['price'].hist(bins=30, edgecolor='black', alpha=0.7)

plt.title('Histogram of BMW Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')

plt.show()
```

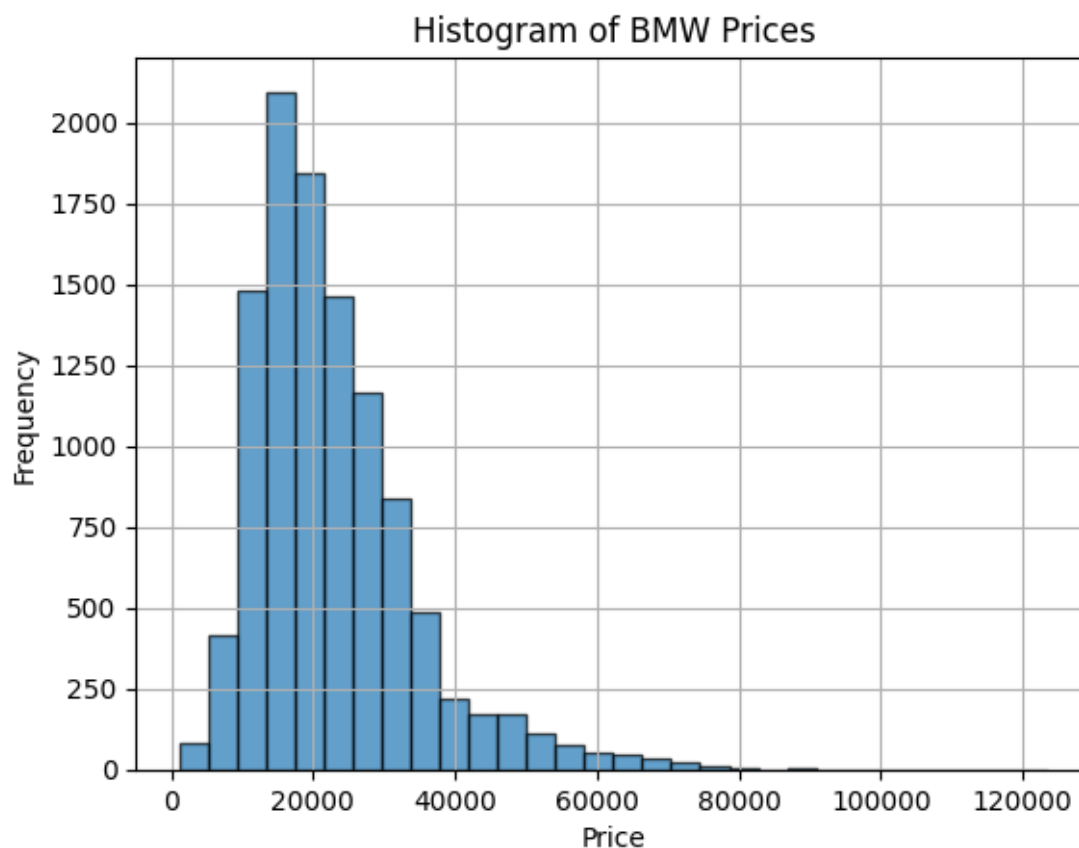


Рисунок 8: Гистограмма столбца Price

## ЗАКЛЮЧЕНИЕ

В ходе выполнения рубежного контроля №1 по дисциплине «Методы машинного обучения» была проведена комплексная работа по обработке и анализу данных из двух различных наборов, связанных с автомобилями. В рамках задачи №16 была успешно применена нормализация данных с использованием преобразования Бокса-Кокса к числовому признаку «price» из набора данных `bmw.csv`. Это позволило преобразовать распределение цен на подержанные автомобили BMW к нормальному виду, что является важным шагом при подготовке данных для многих алгоритмов машинного обучения, чувствительных к масштабу и распределению признаков.

Для задачи №36 была выполнена процедура отбора признаков на наборе данных `Car_Features.csv` с применением класса `SelectKBest` и метода, основанного на взаимной информации. В результате были выявлены пять наиболее значимых признаков для прогнозирования цены автомобиля, что демонстрирует эффективность использования данных методов в задачах предсказания и позволяет упростить модель, исключив менее значимые признаки, что может привести к улучшению её производительности и интерпретируемости.

Дополнительным требованием для группы было построение гистограммы для произвольной колонки данных, что было выполнено на примере столбца «price» из набора данных `bmw.csv`. Гистограмма наглядно представила распределение цен, что помогает визуально оценить форму распределения и presence возможных выбросов или аномалий в данных.