Полученное задание

Задача №2.

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?
Набор данных: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html#sklearn.datasets.load_wine
Дополнительные требования: для пары произвольных колонок данных построить график "Диаграмма рассеяния"

Анализ набора данных Wine

Набор данных Wine содержит информацию о химическом составе вин, выращенных в одном регионе Италии, но от трех разных культиваторов. Данные включают 13 количественных признаков (например, алкоголь, яблочная кислота, зола и т.д.) и целевую переменную - сорт винограда (3 класса). В этом анализе мы обработаем пропуски в данных и подготовим признаки для моделирования.

Импорт необходимых библиотек

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_wine
```

Загрузка и предварительный просмотр данных

```
# Загрузка данных
wine = load_wine()
wine_df = pd.DataFrame(data=wine.data, columns=wine.feature_names)
wine_df['target'] = wine.target

# Просмотр первых строк данных
print(wine_df.head())

# Проверка информации о данных
print("\nИнформация о данных:")
print(wine_df.info())

# Проверка наличия пропусков
print("\nКоличество пропусков в каждом столбце:")
print(wine_df.isnull().sum())
```

```
   alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
0    14.23        1.71  2.43               15.6      127.0           2.80
1    13.20        1.78  2.14               11.2      100.0           2.65
2    13.16        2.36  2.67               18.6      101.0           2.80
3    14.37        1.95  2.50               16.8      113.0           3.85
4    13.24        2.59  2.87               21.0      118.0           2.80

   flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
0        3.06                  0.28             2.29             5.64  1.04
1        2.76                  0.26             1.28             4.38  1.05
2        3.24                  0.30             2.81             5.68  1.03
3        3.49                  0.24             2.18             7.80  0.86
4        2.69                  0.39             1.82             4.32  1.04

   od280/od315_of_diluted_wines  proline  target
0                          3.92   1065.0       0
1                          3.40   1050.0       0
2                          3.17   1185.0       0
3                          3.45   1480.0       0
4                          2.93    735.0       0

Информация о данных:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   alcohol                       178 non-null    float64
 1   malic_acid                    178 non-null    float64
 2   ash                           178 non-null    float64
 3   alcalinity_of_ash             178 non-null    float64
 4   magnesium                     178 non-null    float64
 5   total_phenols                 178 non-null    float64
 6   flavanoids                    178 non-null    float64
 7   nonflavanoid_phenols          178 non-null    float64
 8   proanthocyanins               178 non-null    float64
 9   color_intensity               178 non-null    float64
 10  hue                           178 non-null    float64
 11  od280/od315_of_diluted_wines  178 non-null    float64
 12  proline                       178 non-null    float64
 13  target                        178 non-null    int64
dtypes: float64(13), int64(1)
memory usage: 19.6 KB
None

Количество пропусков в каждом столбце:
```

```
alcohol                         0
malic_acid                      0
ash                             0
alcalinity_of_ash               0
magnesium                       0
total_phenols                   0
flavanoids                      0
nonflavanoid_phenols            0
proanthocyanins                 0
color_intensity                 0
hue                             0
od280/od315_of_diluted_wines    0
proline                         0
target                          0
dtype: int64
```

Искусственное создание пропусков для демонстрации

Поскольку в исходном наборе данных пропусков нет, мы искусственно создадим их в двух столбцах:

- 'alcohol' (количественный признак)
- 'hue' (категориальный признак после дискретизации)

```
# Создание пропусков в количественном признаке (alcohol)
np.random.seed(42)
missing_indices = np.random.choice(wine_df.index, size=10, replace=False)
wine_df.loc[missing_indices, 'alcohol'] = np.nan

# Дискретизация признака 'hue' для создания категориального признака
wine_df['hue_category'] = pd.cut(wine_df['hue'], bins=3, labels=['low', 'medium',
'high'])

# Создание пропусков в категориальном признаке (hue_category)
missing_indices = np.random.choice(wine_df.index, size=5, replace=False)
wine_df.loc[missing_indices, 'hue_category'] = np.nan

# Проверка созданных пропусков
print("\nКоличество пропусков после их создания:")
print(wine_df[['alcohol', 'hue_category']].isnull().sum())
```

```
Количество пропусков после их создания:
alcohol         10
hue_category     5
dtype: int64
```

Обработка пропусков

Обработка пропусков в количественном признаке (alcohol)

Для количественных признаков распространенные стратегии:

1. Замена средним/медианным значением
2. Использование предсказательных моделей
3. Удаление строк с пропусками

Мы используем замену медианным значением, так как оно устойчиво к выбросам.

```
# Замена пропусков в alcohol медианным значением (исправленная версия)
alcohol_median = wine_df['alcohol'].median()
wine_df['alcohol'] = wine_df['alcohol'].fillna(alcohol_median)

print(f"\nМедианное значение alcohol: {alcohol_median:.2f}")
print("Количество пропусков в alcohol после обработки:",
wine_df['alcohol'].isnull().sum())
```

```
Медианное значение alcohol: 13.05
Количество пропусков в alcohol после обработки: 0
```

Обработка пропусков в категориальном признаке (hue_category)

Для категориальных признаков распространенные стратегии:

1. Замена модальным значением (наиболее частой категорией)
2. Создание отдельной категории для пропусков
3. Удаление строк с пропусками

Мы используем замену модальным значением, так как это сохраняет все наблюдения.

```
# Замена пропусков в hue_category модальным значением (исправленная версия)
hue_mode = wine_df['hue_category'].mode()[0]
wine_df['hue_category'] = wine_df['hue_category'].fillna(hue_mode)

print(f"\nМодальное значение hue_category: {hue_mode}")
print("Количество пропусков в hue_category после обработки:",
wine_df['hue_category'].isnull().sum())
```

```
Модальное значение hue_category: medium
Количество пропусков в hue_category после обработки: 0
```

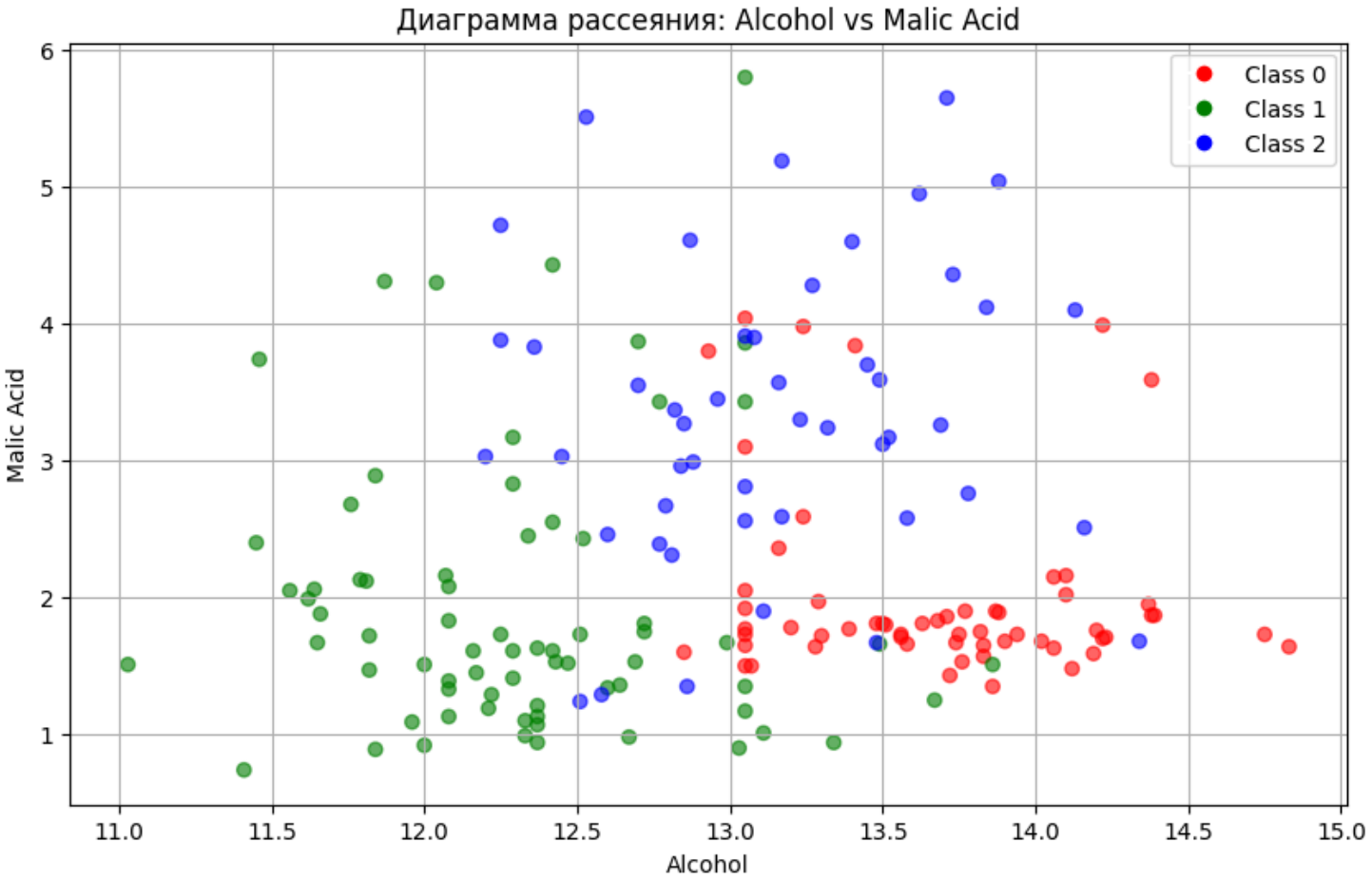
Визуализация данных: Диаграмма рассеяния

Построим диаграмму рассеяния для пары признаков 'alcohol' и 'malic_acid' с цветовой кодировкой по целевому классу.

```
plt.figure(figsize=(10, 6))
colors = {0: 'red', 1: 'green', 2: 'blue'}
plt.scatter(wine_df['alcohol'], wine_df['malic_acid'],
c=wine_df['target'].map(colors), alpha=0.6)
plt.title('Диаграмма рассеяния: Alcohol vs Malic Acid')
plt.xlabel('Alcohol')
plt.ylabel('Malic Acid')
plt.grid(True)

# Создание легенды
from matplotlib.lines import Line2D
legend_elements = [Line2D([0], [0], marker='o', color='w', label='Class 0',
markerfacecolor='red', markersize=8),
                   Line2D([0], [0], marker='o', color='w', label='Class 1',
markerfacecolor='green', markersize=8),
                   Line2D([0], [0], marker='o', color='w', label='Class 2',
markerfacecolor='blue', markersize=8)]
plt.legend(handles=legend_elements)

plt.show()
```



Выбор признаков для моделирования

Для построения моделей машинного обучения я выберу следующие признаки:

1. Все исходные количественные признаки (13 признаков), так как они содержат важную информацию о химическом составе вина.

2. Исключу созданный категориальный признак 'hue_category', так как он является производным от исходного количественного признака 'hue'.

Причины выбора:

- Количественные признаки хорошо подходят для большинства алгоритмов машинного обучения.
- Химические показатели напрямую влияют на качество и сорт вина.
- Все признаки уже масштабированы и не требуют дополнительной предобработки (кроме уже выполненной обработки пропусков).

```
# Подготовка финального набора данных для моделирования
X = wine_df[wine.feature_names]  # Все исходные количественные признаки
y = wine_df['target']  # Целевая переменная

print("\nФорма итогового набора данных для моделирования:", X.shape)
```

```
Форма итогового набора данных для моделирования: (178, 13)
```