ФЕДЕРАЛЬНОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
Государственный университет «Дубна»

ИНСТИТУТ СИСТЕМНОГО АНАЛИЗА И УПРАВЛЕНИЯ

Кафедра системного анализа и управления

КУРСОВАЯ РАБОТА

по дисциплине

«Теория принятия решений»

ТЕМА: Принятие решений в задаче распознавания образов с использованием алгоритма *k* ближайших соседей

(наименование темы)

Выполнил: студент группы 2254

Харитонов Г.С.
(Ф.И.О.)

(подпись студента)

Руководитель:

преподаватель Бархатова И.А.
(ученая степень, ученое звание, занимаемая должность, ФИО)

Дата: _____

Оценка: _____

(подпись руководителя)

Дубна, 2023

Оглавление

Введение

Виноделие — это искусство, требующее глубоких знаний и опыта. Оценка качества вина является важным этапом производства, который включает в себя множество параметров, таких как аромат, вкус и цвет. В современном мире существует огромное количество различных сортов вин, и каждое из них по-своему уникально.

Одним из символов социалистического общества является ответственное отношение к потреблению и умение разбираться в том, что мы едим и пьем. В контексте напитков с использованием машинного обучения возможно достичь более объективной оценки и определения качества вина для народных масс.

Но как обычному человеку, без особый знаний, удостовериться, насколько качественным может быть вино? Как оценить это качество, не имея под рукой искусного сомелье? Тут на помощь приходит машинное обучение, с помощью которого можно формализовать данную задачу для обывателя.

В данной курсовой работе рассматривается применение алгоритма k ближайших соседей (kNN) для решения задачи распознавания качества вина. Будет строиться модель для определения класса вина (плохое, нормальное, хорошее) на основе характеристик, таких как содержание сахара, кислотности, алкоголя и других, описанных в теоретической части.

Лицо принимающее решение — человек, желающий употребить винный напиток. Или же человек, занимающийся оптовой закупкой винной продукции.

Постановка задачи

Цель: Определить качество вина.

Исходные данные: Анализ* физико-химических свойств выбранного вина. Для решения задачи будет использоваться материал из открытого источника, собранный на основе различных вариаций португальского вина "*Vinho Verde*" ("Винью Верде").

Ожидаемый результат: Программа для определения качества вина.

Критерий оценки: Совпадение результатов материала обучения с результатами материалов экзамена.

Определим, следующие уровни оценок вин:

- хорошее - 5-6
- нормальное 3-4
- плохое 1-2

Пользователю будет выдана оценка вина и один из трёх вышеописанных вердиктов.

*Анализ физико-химических свойств можно провести в специализированной лаборатории, например, здесь.

Теоретическая часть

Теория принятия решений — область исследования, вовлекающая понятия и методы математики, статистики, экономики, менеджмента и психологии с целью изучения закономерностей выбора людьми путей решения проблем и задач, а также способов достижения желаемого результата.

Гистограмма — способ представления табличных данных в графическом виде — в виде столбчатой диаграммы.

График рассеяния (точечная диаграмма) — показывает распределение элементов множества в плоскости между двумя переменными.

Диаграммы размаха («ящик с усами») — это удобный способ визуального представления групп числовых данных через квартили.

Пользовательский интерфейс — средства удобного и эффективного взаимодействия пользователя с устройствами компьютера. Пользовательский интерфейс представляет собой набор пиктограмм (иконок) для различных программ, кнопки для различных действий и окна, в которых отображаются данные при работе с приложением.

Корреляция — это статистическая взаимосвязь двух или более величин, параметров, значений.

Выброс (статистика) — результат измерения, не подпадающий под общее распределение.

Предобработка данных — предварительное отсеивание данных, не влияющих на результат или данных являющихся выбросами.

Тепловая карта (англ. *heatmap*) — графическое представление данных, где индивидуальные значения в таблице отображаются при помощи цвета.

Метод *k*-ближайших соседей (*k Nearest Neighbors*, или *kNN*) – популярный алгоритм классификации, который используется в разных типах задач машинного обучения.

Датасет — набор данных (графических или текстовых).

В качестве алгоритма классификации будет использоваться *kNN*, подразумевающий собой поиск ближайших соседей (наикратчайших расстояний до соседей) на основе подсчёта расстояний от объекта материала экзамена до всех объектов материала обучения.

Расстояние между объектами (сумма квадратов разностей всех свойств двух вин):

$\sqrt{\sum_1^n (p_i - q_i)^2}$

"Взвешенное" евклидово расстояние между объектами (сумма квадратов разностей всех свойств двух вин, но с учётом заданных весов):

$$\sqrt{\sum_{1}^{n} w_i * (p_i - q_i)^2}$$

Исходные атрибуты вина:

Fixed Acidity (Фиксированная кислотность): большинство кислот, входящих в состав вина, либо фиксированные, либо нелетучие (не испаряются легко).

Volatile Acidity (Летучая кислотность): количество уксусной кислоты в вине, слишком высокое содержание которой может привести к неприятному уксусному привкусу.

Citric Acid (Лимонная кислота): содержащаяся в небольших количествах лимонная кислота может придать винам "свежесть" и аромат.

Residual Sugar (Остаточный сахар): количество сахара, остающегося после прекращения ферментации, редко встречается в винах с содержанием менее 1 грамма на литр.

Chlorides (Хлориды): количество соли в вине.

Free Sulfur Dioxide (Свободный диоксид серы): свободная форма SO2 существует в равновесии между молекулярным SO2 (в виде растворенного газа) и бисульфит-ионом.

Total Sulfur Dioxide (Общий диоксид серы): количество свободной и связанной форм S02; в низких концентрациях SO2 в основном не обнаруживается в вине, но в свободном SO2.

Density (Плотность): плотность воды близка к плотности обычной воды в зависимости от процентного содержания алкоголя и сахара.

PH: описывает, насколько кислым или основным является вино по шкале от 0 (очень кислое) до 14 (очень простое); большинство вин находятся в диапазоне 3-4.

Sulphates (Сульфаты): добавка к вину, которая может способствовать повышению уровня диоксида серы (S02), который действует как антимикробное средство.

Alcohol (Алкоголь): процент содержания алкоголя.

Quality (Качество): используется в качестве класса (оценка от 1 до 6) определения качества вина.

Практическая часть

Предобработка данных

Первым делом анализируются данные и происходит избавление от лишних атрибутов (если такие есть), которые не будут влиять на результат исследования.

Для этого создаётся *Notebook* в *Google colab.* Рассматривается количество вин разных классов (в исходном датасете).

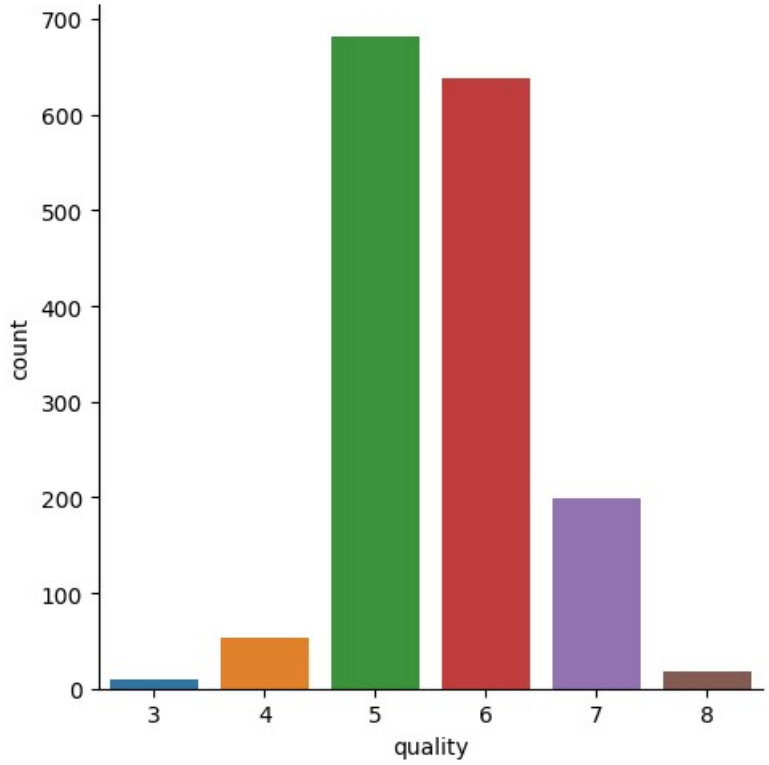

Рис. 1. Количество вин

Преобладают вина 5 и 6 (они же вина 3 и 4-ой категории, то есть нормальной, если смотреть на конечный результат).

Наблюдение закономерностей и рассмотрение гистограммы, графика рассеяния, графика размаха и столбчатой диаграммы.

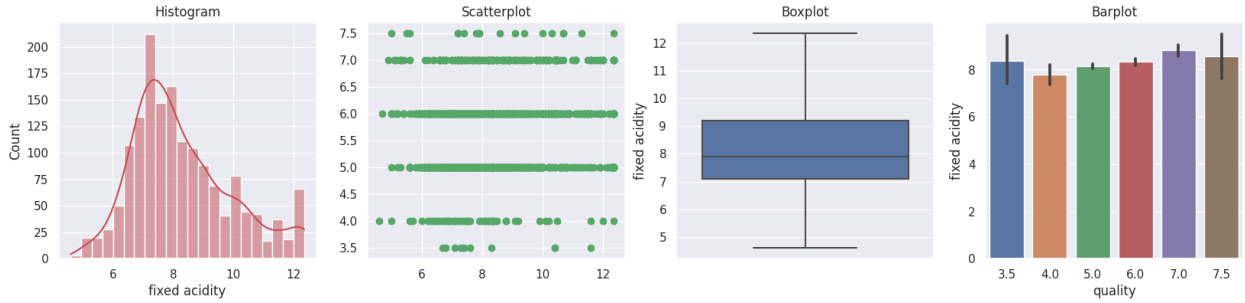

Рис. 2. Графики *fixed acidity*

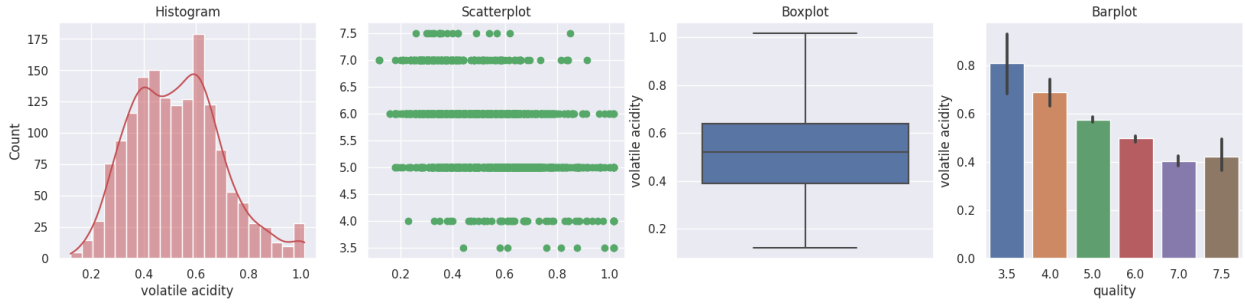

Рис. 3. Графики *volatile acidity*

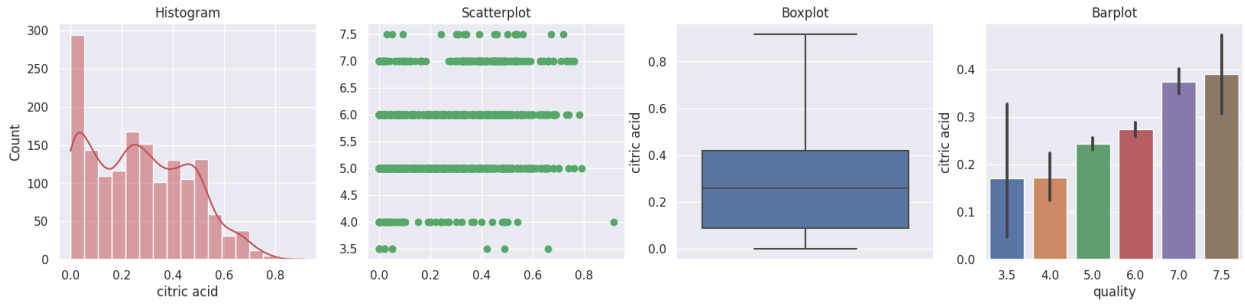

Рис. 4. Графики *citric acid*

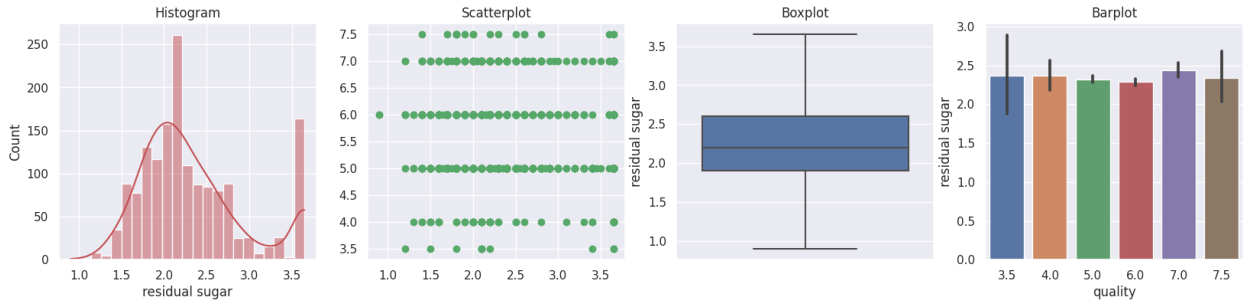

Рис. 5. Графики *residual sugar*

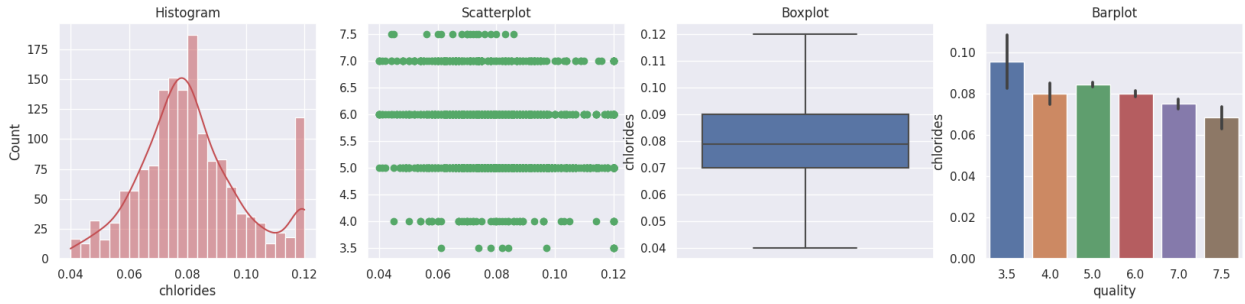

Рис. 6. Графики *chlorides*

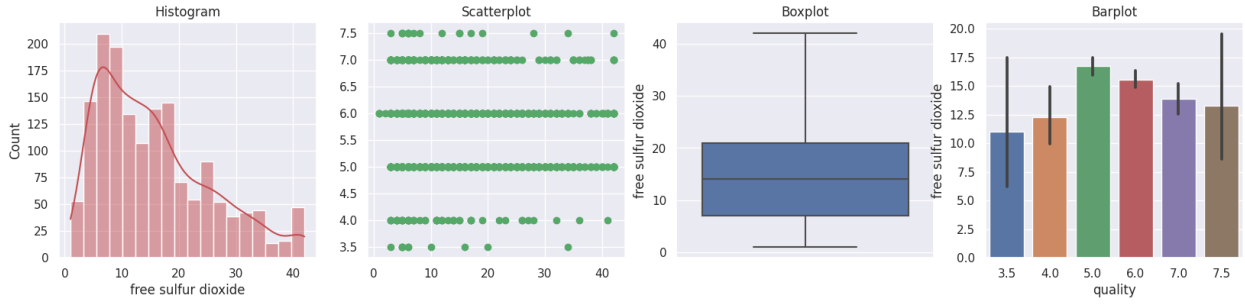

Рис. 7. Графики *free sulfur dioxide*

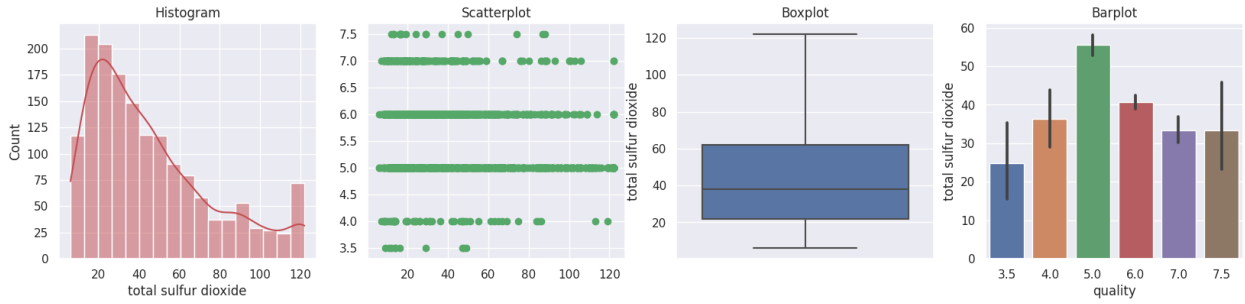

Рис. 8. Графики *total sulfur dioxide*

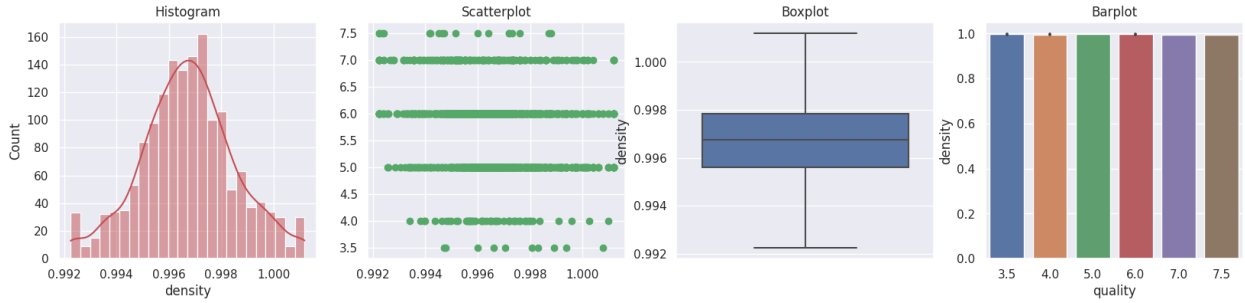

Рис. 9. Графики *density*

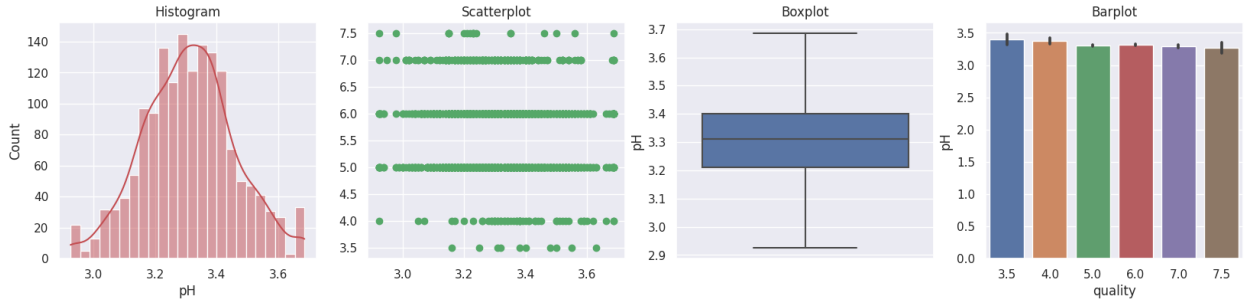

Рис. 10. Графики *pH*

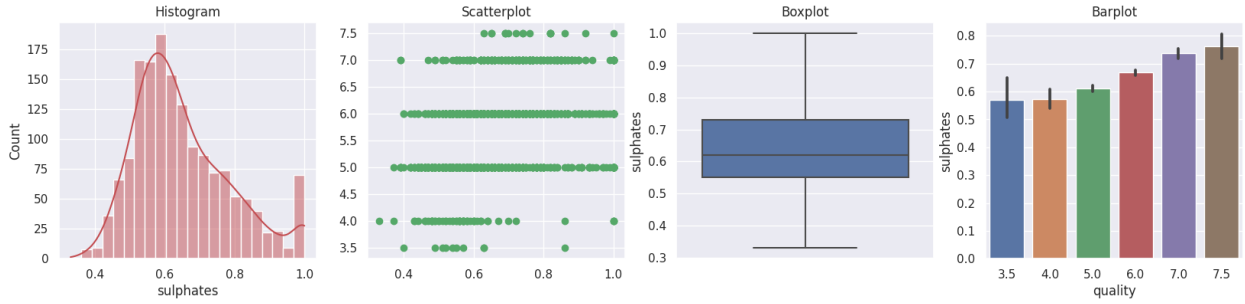

Рис. 11. Графики *sulphates*

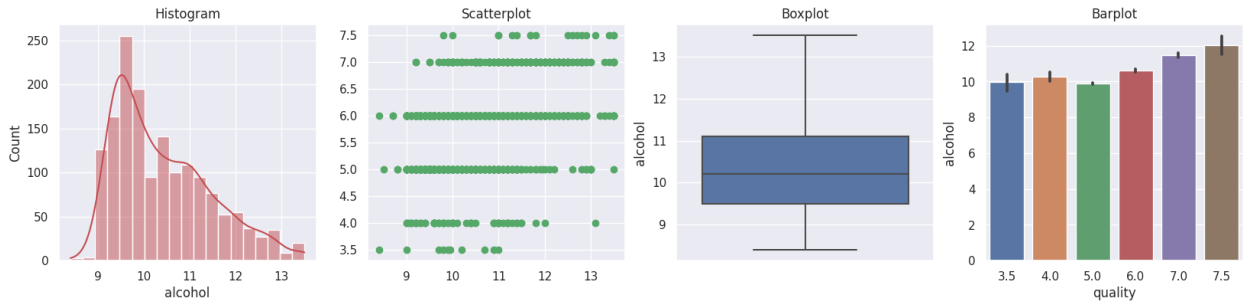

Рис. 12. Графики *alcohol*

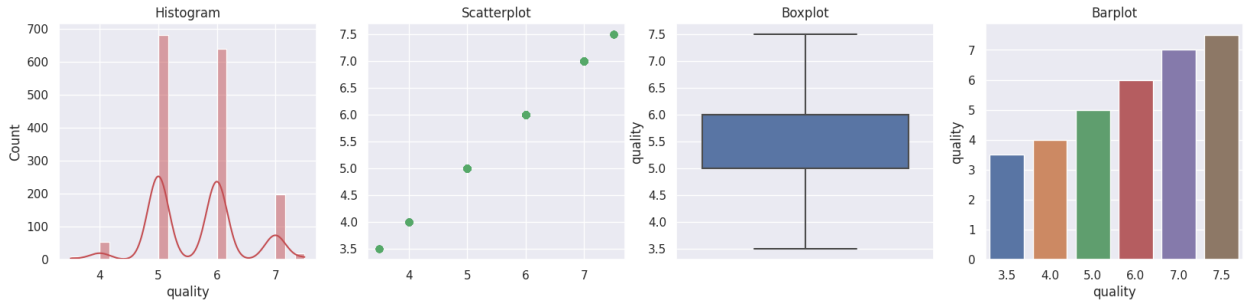

Рис. 13. Графики *quality*

По данным графикам можно сделать следующие выводы:

- Фиксированная кислотность не дает никаких указаний для классификации качества;
- Заметная тенденция к снижению летучей кислотности по мере того, как мы повышаем качество;
- Состав лимонной кислоты повышается по мере того, как мы повышаем качество получаемого продукта;
- Содержание хлоридов также снижается по мере того, как мы повышаем качество вина;
- Уровень сульфатов повышается вместе с качеством вина;
- Уровень алкоголя также повышается по мере повышения качества вина.

В наборе данных много выбросов, и поэтому они будут оптимизироваться их до желаемых значений верхней и нижней границ. А также удаляются повторы данных, которые можно заметить здесь:

```
[9]  # Check For Missing Values
     info = pd.DataFrame(data.isnull().sum(),columns=["IsNull"])
     info.insert(1,"IsNa",data.isna().sum(),True)
     info.insert(2,"Duplicate",data.duplicated().sum(),True)
     info.insert(3,"Unique",data.nunique(),True)
     info.insert(4,"Min",data.min(),True)
     info.insert(5,"Max",data.max(),True)
     info.T
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
IsNull	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.00000	0.00	0.00	0.0	0.0
IsNa	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.00000	0.00	0.00	0.0	0.0
Duplicate	240.0	240.00	240.0	240.0	240.000	240.0	240.0	240.00000	240.00	240.00	240.0	240.0
Unique	96.0	143.00	80.0	91.0	153.000	60.0	144.0	436.00000	89.00	96.00	65.0	6.0
Min	4.6	0.12	0.0	0.9	0.012	1.0	6.0	0.99007	2.74	0.33	8.4	3.0
Max	15.9	1.58	1.0	15.5	0.611	72.0	289.0	1.00369	4.01	2.00	14.9	8.0

Рис. 14. Повторения

После избавления от выбросов (Рис. 15) были построены графики во всё том же *Notebook*.

```
#Get rid of emissions
def mod_outlier(df):
        df1 = df.copy()
        df = df._get_numeric_data()
        q1 = df.quantile(0.25)
        q3 = df.quantile(0.75)

        iqr = q3 - q1

        lower_bound = q1 -(1.5 * iqr)
        upper_bound = q3 +(1.5 * iqr)


        for col in df.columns:
            for i in range(0,len(df[col])):
                if df[col][i] < lower_bound[col]:
                    df[col][i] = lower_bound[col]

                if df[col][i] > upper_bound[col]:
                    df[col][i] = upper_bound[col]


        for col in df.columns:
            df1[col] = df[col]

        return(df1)

df1 = mod_outlier(df)

for col in df1:
    diagnostic_plots(df1,col,'quality')

#Drop dublicates
df2=df1.drop_duplicates()
```

Рис. 15 Очистка

Далее проведён анализ и для окончательного понимания того, можно ли выбросить какие-то свойства из исходного набора данных. Будет построена корреляционная таблица на основе качества и рассмотрено наибольшее и наименьшее влияние атрибутов.

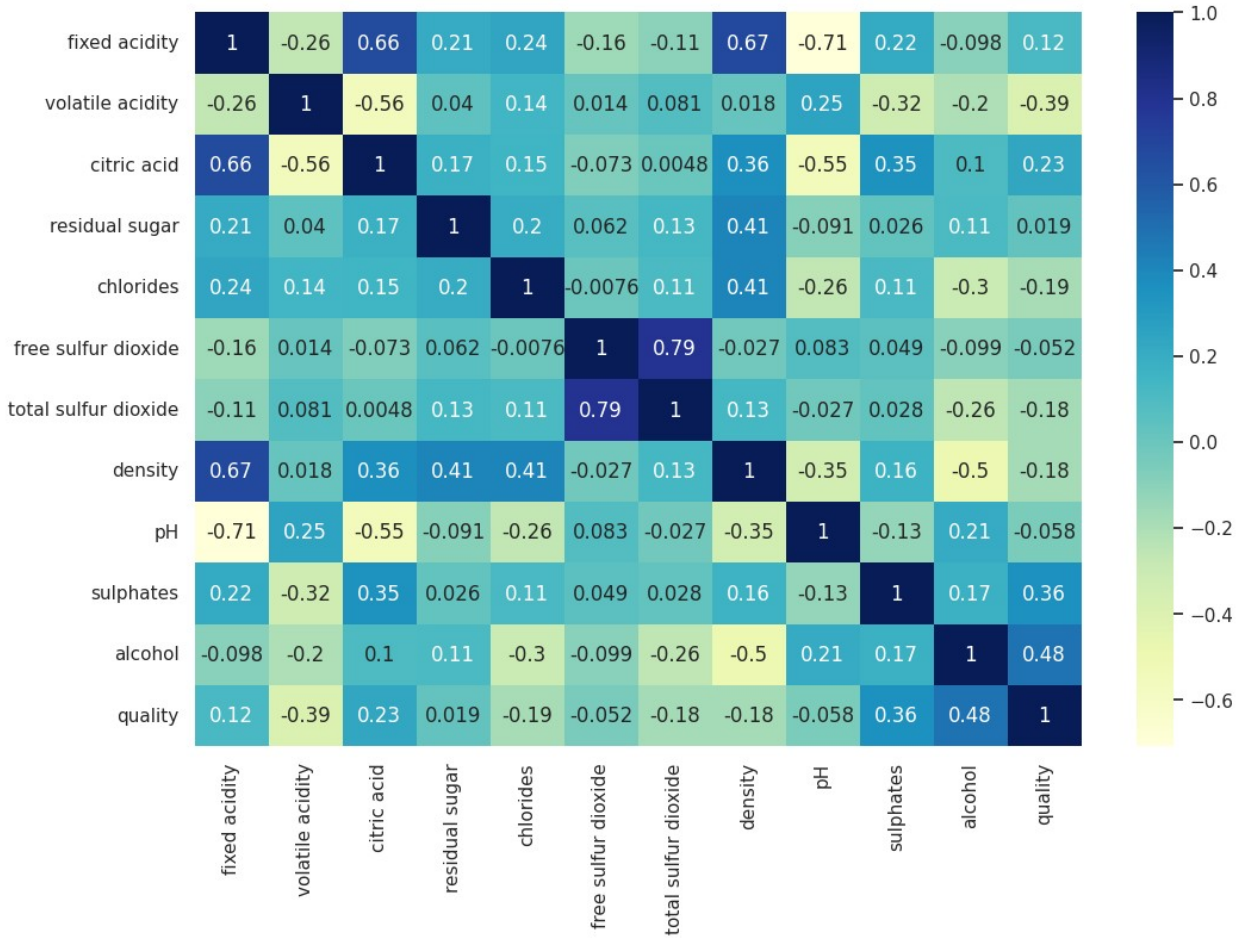

Рис. 16. Корреляционная таблица

Далее сортируются все зависимости по убыванию:

```
In [15]:
correlation = df2.corr()
print(correlation['quality'].sort_values(ascending = False),'\n')
```

```
quality                 1.000000
alcohol                 0.483417
sulphates               0.355137
citric acid             0.230049
fixed acidity           0.116810
residual sugar          0.018961
free sulfur dioxide    -0.051531
pH                     -0.058199
density                -0.177360
total sulfur dioxide   -0.177391
chlorides              -0.187745
volatile acidity       -0.392559
Name: quality, dtype: float64
```

Рис. 17. Отсортированные зависимости

Представление коэффициентов корреляции в виде графика:

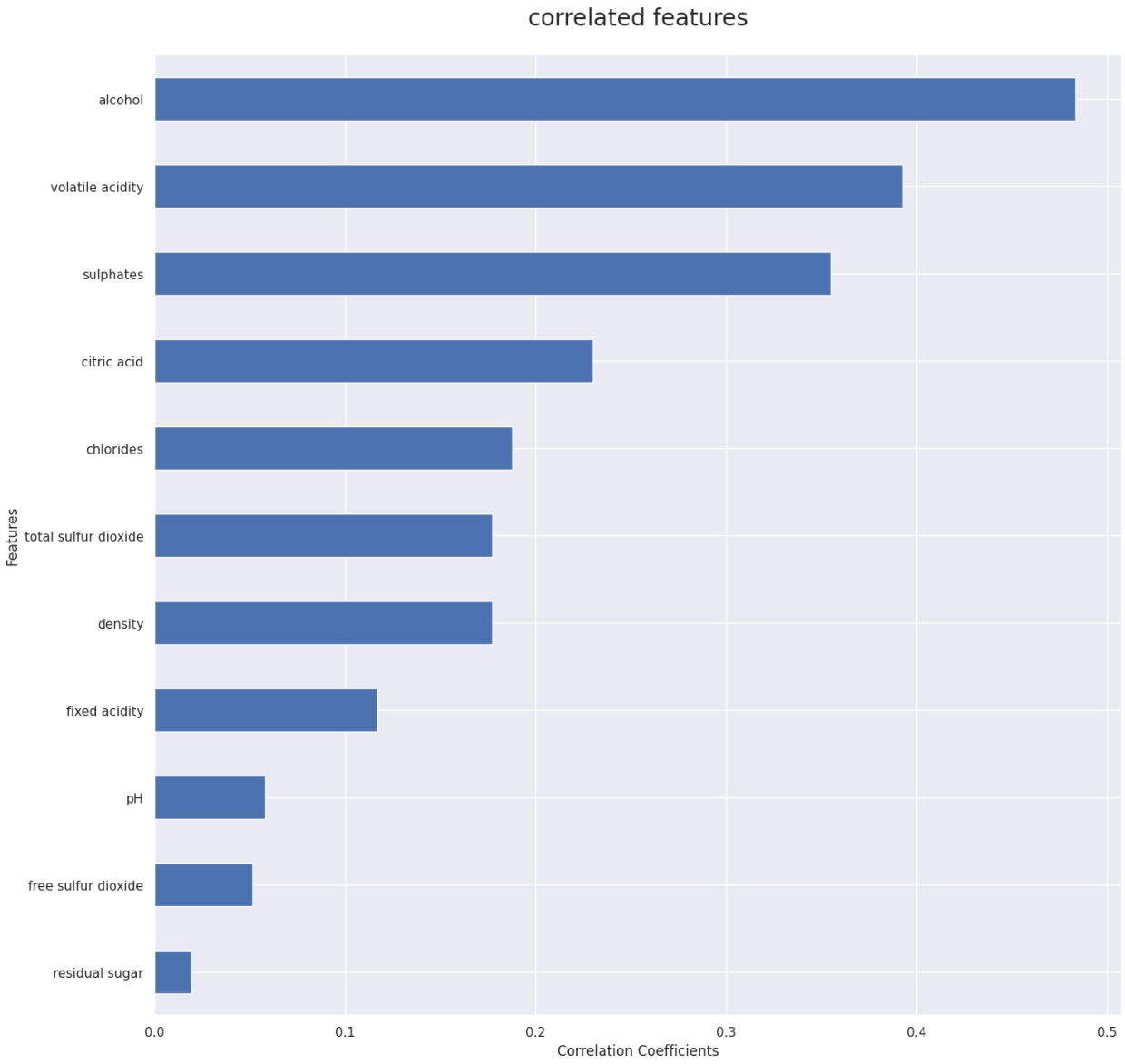

Рис. 18. Коэффициенты корреляции

Очевидно, что три последних атрибута (*pH, free sulfur dioxide* и *residual sugar*) имеют наименьшее влияние на качество вина, а если смотреть на их коэффициенты корреляции, то они близки к сотым, значит, их можно было бы выбросить, но вместо этого можно пересмотреть подход к поиску расстояния и использовать "взвешенное" евклидово расстояние, где в качестве веса будут использоваться значения коэффициентов корреляции. Однако некоторые коэффициенты отрицательные (расстояние не может быть отрицательным), поэтому их необходимо сделать положительными, а уже положительные умножить на 2, чтобы сохранить их значимость.

Обучение

Производится разбиение исходных материалов на две части: тренировочные и экзаменационные в соотношении 1:3. Получим следующее количество по классам:

Classes	1	2	3	4	5	6
All	10	53	576	535	165	17
Train	8	40	432	401	124	13
Exam	2	13	144	134	43	4

Таблица. 1. Разбиение материала

Параметр k будет определяться следующим образом: берётся любое k от 1 до n, где n — количество элементов выборки. Это k является подходящим тогда и только тогда, когда k % количество классов = 1, в противном случае возникает неопределённость. В нашем случае k % 6 = 1.

Находится процент распознавания классов вина путём деления количества совпадений на количество экзаменационных материалов:

```
259  func getProcentQuality(wineTrainList []wine_sort, weights wine_sort) float64 {
260      wineExamList := getWineList("proc_exam_materials.csv")
261      var countExam = float64(len(wineExamList))
262      var countMatches float64
263      for _, wine := range wineExamList {
264          distances := allDistances(wine, weights, wineTrainList)
265          class := kNNClassify(7, distances)
266          if class == wine.Quality {
267              countMatches++
268          }
269      }
270      return countMatches / countExam * 100
271  }
```

Рис. 19. Функция процента распознавания класса

```
gleb@rotten-pc:~/Desktop/University/DT/DT_coursework$ go run parse.go main.go
72.86%
```

Рис. 20. Результат процесса распознавания класса

Конечный результат: должен быть одним из 3-х оценок, в зависимости от класса вина. Далее рассмотрено, сколько вин попадают в свою категорию:

```go
func getProcentCategory(wineTrainList []wine_sort, weights wine_sort) float64 {
    wineExamList := getWineList("proc_exam_materials.csv")
    var countExam = float64(len(wineExamList))
    var countMatches float64
    for _, wine := range wineExamList {
        var examQuality string
        switch wine.Quality {
        case 1, 2:
            examQuality = "bad"
        case 3, 4:
            examQuality = "normal"
        case 5, 6:
            examQuality = "good"
        default:
            fmt.Printf("out of range \n", examQuality)
        }
        distances := allDistances(wine, weights, wineTrainList)
        class := kNNClassify(7, distances)
        var predictedQuality string
        switch class {
        case 1, 2:
            predictedQuality = "bad"
        case 3, 4:
            predictedQuality = "normal"
        case 5, 6:
            predictedQuality = "good"
        default:
            fmt.Printf("out of range \n", predictedQuality)
        }
        if examQuality == predictedQuality {
            countMatches++
        }
    }
    return countMatches / countExam * 100
}
```

Рис. 21. Функция процента распознавания категории

```
gleb@rotten-pc:~/Desktop/University/DT/DT_coursework$ go run parse.go main.go
87.61%
```

Рис. 22. Результат процесса распознавания категории

Тестирование

Далее будут рассмотрены случаи ошибок и совпадений:

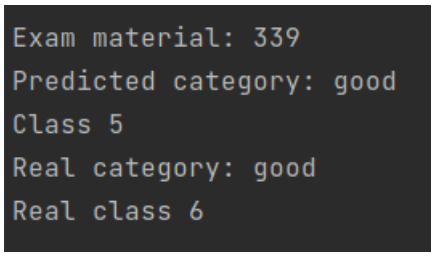

Рис. 23. Совпадение категории и ошибка предсказания класса

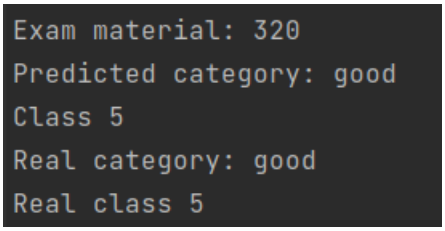

Рис. 24. Совпадение категории и класса

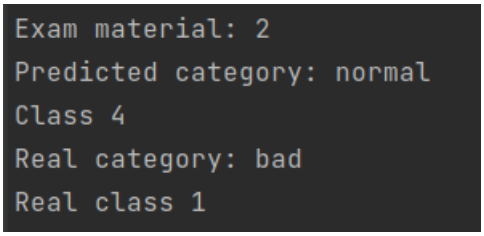

Рис. 25. Ошибка предсказания класса и категории

Замечено, что наблюдаются ошибки второго рода, однако отклонений на 2 категории не наблюдалось.

Заключение

В данной курсовой работе был использован алгоритм распознавания *k*-ближайшего соседа для решения многокритериальной задачи на реальных данных. С помощью средств аналитики была произведена обработка первичных данных, в результате которой датасет стал максимально независим для материалов обучения и экзамена, также при исследовании данных были получены весовые коэффициенты для подсчёта расстояний между точками.

В результате проведенной работы были улучшены знания и понимание решений многокритериальных задач, а также получен опыт программирования на языке *Go*.

Работа выложена на *GitHub* и добавлена в портфолио автора.

Список литературы и интернет-ресурсов

1) https://colab.research.google.com/drive/1Xd4e3rhXDHjlxWd9rypOZsK__ZbhOuKA?usp=sharing (Анализ графиков в Google colab)

2) https://habr.com/ru/articles/149693/ (Статья про алгоритм *kNN*)

3) https://prse.ru/ekspertizy/tovarovedcheskie-ekspertizy/ekspertiza-alkogolnyh-napitkov (Экспертиза вина)

4) https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009 (Используемый датасет)

5) https://go.dev/doc/ (Документация по Golang)

6) https://seaborn.pydata.org/ (Документация по отрисовке графиков в Python)

7) https://github.com/Ser1ousSAM/DT_coursework (Ссылка на программу)