Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное 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 и искусственный интеллект

КАФЕДРА Системы обработки информации и управления

Лабораторная работа №3 по курсу «Методы машинного обучения в автоматизированных системах обработки информации и управления»

Подготовили:

У Жун

ИУ5И-25М

20.05.2024

Проверил:

Гапанюк Ю. Е.

2024 г.

Задание:

1. Выбрать один или несколько наборов данных (датасетов) для решения следующих задач. Каждая задача может быть решена на отдельном датасете, или несколько задач могут быть решены на одном датасете. Просьба не использовать датасет, на котором данная задача решалась в лекции.
2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
    i. масштабирование признаков (не менее чем тремя способами);
    ii. обработку выбросов для числовых признаков (по одному способу для удаления выбросов и для замены выбросов);
    iii. обработку по крайней мере одного нестандартного признака (который не является числовым или категориальным);
    iv. отбор признаков:
        - один метод из группы методов фильтрации (filter methods);
        - один метод из группы методов обертывания (wrapper methods);
        - один метод из группы методов вложений (embedded methods).

1. Загрузка наборов данных и информации

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, MinMaxScaler, RobustScaler
from sklearn.ensemble import IsolationForest
from sklearn.impute import SimpleImputer
from sklearn.feature_selection import SelectKBest, f_classif, RFE
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import Lasso

# Загрузка данных
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"
df = pd.read_csv(url, delimiter=';')

print(df.info())
print(df.describe())

# Распределение целевой переменной
sns.countplot(df['quality'])
plt.title('Распределение качества вина')
plt.show()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
None
```

	fixed acidity	volatile acidity	citric acid	residual sugar	\
count	1599.000000	1599.000000	1599.000000	1599.000000	
mean	8.319637	0.527821	0.270976	2.538806	
std	1.741096	0.179060	0.194801	1.409928	
min	4.600000	0.120000	0.000000	0.900000	
25%	7.100000	0.390000	0.090000	1.900000	
50%	7.900000	0.520000	0.260000	2.200000	
75%	9.200000	0.640000	0.420000	2.600000	
max	15.900000	1.580000	1.000000	15.500000	

	chlorides	free sulfur dioxide	total sulfur dioxide	density	\
count	1599.000000	1599.000000	1599.000000	1599.000000	
mean	0.087467	15.874922	46.467792	0.996747	
std	0.047065	10.460157	32.895324	0.001887	
min	0.012000	1.000000	6.000000	0.990070	
25%	0.070000	7.000000	22.000000	0.995600	
50%	0.079000	14.000000	38.000000	0.996750	
75%	0.090000	21.000000	62.000000	0.997835	
max	0.611000	72.000000	289.000000	1.003690	

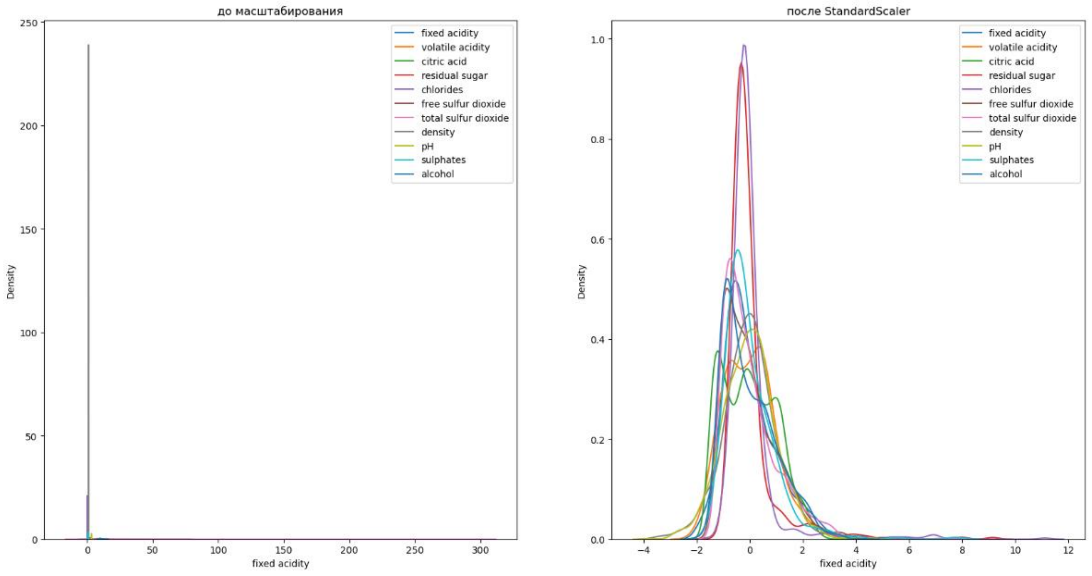
	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000
mean	3.311113	0.658149	10.422983	5.636023
std	0.154386	0.169507	1.065668	0.807569
min	2.740000	0.330000	8.400000	3.000000
25%	3.210000	0.550000	9.500000	5.000000
50%	3.310000	0.620000	10.200000	6.000000
75%	3.400000	0.730000	11.100000	6.000000
max	4.010000	2.000000	14.900000	8.000000

Рис.1-Загрузка наборов данных и информации

2. Масштабирование признаков

Я использую три метода масштабирования: StandardScaler, MinMaxScaler и RobustScaler.

```python
# Функция для отрисовки KDE графиков до и после масштабирования
def draw_kde(col_list, df1, df2, label1, label2):
    fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(20, 10))
    # первый график
    ax1.set_title(label1)
    for col in col_list:
        sns.kdeplot(df1[col], ax=ax1, label=col)
    ax1.legend()

    # второй график
    ax2.set_title(label2)
    for col in col_list:
        sns.kdeplot(df2[col], ax=ax2, label=col)
    ax2.legend()

    plt.show()

scalers = {
    'StandardScaler': StandardScaler(),
    'MinMaxScaler': MinMaxScaler(),
    'RobustScaler': RobustScaler()
}

scaled_features = {}
for key, scaler in scalers.items():
    scaled_features[key] = scaler.fit_transform(df.drop(columns='quality'))
    scaled_df = pd.DataFrame(scaled_features[key], columns=df.columns[:-1])
    scaled_df['quality'] = df['quality']

    draw_kde(df.columns[:-1], df, scaled_df, 'до масштабирования', f'после {key}')
```



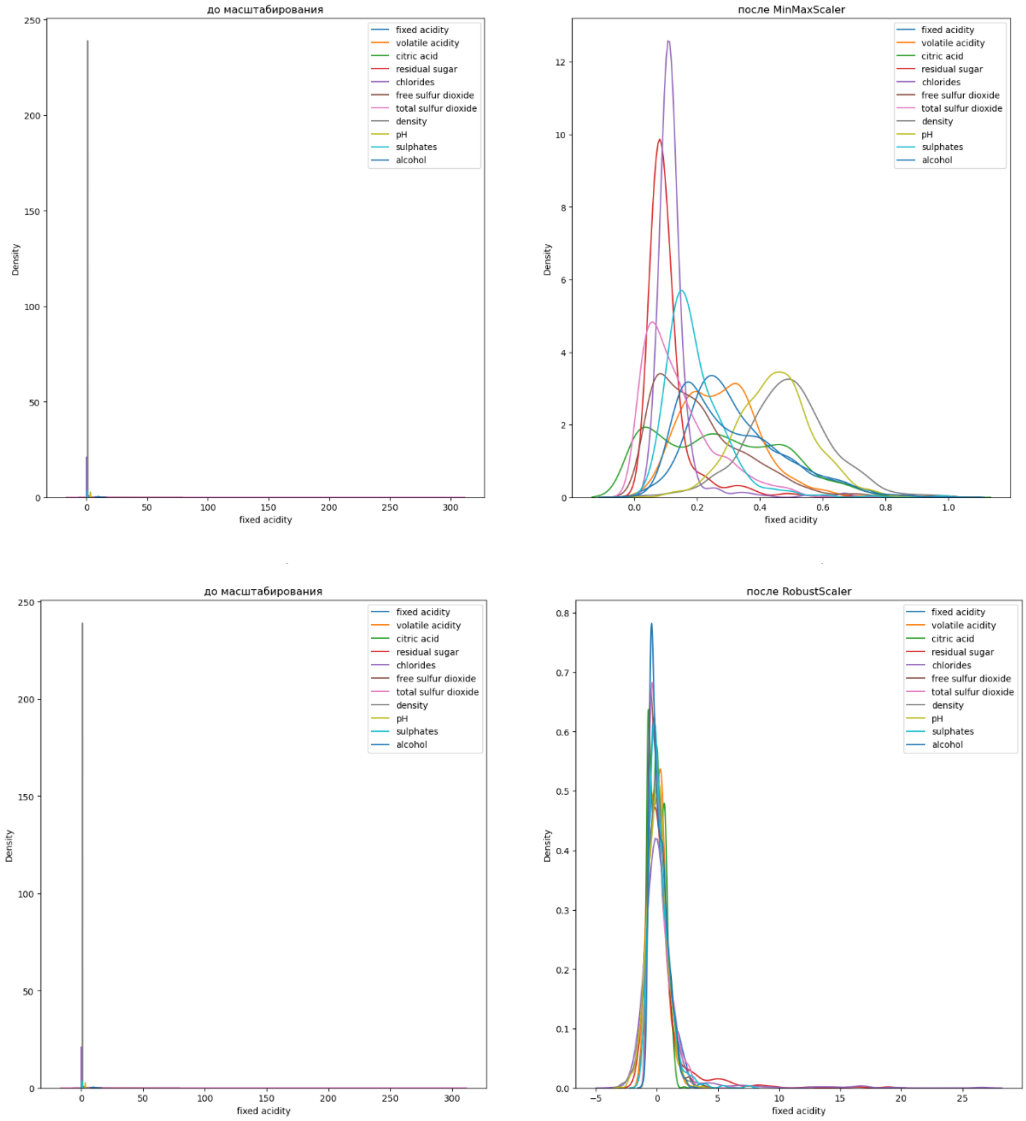


Рис. 2 – Кед-диаграммы трех методов масштабирования

3. Обработка выбросов для числовых признаков

Я использую IsolationForest для удаления провалов и замены провалов медианной заменой.

```
: iso = IsolationForest(contamination=0.1)
yhat = iso.fit_predict(df.drop(columns='quality'))
mask = yhat != -1
df_no_outliers = df[mask]


df_imputed = df.copy()
for column in df.columns[:-1]:
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    median = df[column].median()
    df_imputed[column] = np.where((df[column] < lower_bound) | (df[column] > upper_bound), median, df[column])

plt.figure(figsize=(14, 7))
plt.subplot(1, 2, 1)
sns.boxplot(data=df_no_outliers)
plt.title('Без выбросов')

plt.subplot(1, 2, 2)
sns.boxplot(data=df_imputed)
plt.title('Замена выбросов медианой')
plt.show()
```

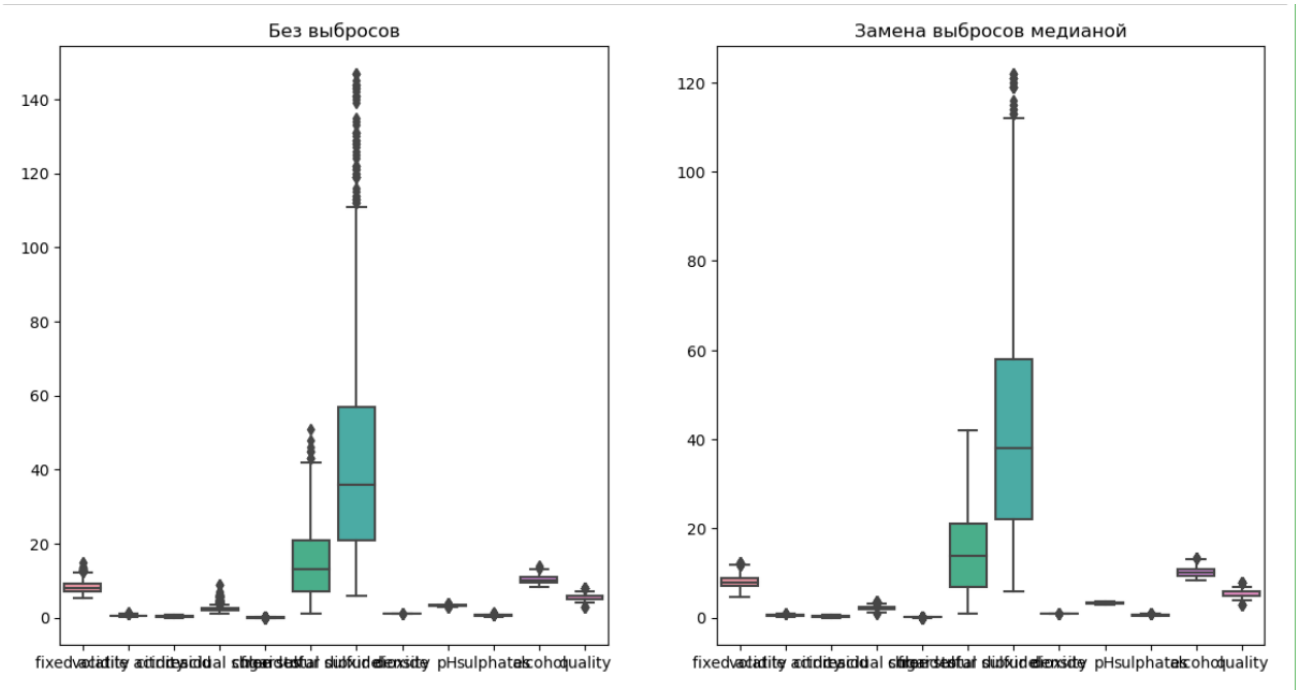
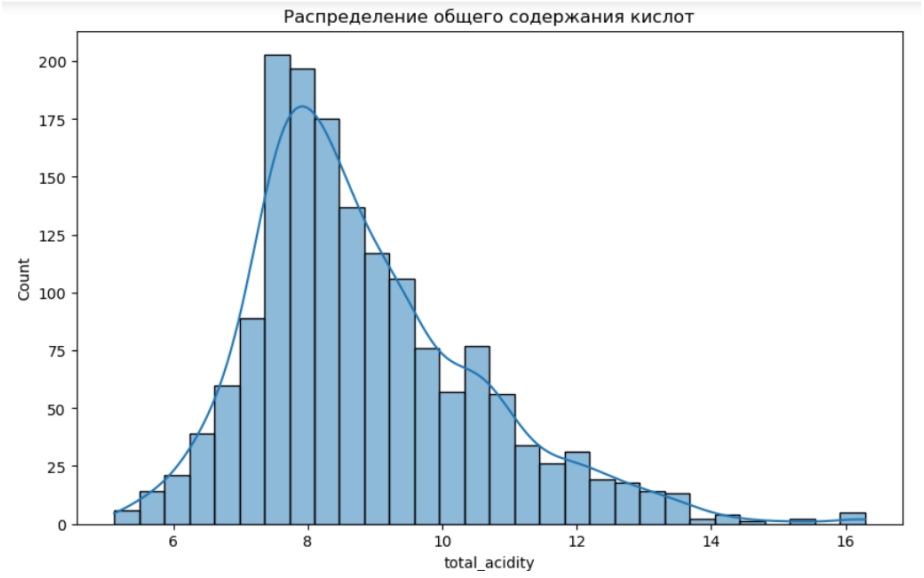


Рис.3– Сравнение распределений до и после обработки выбросов

4. Обработка по крайней мере одного нестандартного признака

```
df['total_acidity'] = df['fixed acidity'] + df['volatile acidity']


plt.figure(figsize=(10, 6))
sns.histplot(df['total_acidity'], bins=30, kde=True)
plt.title('Распределение общего содержания кислот')
plt.show()

# Q-Q plot
import scipy.stats as stats
plt.figure(figsize=(6, 6))
stats.probplot(df['total_acidity'], dist="norm", plot=plt)
plt.title('Q-Q plot общего содержания кислот')
plt.show()
```



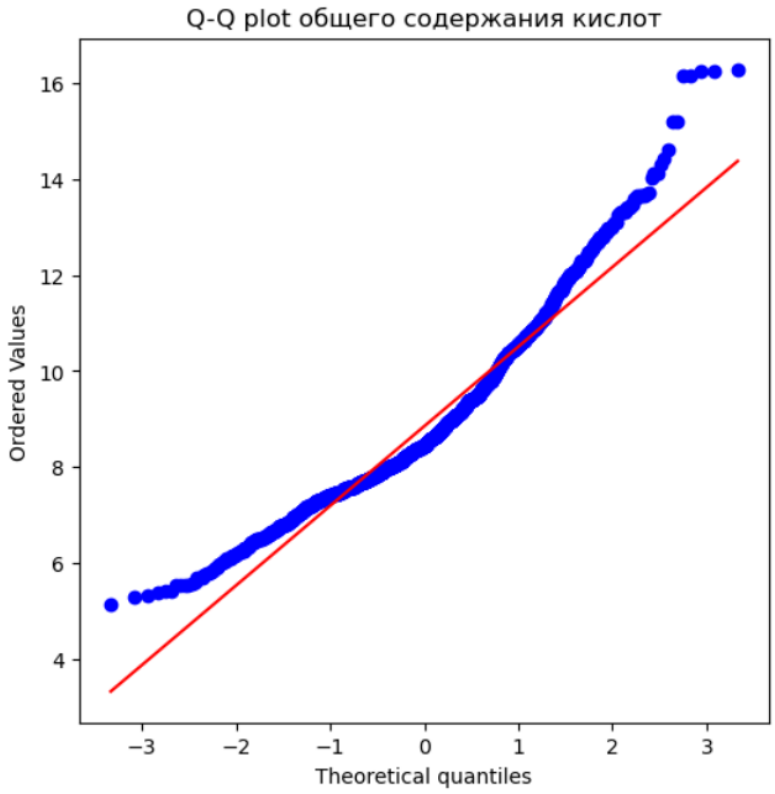


Рис.4– визуализируем его распределение и Q-Q plot.

5. Отбор признаков

Я использую методы SelectKBest (фильтрация), RFE (обертывание) и Lasso (встраивание).

```
from sklearn.model_selection import train_test_split

X = df.drop(columns='quality')
y = df['quality']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Filter method: SelectKBest
selector = SelectKBest(f_classif, k=5)
X_train_kbest = selector.fit_transform(X_train, y_train)
selected_features_kbest = X.columns[selector.get_support()]

print("Выбранные признаки (SelectKBest):", selected_features_kbest)

# Wrapper method: RFE
model = RandomForestClassifier()
rfe = RFE(model, n_features_to_select=5)
X_train_rfe = rfe.fit_transform(X_train, y_train)
selected_features_rfe = X.columns[rfe.get_support()]

print("Выбранные признаки (RFE):", selected_features_rfe)

# Embedded method: Lasso
lasso = Lasso(alpha=0.01)
lasso.fit(X_train, y_train)
selected_features_lasso = X.columns[lasso.coef_ != 0]

print("Выбранные признаки (Lasso):", selected_features_lasso)
```

```
Выбранные признаки (SelectKBest): Index(['volatile acidity', 'total sulfur dioxide', 'density', 'sulphates',
       'alcohol'],
      dtype='object')
Выбранные признаки (RFE): Index(['volatile acidity', 'total sulfur dioxide', 'density', 'sulphates',
       'alcohol'],
      dtype='object')
Выбранные признаки (Lasso): Index(['fixed acidity', 'volatile acidity', 'residual sugar',
       'free sulfur dioxide', 'total sulfur dioxide', 'sulphates', 'alcohol'],
      dtype='object')
```

Рис.5–Отбор признаков