Московский авиационный институт

(Национальный исследовательский университет)

Институт: «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»
Дисциплина: «Методы, средства и технологии мультимедиа»

Курсовая работа

Студент: Будникова В. П.
Преподаватель: Вишняков Б. В.
Группа: М8О-407Б-19
Дата:
Оценка:
Подпись:

Москва, 2022

Задача:

1. Выбрать задачу(классификация или регрессия)
2. Выбрать датасет
3. Сделать описание датасета
4. Сделать и описать предпроцессинг над датасетом
5. Выбрать алгоритм
6. Реализовать алгоритм
7. Сделать описание алгоритма
8. Выбрать метрики качества
9. Продемонстрировать полученные результаты
10. Сравнить результат алгоритма с алгоритмом из sklearn

Описание датасета

Датасет: Heart Attack Data Sets
Ссылка: https://www.kaggle.com/datasets/sumittemplatic/heart-attack-data-sets

Данный набор данных содержит такие характеристики состояния пациента как возраст, пол, информация частоте сердечных сокращений, боли в груди, а также результаты стенокардии и др. параметры.

Характеристики датасета:

- Age : Возраст пациента
- Sex : Пол пациента
- exang: Стенокардия, вызванная физической нагрузкой (1 = да; 0 = нет)
- ca: Количество крупных сосудов (0-3)
- cp : Тип боли в грудной клетке
    - Value 1: типичная стенокардия
    - Value 2: атипичная стенокардия
    - Value 3: неангинальная боль
    - Value 4: неангинальная боль
- trtbps : кровяное давление в состоянии покоя (в мм рт.ст.)
- chol : холесторальный в мг/дл, полученный через датчик ИМТ
- fbs : (устой в крови > 120 мг/дл) (1 = больше значения; 0 = меньше значения)
- rest_ecg : результаты электрокардиографии в состоянии покоя.
    - Value 0: норма
    - Value 1: наличие аномалии ST-T волны (инверсии T волны и/или повышение или понижение ST > 0,05 мВ)
    - Value 2: наличие вероятной или определенной гипертрофии левого желудочка по критериям Эстеса
- thalach : максимальная достигнутая частота сердечных сокращений
- target : 0 = маленькая вероятность сердечного приступа 1 = большая вероятность сердечного приступа

Оборудование:

Ноутбук, процессор: 1,6 GHz 2ядерный процессор Intel Core i5, память 8ГБ.

Подготовка данных:

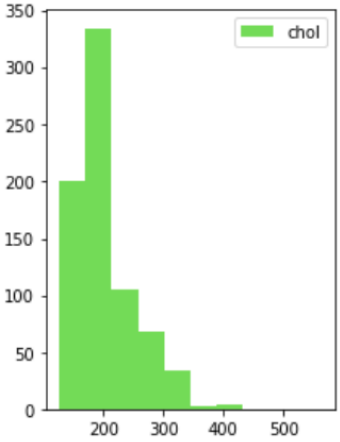
После загрузки данные очищаются от параметров, которые не влияют и не должны влиять на результат, например, таких как «индекс». Далее проводится предпроцессинг данных.

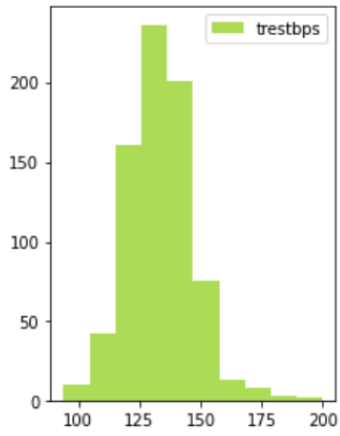
Для начала необходимо проверить, если ли отсутствующие(пропущенные) значения в данном датасете. После проверки оказалось, что пропущенных данных в датасете не присутствует, следовательно можно пропустить этап очистки от пропусков или заполнения значениями пропусков.

Посмотрим на описание значений данных.

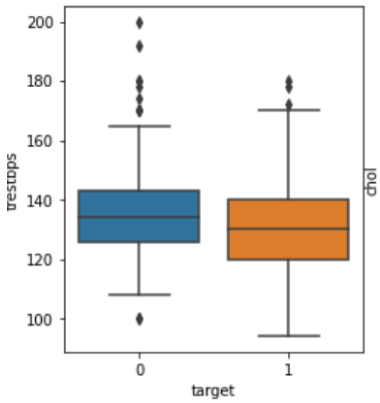
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000	752.000000
mean	64.962766	0.569149	0.666223	133.417553	203.336436	0.343085	0.514628	133.884309	0.421543	1.381004	0.561170	0.886968	1.507979	0.218085
std	10.424221	0.495525	0.797150	13.167410	49.542620	0.475056	0.510658	22.383087	0.494135	1.237566	0.788724	0.909626	0.989931	0.413220
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	58.000000	0.000000	0.000000	125.000000	169.000000	0.000000	0.000000	115.000000	0.000000	0.364012	0.000000	0.000000	1.000000	0.000000
50%	70.000000	1.000000	0.000000	132.000000	188.000000	0.000000	1.000000	132.000000	0.000000	1.101973	0.000000	1.000000	2.000000	0.000000
75%	72.000000	1.000000	1.000000	142.000000	227.250000	1.000000	1.000000	147.000000	1.000000	2.102197	1.000000	2.000000	2.000000	0.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.772167	2.000000	4.000000	3.000000	1.000000

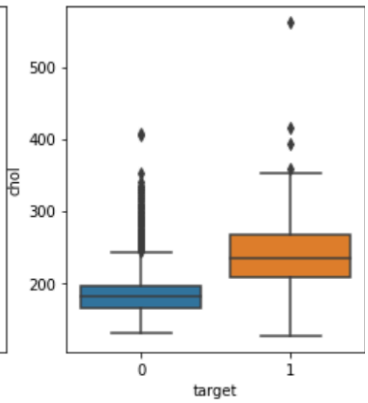
Как можно заметить из таблицы, параметр chol имеет максимальное значение 564, но 75% данных в среднем имеют значения в два раза меньше, это может свидетельствовать о выбросах. Похожая ситуация происходит с некоторыми другими параметрами. Посмотрим на гистограмму этих характеристик.



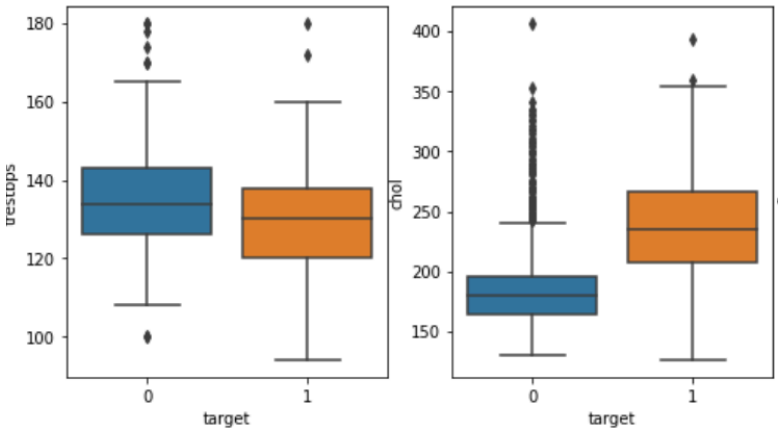


Воспользуемся boxplot, для наглядной визуализации:

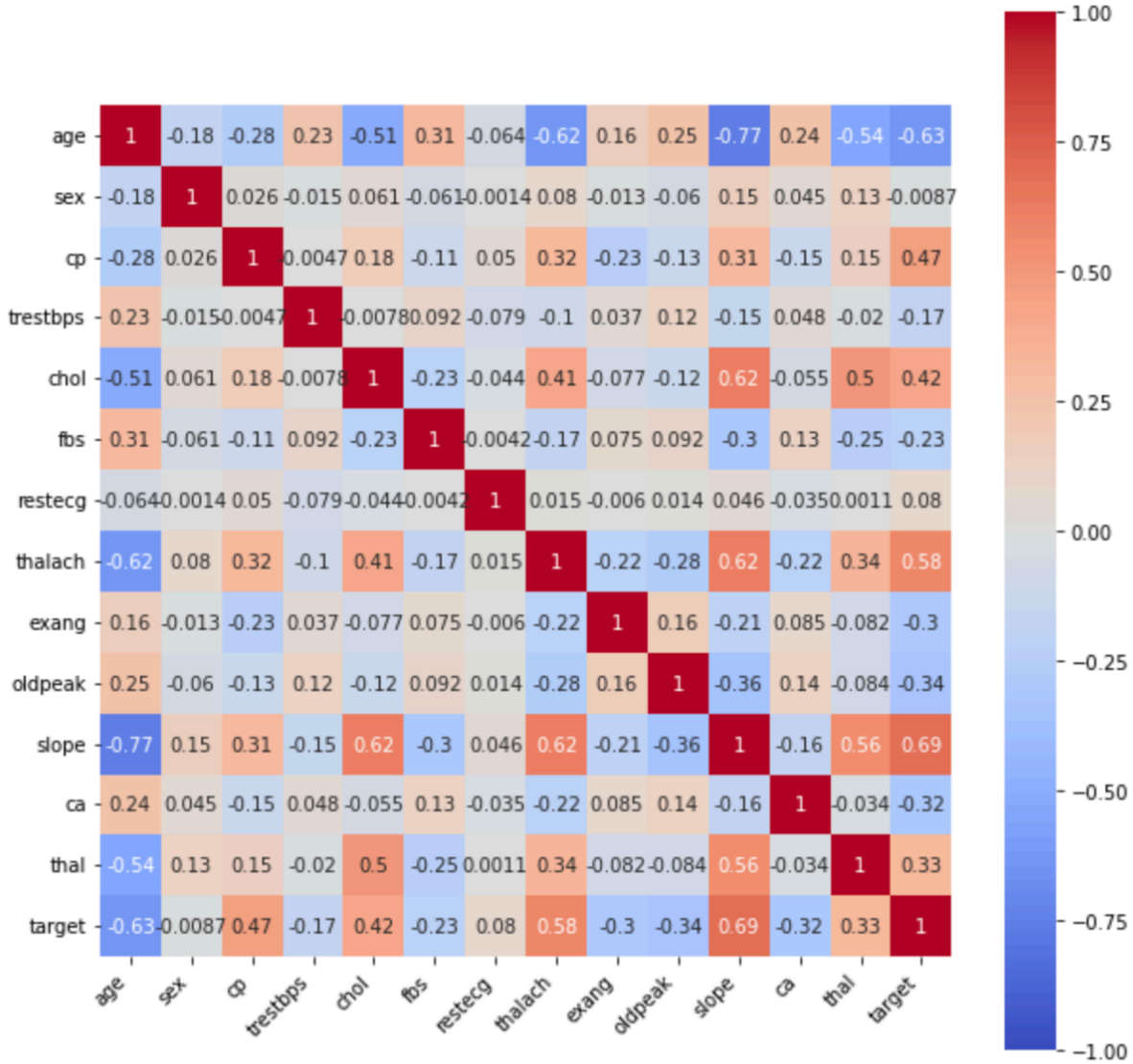




Удалим выбросы и посмотрим на результат:



Посмотрим на матрицу корреляции данных датасета:



Данные матрицы показывают, насколько сильно одни значения характеристик влияют на другие. Если характеристики имеют сильную прямую или обратную связь, то нахождение этих характеристик вместе в одном датасете при расчетах излишне, можно удалить из рассмотрения одну из таких величин.
В корреляционной матрице для данного датасета значений близких к 1 нет, а значит и нет характеристик, обладающих сильной линейной связью.
Также можно обратить внимание на то, что есть такие характеристики, значение которых в корреляционной матрице по отношению к результату близко к нулю, это свидетельствует о том, что данные характеристики напрямую почти не влияют на результат. Но данные характеристики могут косвенно влиять на результат, если их значение матрице будет отлично от 0 по отношению к характеристикам, сильно или слабо влияющих на ответ.

Перед тем, как разделить данные на выборки для обучения и тестирования, посмотрим на количество данных, которые принадлежат каждому из классов:

```
target
0    588
1    164
dtype: int64
```

Как можно видеть, данные несбалансированные, количество данных, принадлежащих второму классу в несколько раз меньше, чем к первому.
Сгенерируем искусственные данные с помощью GaussianCopula.

```
target
0    588
1    402
dtype: int64
```

Далее разделим данные на тренировочные и тестовые.

Описание алгоритма:

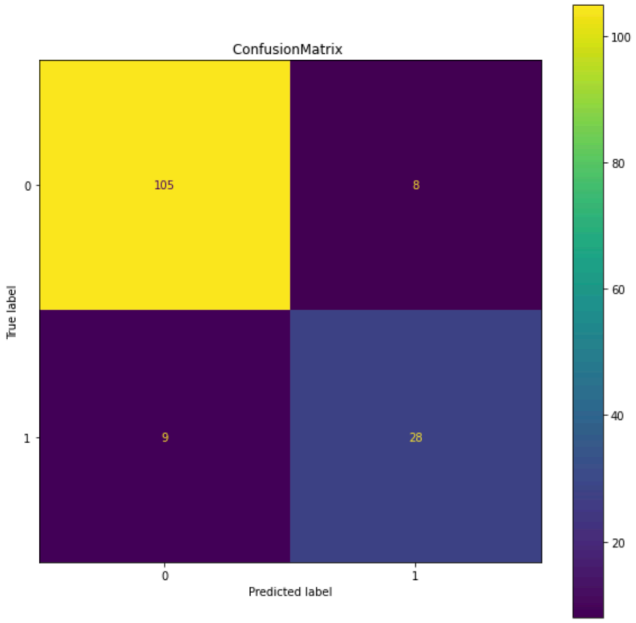
Для данной курсовой работы я решила выбрать алгоритм KNN - Алгоритм k ближайших соседей. Данный алгоритм запоминает обучающую выборку при обучении. Далее, когда нам необходимо предсказать результат, алгоритм считает расстояние между каждой характеристикой входного экземпляра и экземпляра из обучающей выборки соответственно. После подсчета, определяется k ближайших соседей, т. е. экземпляров, наиболее близких к входному. После этого выбирается наиболее часто-встречающийся из этих «соседей».

Реализация:

```
class KNN():
    def __init__(self, k, countClasses):
        self.k = k
        self.countClasses = countClasses

    def dist (self, a, b):
        return math.sqrt(np.sum(np.power(a - b, 2)))

    def fit(self, data, labels):
        self.data = data
        self.labels = labels

    def predict(self, data):
        predLables = []

        for i, d_i in enumerate(data):
            distance = list()
            for j, d_j in enumerate(self.data):
                distance.append([self.dist(d_j, d_i), self.labels[j]])

            stat_classes = np.zeros(self.countClasses)
            sort_dist = sorted(distance)
            for d in sort_dist[0:self.k]:
                stat_classes[d[1]] += 1

            predLables.append(np.argmax(stat_classes))

        return predLables

    def pred_proba(self, data):
        predLables = []
        s = Softmax()
        for i, d_i in enumerate(data):
            distance = list()
            for j, d_j in enumerate(self.data):
                distance.append([self.dist(d_j, d_i), self.labels[j]])

            stat_classes = np.zeros(self.countClasses)
            sort_dist = sorted(distance)
            for d in sort_dist[0:self.k]:
                stat_classes[d[1]] += 1

            predLables.append(stat_classes / np.max(stat_classes))

        return s.Next(np.array(predLables))
```

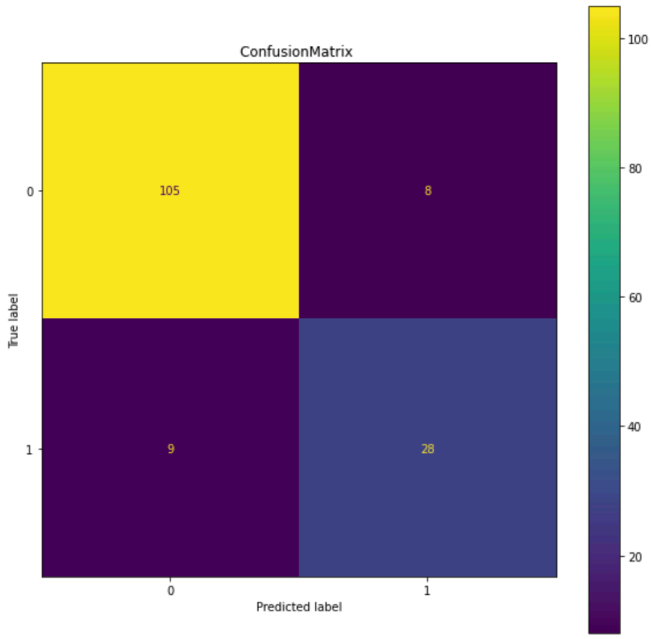
Результаты работы:
Значение k = 5

```
Accuracy: 0.8866666666666667
Precision: 0.7567567567567568
Recall: 0.7777777777777778
F-metric 0.7671232876712328
```



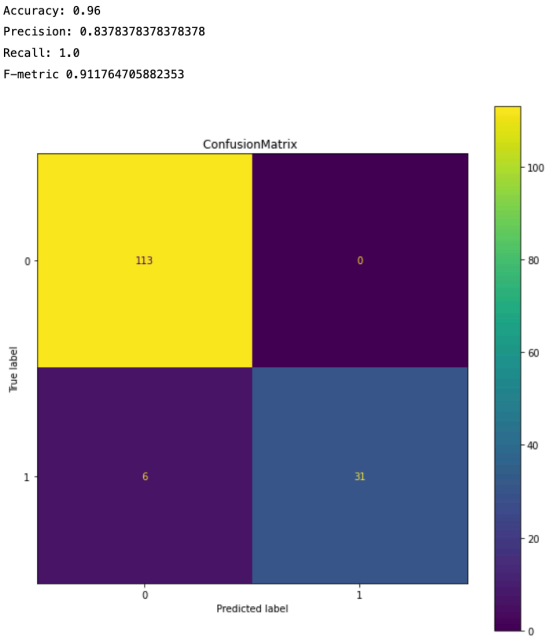
Алгоритм из sklearn:

```
Accuracy: 0.8866666666666667
Precision: 0.7567567567567568
Recall: 0.7777777777777778
F-metric 0.7671232876712328
```

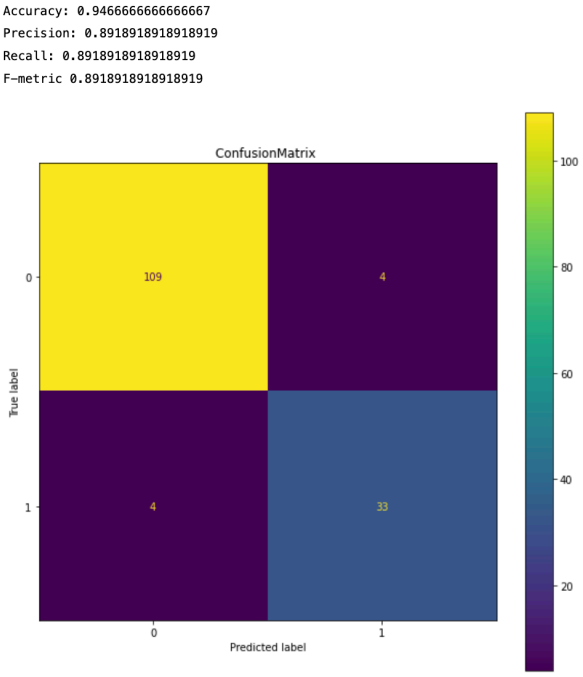


Как можно увидеть из результатов работы алгоритм показал хорошую точность. Аналогичная точность у алгоритма из sklearn. Попробуем увеличить точность нашего алгоритма, используя дополнительные алгоритмы. Я решила использовать алгоритмы мягкого и жесткого голосования.
Алгоритм жесткого голосования использует несколько алгоритмов KNN с различными параметрами k. Далее после получения ответа из всех алгоритмов, жесткое голосование берет в качестве своего ответа наиболее часто встречающийся.
Алгоритм мягкого голосования использует вероятности в своей реализации, поэтому было решено добавить в алгоритм KNN модификацию, которая выдает не само значение класса, а пару значений - вероятности принадлежности к каждому из классов. После получения вероятностей от каждой модели мягкое голосование принимает решения в пользу такого класса, суммарная вероятность которого наибольшая.

Результаты жесткого голосования:



Результаты мягкого голосования:



Из результатов алгоритмов можно увидеть, что точность возросла, по сравнению с простым KNN.

Выводы:

В данной работе был проведен анализ датасета. Были представлены гистограммы, описание и матрица корреляций данных. Также был реализован алгоритм KNN, алгоритм жесткого и мягкого голосования, посчитаны метрики всех алгоритмов и использован алгоритм из sklearn для сравнения точности.