ЛАБОРАТОРНАЯ РАБОТА 3. МЕТРИЧЕСКИЕ МЕТОДЫ КЛАССИФИКАЦИИ

**Исследование набора данных**

**Цели и задачи**

Цель лабораторной работы: изучение принципов построения информационных систем с использованием метрических методов классификации.

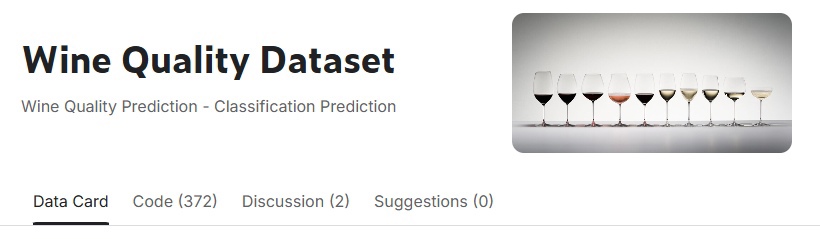
Основные задачи:

* Изучение инструментария Python для реализации алгоритмов метрической классификации;
* Изучение методов оптимизации параметров метрической классификации;
* Освоение модификаций kNN-метода.

**Индивидуальное задание**

**Описание набора данных**

Этот набор данных содержит информацию о различных характеристиках вина. Он включает 12 столбцов, каждый из которых представляет собой определенный атрибут вина. Данные могут использоваться для анализа и прогнозирования качества вина на основе его химических свойств.



Цель набора данных:

Основная цель этого набора данных — исследовать, как различные физико-химические свойства вина влияют на его качество. Это может помочь виноделам и исследователям оптимизировать процесс производства вина и улучшить его качество.

Описание каждого признака и его тип:

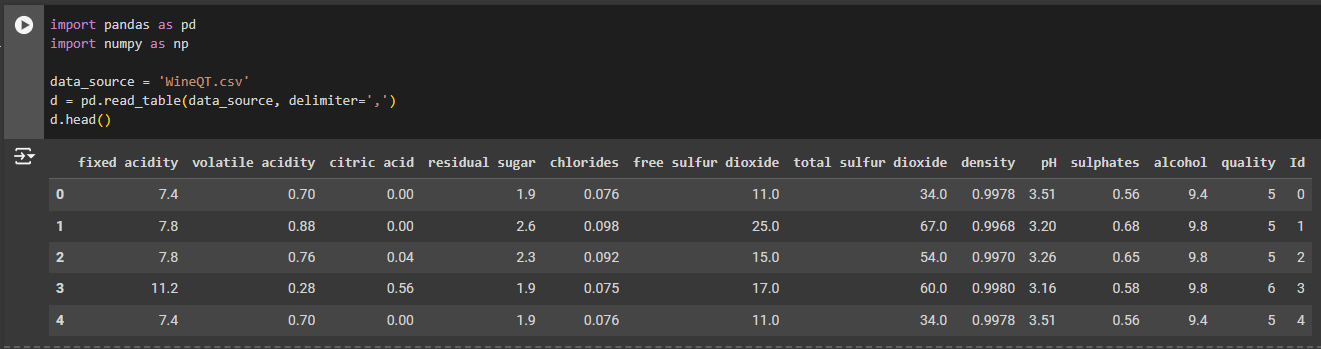
* Id (Тип: Целое число): Уникальный идентификатор.
* fixed acidity (Тип: Число с плавающей точкой): Количество непарных кислот в вине.
* volatile acidity (Тип: Число с плавающей точкой): Количество парных кислот в вине.
* citric acid (Тип: Число с плавающей точкой): Количество лимонной кислоты в вине.
* residual sugar (Тип: Число с плавающей точкой): Количество остаточного сахара в вине после ферментации.
* chlorides (Тип: Число с плавающей точкой): Количество солей (хлоридов) в вине.
* free sulfur dioxide (Тип: Число с плавающей точкой): Количество свободного диоксида серы в вине.
* total sulfur dioxide (Тип: Число с плавающей точкой): Общее количество диоксида серы в вине.
* density (Тип: Число с плавающей точкой): Плотность вина.
* pH (Тип: Число с плавающей точкой): Уровень кислотности вина.
* sulphates (Тип: Число с плавающей точкой): Количество сульфатов в вине.
* alcohol (Тип: Число с плавающей точкой): Количество алкоголя в вине.
* quality (Тип: Целое число): Оценка качества вина (обычно от 0 до 10).

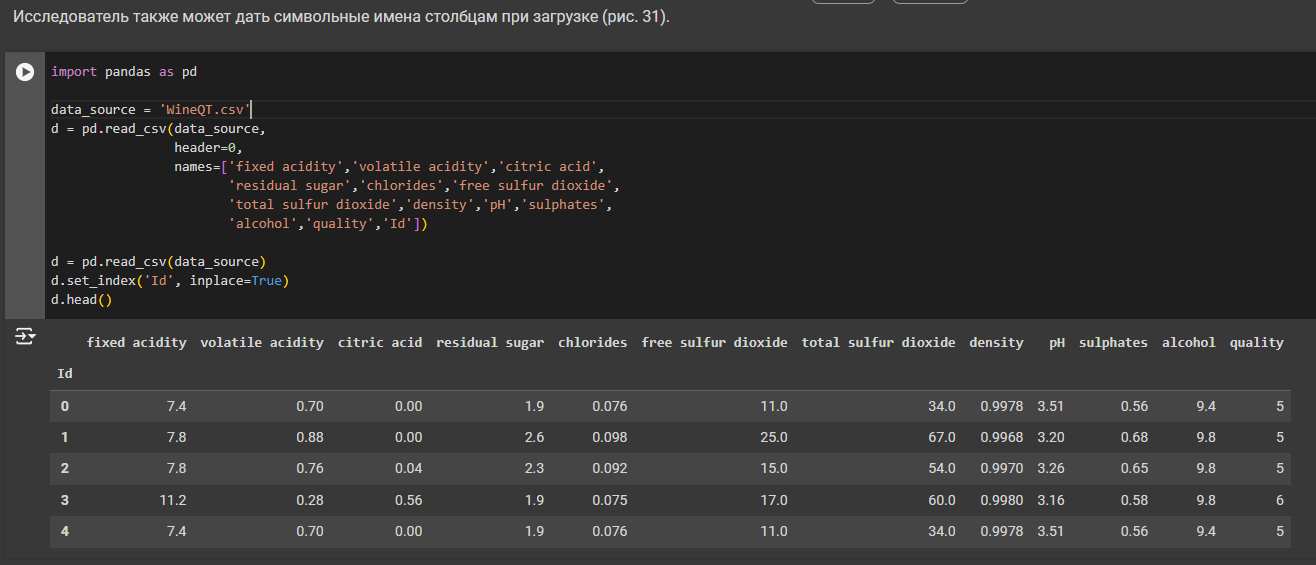
**Форма набора данных**

Характеристики набора данных:

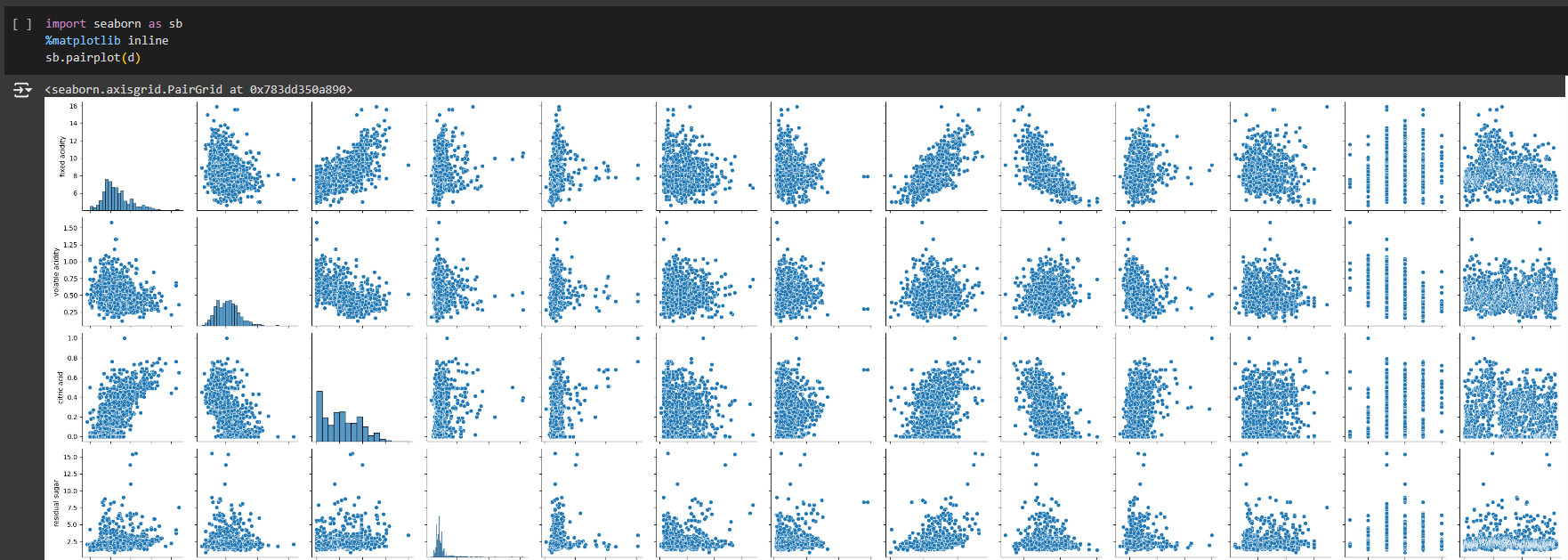
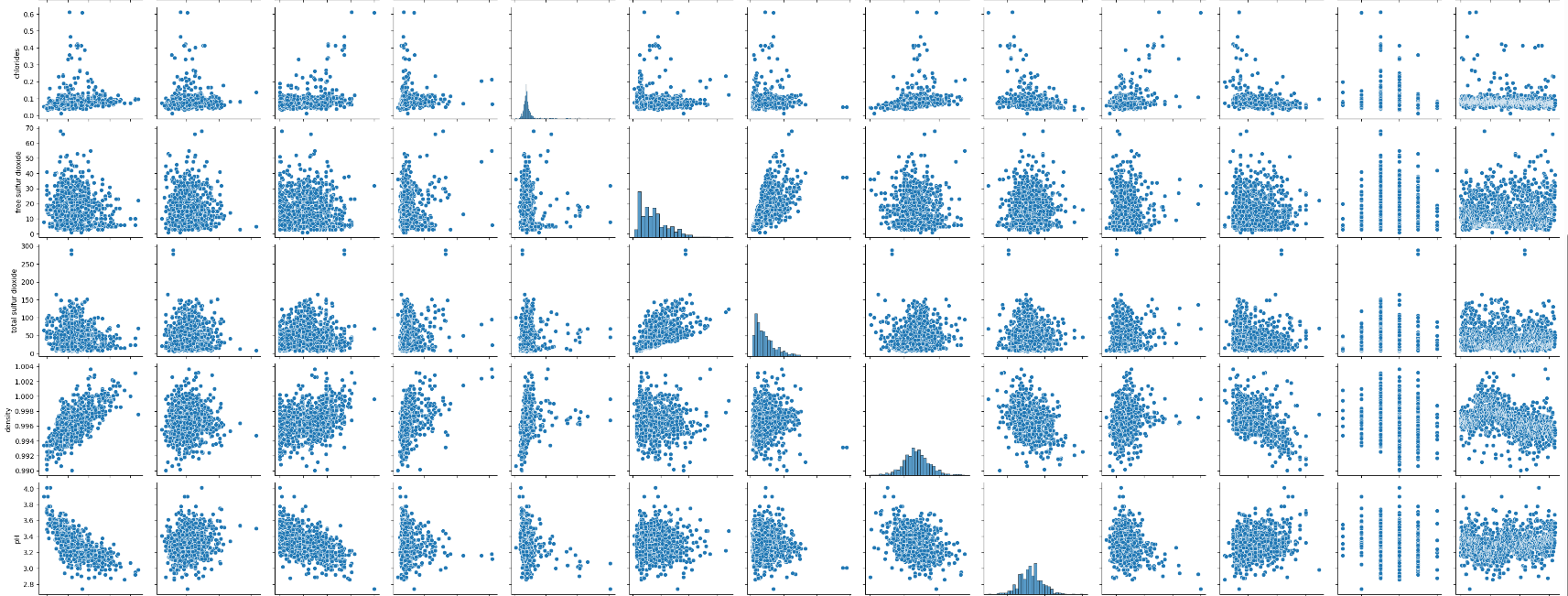
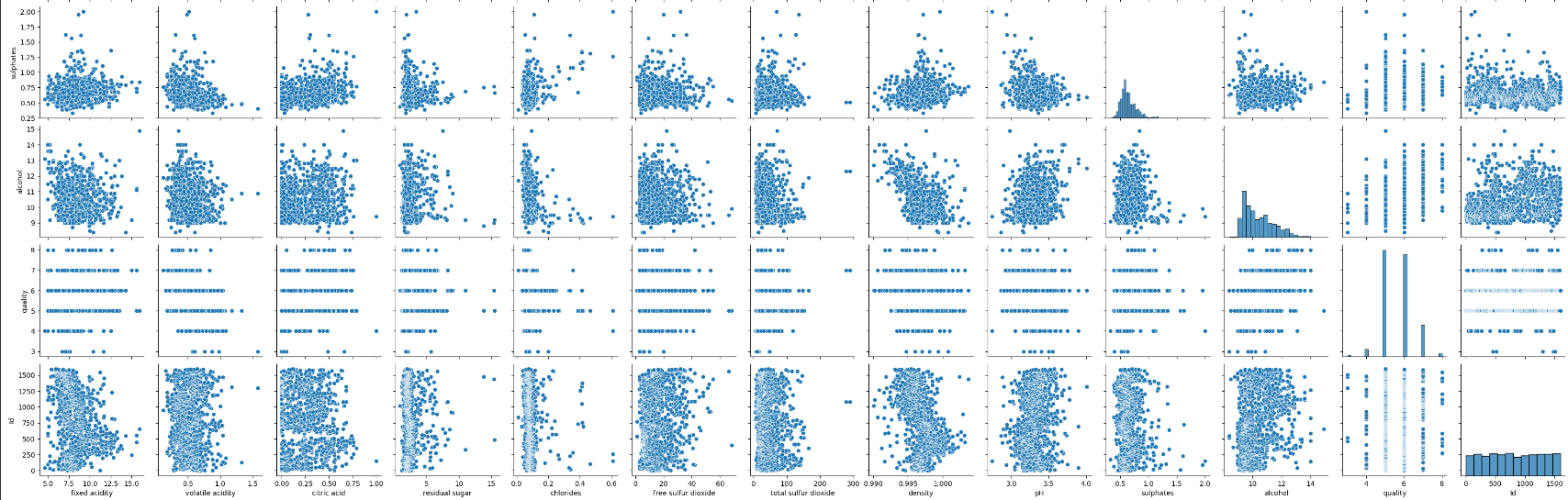
* Количество элементов (строк): 1144
* Количество признаков (столбцов): 13
* Типы данных: Float (11 шт.), int(2 шт.)

1. Загрузка исходного набора данных.

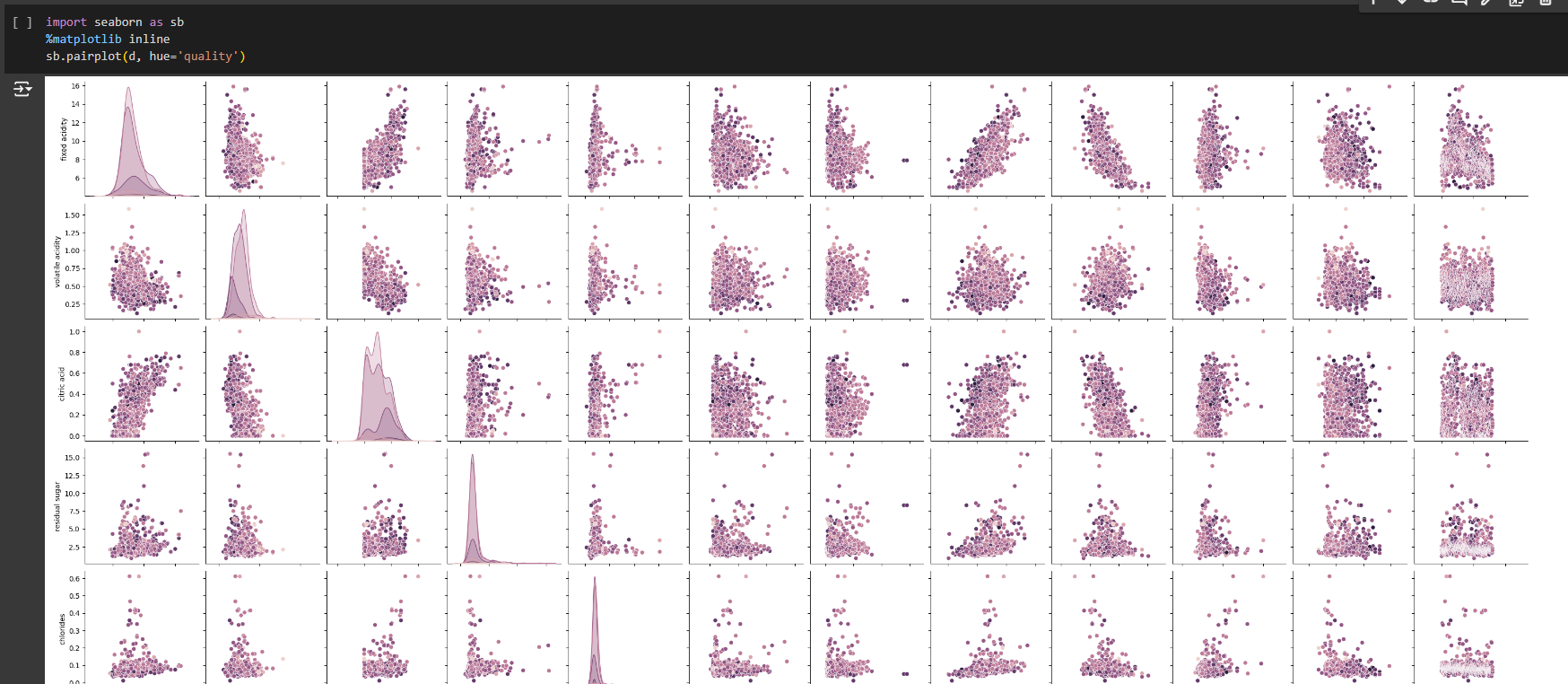
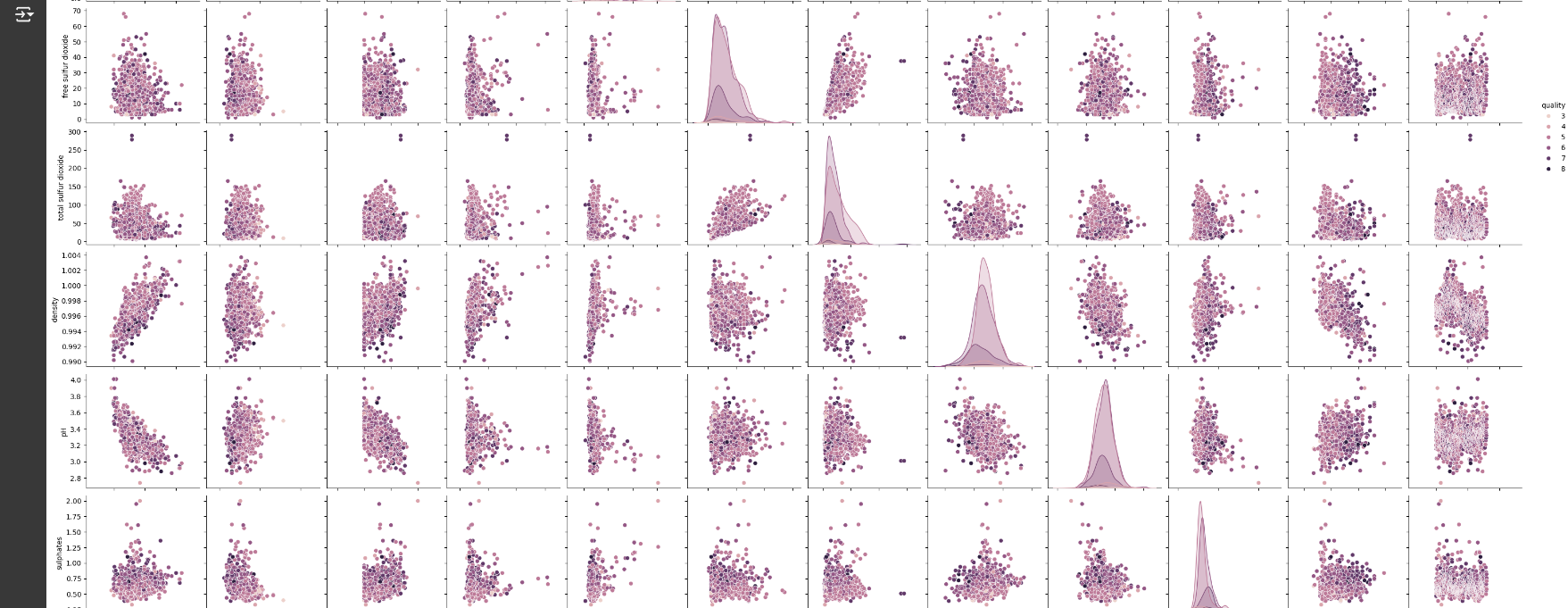
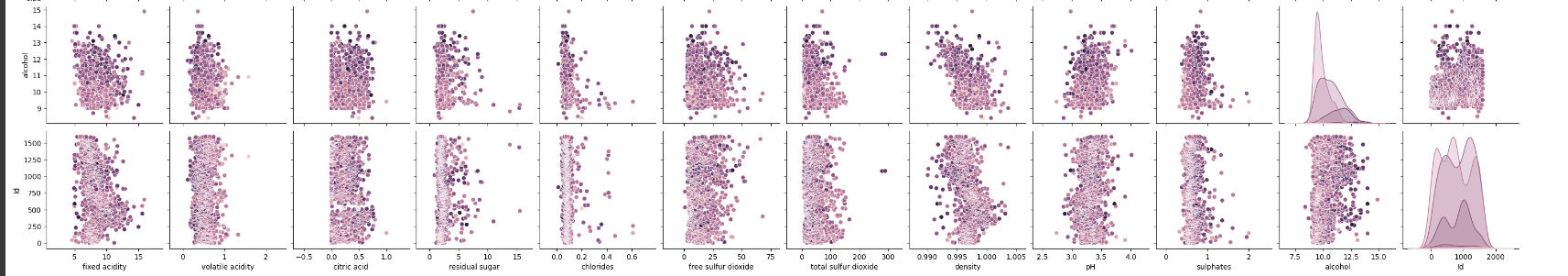




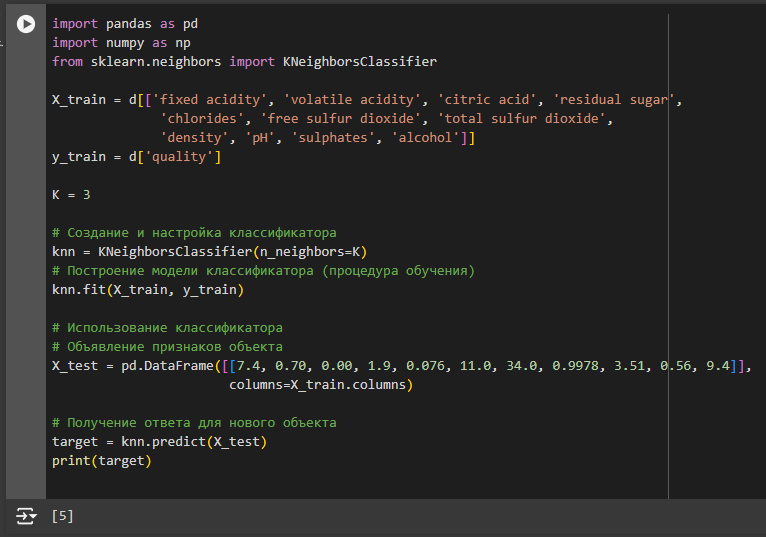
1. После загрузки данных можно визуализировать полученный набор данных.

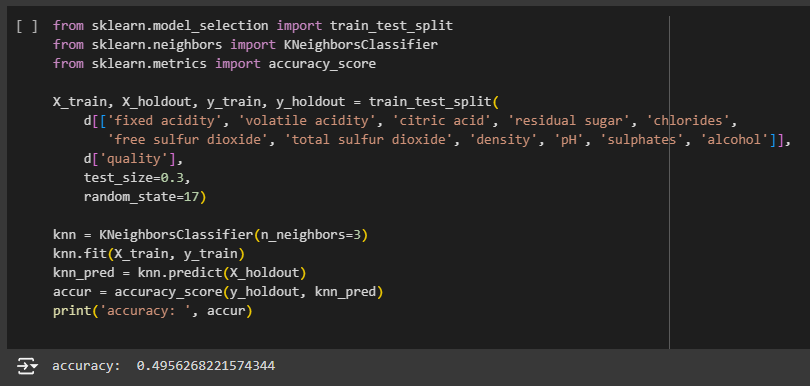
1. Для придания отдельным классам своих цветов необходимо указать, по какому признаку разделяются точки.

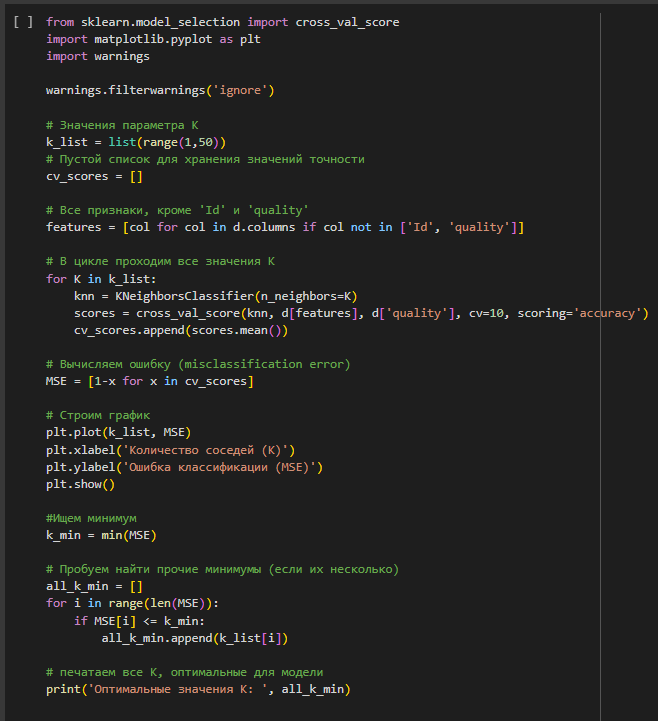
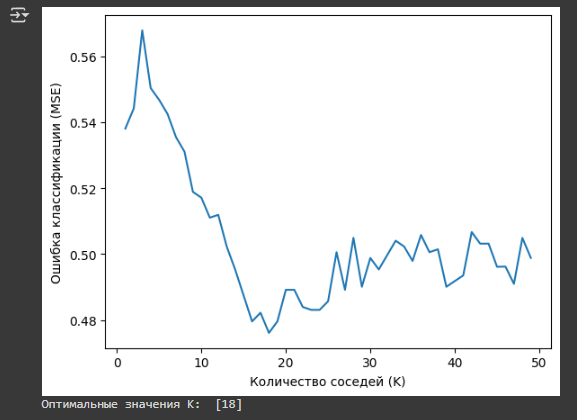
1. Перейдем к построению модели. Модель метрической классификации должна обеспечивать следующий алгоритм работы: пользователь вводит новое признаковое описание объекта (объект отсутствует в обучающей выборке), а алгоритм классификации относит новый объект к одному из классов ирисов.
2. Существует несколько вариаций метода ближайших соседей. Каждая модель предполагает наличие различных параметров для оптимизации. Воспользуемся библиотекой scikit для построения классификатора (рис. 36).



1. Модель построена и выдает ответ для новых (отсутствующих в исходной выборке) объектов. Но, анализируя код на рис. 36, следует отметить следующие недостатки такого подхода: – в качестве количества ближайших соседей выбрано значение K=3, выбор данного значения не обосновывается, но в данном методе именно данный параметр должен оптимизироваться; – отсутствует какое-либо графическое представление модели, нет визуализации процесса принятия решения. Исправим данные недостатки.
2. Займемся обоснованием выбора оптимального значения количества ближайших соседей. Для этого будем использовать простейшую оценку качества hold-out (рис. 37).



1. Еще одна оценка качества – cross validation (CV) error. На рис. 3.10 показан алгоритм получения оценки точности классификации CV и процедура выбора оптимального значения количества соседей в алгоритме kNN на основе данной оценки.

**Контрольные вопросы**

1. Поясните особенности основных методов метрической классификации: метод ближайшего соседа, метод k ближайших соседей.

Метод ближайших соседей — простейший метрический классификатор, основанный на оценивании сходства объектов. Классифицируемый объект относится к тому классу, которому принадлежат ближайшие к нему объекты обучающей выборки.

Метод k ближайших соседей (kNN) позволяет повысить надёжность классификации. Для этого объект относится к тому классу, которому принадлежит большинство из его k ближайших к нему объектов обучающей выборки.

1. Поясните основные принципы и этапы реализации метода kNN.

Основные принципы метода KNN: для классификации нового объекта алгоритм находит K объектов в обучающем наборе данных, которые находятся ближе всего к новому объекту. Затем новый объект классифицируется в тот класс, который является наиболее распространённым среди этих K соседей.

1. Поясните принцип выбора количества соседних объектов, по которым определяется принадлежность целевого объекта к результирующему классу.

Принцип выбора количества соседних объектов, по которым определяется принадлежность целевого объекта к результирующему классу, основан на компромиссе между точностью и обобщающей способностью модели. [1](https://loginom.ru/blog/knn)

В методе k-ближайших соседей (KNN) задаётся параметр алгоритма k, который задаёт число «соседей», которые будут использоваться при классификации. [1](https://loginom.ru/blog/knn) На первом этапе k выбирается произвольно, затем итеративно подбирается лучшее значение k на основе точности полученных прогнозов при каждом из выбранных k.

1. В чем заключается метод парзеновского окна?

В основе подхода лежит идея о том, что плотность выше в тех точках, рядом с которыми находится большое количество объектов выборки.

Если мощность множества элементарных исходов много меньше размера выборки, то в качестве восстановленной по выборке плотности мы вполне можем взять и гистограмму значений выборки.

В противном случае (например, непрерывном) данный подход не применим, так как плотность концентрируется вблизи обучающих объектов, и функция распределения претерпевает резкие скачки.

1. Поясните принцип метода потенциальных функций.

Принцип метода потенциальных функций заключается в том, что при классификации объект проверяется на близость к объектам из обучающей выборки. Считается, что объекты из обучающей выборки «заряжены» своим классом, а мера «важности» каждого из них при классификации зависит от его «заряда» и расстояния до классифицируемого объекта.

1. Назовите, какие параметры оптимизируют в методах kNN?

 параметры, которые оптимизируют в методах KNN:

* n\_neighbors. Основной параметр, который регулирует, сколько соседей следует проверять при классификации объекта. [1](https://holypython.com/knn/k-nearest-neighbor-optimization-parameters/)
* weights. Определяет, как вес должен распределяться между значениями соседей. По умолчанию вес распределяется равномерно среди всех значений соседей, а при значении «distance» — в зависимости от расстояния (ближе соседи имеют больший вес в алгоритме). [1](https://holypython.com/knn/k-nearest-neighbor-optimization-parameters/" \t "_blank)
* algorithm. Указывает алгоритм, который будет использоваться для вычисления ближайших соседей. [1](https://holypython.com/knn/k-nearest-neighbor-optimization-parameters/" \t "_blank)
* leaf\_size. [1](https://holypython.com/knn/k-nearest-neighbor-optimization-parameters/" \t "_blank)[2](https://www.programmingr.com/knn-hyperparameters-a-friendly-guide-to-optimization/" \t "_blank) Этот параметр влияет на производительность и размер алгоритма, используется при выборе алгоритмов BallTree или KDTree. [1](https://holypython.com/knn/k-nearest-neighbor-optimization-parameters/" \t "_blank)
* p. Параметр обозначает степень для расстояния Минковского. [1](https://holypython.com/knn/k-nearest-neighbor-optimization-parameters/" \t "_blank)
* n\_jobs. Указывает, сколько параллельных заданий разрешено выполнять одновременно для алгоритма соседей. [1](https://holypython.com/knn/k-nearest-neighbor-optimization-parameters/)