|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»** |
| **РТУ МИРЭА** |
| Институт кибербезопасности и цифровых технологий |
| *(наименование института, филиала)* |
|  |
| Кафедра информатики |
| *(наименование кафедры)* |

**Курсовая работа**

|  |  |
| --- | --- |
| по дисциплине | **Основы машинного обучения** |
|  | *(наименование дисциплины)* |

|  |  |
| --- | --- |
| **Тема курсовой работы** | **Реализация алгоритмов классификации** |
| **на языке Python** | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Студент группы** | Мустафин Александр Русланович БФБО-01-22 |  |  |
|  | *(учебная группа, фамилия, имя, отчество студента)* |  | *(подпись студента)* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Руководитель курсовой работы** | зав. каф., к.ф.-м.н., доцент, Шмелева А.Г. |  |  |
|  | *(должность, звание, ученая степень, ФИО)* |  | *(подпись руководителя)* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Рецензент** (при наличии) | – |  | – |
|  | *(должность, звание, учёная степень)* |  | *(подпись рецензента)* |

Работа представлена к защите « »2023 г.

Допущен к защите « »2023 г.

|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»** |
| **РТУ МИРЭА** |
| Институт кибербезопасности и цифровых технологий |
| *(наименование института, филиала)* |
| Кафедра информатики |
| *(наименование кафедры)* |

Утверждаю

заведующий кафедрой информатики

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шмелева А.Г.

*Подпись ФИО*

«\_\_\_\_» \_\_\_\_\_\_\_ 20\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение курсовой работы по дисциплине**

**«Основы машинного обучения»**

Студент Мустафин Александр Русланович Группа БФБО-01-22

**Тема:** **«**Реализация алгоритмов классификации на языке Python»

**Исходные данные:** Датасет «Набор данных о раке молочной железы Висконсин»

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:** Изучения библиотек NumPy, Pandas, Matplotlib, Seaborn и Scikit-learn на языке Python, метод ближайших соседей и методы классификации

**Срок предоставления к защите курсового проекта (работы):**  до «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**Задание на курсовой проект (работу) выдал \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

*Подпись руководителя Ф.И.О. руководителя*

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ г. ***Задание на курсовой проект(работу) получил (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)***

*Подпись обучающегося Ф.И.О. исполнителя*.

Оглавление

[Введение 4](#_Toc133424123)

[1. Теоретические аспекты реализации алгоритмов классификации. 6](#_Toc133424124)

[1.1. Основные определения реализации алгоритмов классификации. 6](#_Toc133424125)

[1.2. Описание алгоритма и приведение примера решения модельной задачи с описанием ее. 11](#_Toc133424126)

[2. Практическое применение реализации алгоритмов классификации данных. 16](#_Toc133424127)

[2.1. Постановка задачи с помощью dataset (Набор данный о раке молочной железы Висконсин). 16](#_Toc133424128)

[2.2. Реализация задачи с помощью dataset (Набор данных о раке молочной железы Висконсин). 17](#_Toc133424129)

[Заключение 20](#_Toc133424130)

[Список использованных источников 21](#_Toc133424131)

[Приложение 1. Листинг программы 22](#_Toc133424132)

Введение

В данной курсовой работе мы рассмотрим тему «Реализация алгоритмов классификации на языке Python». При изучении данной темы мы рассмотрим методы классификации, методы ближайших соседей, библиотеки NumPy, Pandas, Scikit-learn, Seaborn и Matplotlib, а также dataset «Набор данных о раке молочной железы Висконсин» по которому мы будем писать нашу программу, но для начала узнаем, что такое машинное обучение и методы ближайших соседей.

Тема реализации алгоритмов классификации всегда актуальна, так как классификация – это одна из основных задач машинного обучения. Алгоритмы классификации используются для решения многих задач, таких как определение категории товара на основе его описания, определение темы текстового документа, распознавание рукописного текста и т.д. Реализация алгоритмов классификации может быть полезна для разработки различных приложений, таких как системы автоматической классификации документов, системы фильтрации спама, системы рекомендации товаров и так далее.

Цель данной работы изучить библиотеки Scikit-learn, Pandas, NumPy, Matplotlib, Seaborn, а также научиться работать с классификацией и один из его типов – это метод ближайших соседей.

Машинное обучение – наука, изучающая алгоритмы и статистические модели для эффективного выполнения конкретной задачи без использования явных инструкции или шаблонов. В более простых словах можно сказать, что машинное обучение – это наука о том, как заставить компьютер учиться на основе их опыта без фактического традиционного программирование, то есть без какого-либо вмешательства человеческой помощи. Данная наука стремится ответить на вопрос “как люди могут построить компьютерные системы, которые автоматически улучшаются с опытом. И каковы фундаментальные законы, которые управляют всеми процессами обучения?”. Так же можно отметить, что цель данной науки ‒ стремится к тому, чтобы сделать машину “умной”. Так как на Земле самые умные существа – это люди, то можно смело сказать, что на данном этапе цель науки машинного обучения — это сделать машину с таким интеллектом, который будет максимально приближен к человеческому.

Классификация – это задача машинного обучения, которая заключается в отнесении объектов к заранее определенным категориям или классам на основе их характеристик. Классификация может быть бинарной, когда объекты относятся только к двум классам, или много классовой, когда объекты относятся к нескольким классам.

Метод ближайших соседей — это метод машинного обучения, используемый для классификации и регрессии. Он основывается на принципе, что объекты, находящиеся ближе друг к другу в пространстве признаков, скорее всего, принадлежат к одному и тому же классу или имеют похожие значения целевой переменной.

Алгоритм метода ближайших соседей достаточно прост: для каждого нового объекта вычисляется расстояние до всех объектов в обучающей выборке и выбирается ближайший объект. Затем классификация или регрессия нового объекта происходит на основе класса или значения целевой переменной выбранного ближайшего объекта.

Метод ближайших соседей является одним из самых простых и интуитивно понятных методов машинного обучения, но имеет ряд недостатков. Например, он не учитывает взаимосвязи между признаками и может быть чувствителен к выбросам в данных. Также он может быть неэффективным для больших объемов данных, так как может потребоваться много времени на вычисление расстояний между объектами.

1. Теоретические аспекты реализации алгоритмов классификации.

1.1. Основные определения реализации алгоритмов классификации.

Дадим более точное определение машинному обучению, расскажем историю создания, где оно применяется в современном мире и какие разделы включает в себя.

Машинное обучение – это наука о разработке алгоритмов и статистических моделей, которые информационные системы используют для выполнения задач без инструкций, полагаясь на шаблоны и логические выводы. Информационные системы используют алгоритмы машинного обучения для обработки больших объемов статистических данных и выявления шаблонов данных. Таким образом, системы могут более точно прогнозировать результаты на основе заданного набора входных данных. Например, специалисты по работе с данными могут обучить медицинское приложение диагностировать рак по рентгеновским изображениям, сохраняя огромное количество отсканированных изображений и сопутствующие диагнозы.

В машинном обучении выделяют 2 основных подхода — обучение с учителем и обучение без учителя.[11]

История создания машинного обучения началась еще в 1940-х годах, когда появилась идея создать машины, способные самостоятельно обучаться и принимать решения. В те годы ученые работали над различными методами и алгоритмами, которые могли бы помочь роботам и компьютерам обучаться на основе данных.

В 1950-х годах американский математик Артур Сэмюэль создал программу для игры в шашки, которая могла учиться на своих ошибках и становиться все более сильной. Это был один из первых примеров машинного обучения.

В 1960-х годах американский ученый Артур Л. Сэмуэль опубликовал статью, в которой впервые использовал термин "машинное обучение". С тех пор машинное обучение стало активно развиваться, появились новые методы и алгоритмы, которые позволили создать более эффективные и умные системы.

Сегодня машинное обучение используется в различных областях, таких как медицина, финансы, транспорт, робототехника и многие другие. Оно позволяет компьютерам и роботам обучаться на основе больших объемов данных и принимать решения, которые раньше были доступны только человеку.

Задачи, решаемые с помощью машинного обучения, можно разделить на следующие категории:

1. Задачи регрессии
2. Задачи классификации
3. Задачи кластеризации
4. Задачи уменьшения размерности
5. Задачи выявления аномалий

, но данной работе нам понадобится лишь классификация. Рассмотрим, что такое классификация, где она берет свое начало и на какие разделы разделяется [1].

Классификация определяется как процесс распознавания, понимания и группировки объектов и идей по заранее заданным категориям, также известным как “подгруппы населения”. С помощью этих предварительно классифицированных наборов обучающих данных классификация в программах машинного обучения использует широкий спектр алгоритмов для классификации будущих наборов данных по соответствующим категориям [2].

Алгоритмы классификации, используемые в машинном обучении, используют входные обучающие данные с целью прогнозирования вероятности того, что последующие данные попадут в одну из заранее определенных категорий. Одним из наиболее распространенных применений классификации является фильтрация электронных писем на “спам” или “не-спам”, как это используется ведущими современными поставщиками услуг электронной почты.

Проще говоря, классификация – это форма распознания образов.

Классификация в машинном обучении начала развиваться в 1950-х годах, когда появились первые алгоритмы машинного обучения. Одним из первых и самых известных алгоритмов классификации был алгоритм k-ближайших соседей (k-NN), который был разработан еще в 1951 году [3].

С течением времени появились и другие алгоритмы классификации, такие как логистическая регрессия, деревья решений, случайные леса и множество других. В 1960-х годах была разработана логистическая регрессия, которая до сих пор является одним из наиболее популярных алгоритмов классификации.

В 1970-х годах появилась концепция нейронных сетей, которая в конечном итоге привела к разработке многослойных нейронных сетей, которые стали очень популярными в 1990-х годах.

С появлением больших объемов данных и развитием вычислительных мощностей, в 2000-х годах стали появляться новые алгоритмы классификации, такие как глубокие нейронные сети, которые позволяют обрабатывать и анализировать огромные объемы данных с высокой точностью.

Сегодня классификация является одним из самых важных и распространенных подходов в машинном обучении, который используется в различных областях, таких как компьютерное зрение, обработка естественного языка, биоинформатика, маркетинг и многих других.

На данный момент существует 6 типов классификаторов:

1. Метод k-ближайших соседей(k-NN)
2. Метод опорных векторов (Support Vector Machines)
3. Классификатор дерева решений (Decision Tree Classifier)/ Случайный лес (Random Forests)
4. Наивный байесовский метод (Navies Bayes)
5. Линейный дискриминантный анализ (Linear Discriminant Analysis)
6. Логическая регрессия (Logistic Regression)

, но мы будем рассматривать метод ближайших соседей (k-NN) [5][12].

Метод ближайших соседей (k-nearest neighbors, KNN) – это алгоритм машинного обучения, который используется для решения задач классификации и регрессии. Он основан на идее, что близкие объекты имеют похожие характеристики, и поэтому могут быть отнесены к одному классу или иметь похожее значение целевой переменной.

Машинное обучение активно применяется во многих сферах жизни людей. Одним из разделов машинного обучения является классификация. Она призвана решать задачи отнесения некоторого объекта к одному из заданных классов. Пусть задано конечное множество объектов, для которых определена принадлежность к тому или иному классу. Назовем это множество обучающей выборкой. Для остальных объектов неизвестно, к какому классу они относятся. Необходимо построить алгоритм, способный классифицировать произвольный объект, не относящийся к исходному множеству [13].

Метод k-ближайших соседей (KNN) – это один из самых простых алгоритмов машинного обучения, который был разработан в 1950-х годах.

История создания метода ближайшего соседа начинается с двух людей Эвелин Фикс и Джозеф Лоусон Ходжес – младший

Эвелин Фикс (1904-1965) была математиком-статистиком, защитила докторскую диссертацию в Беркли и продолжала преподавать там статистику.

Джозеф Лоусон Ходжес – младший (1922-2000) также был статистиком в Беркли и участвовал в статистическом сотрудничестве с Двадцатыми военно-воздушными силами ВВС США, начиная с 1944 года.

Эти два блестящих ума встретились в 1951 году в отчете по техническому анализу, подготовленном для ВВС США, где они представили непараметрический метод классификации (дискриминантный анализ). Они никогда не публиковали статью официально, вероятно, из-за характера работы и конфиденциальности, особенно учитывая глобальную атмосферу в те дни, вскоре после Второй мировой войны.

Это первое известное внедрение непараметрической классификации, которая позже стала знаменитым алгоритмом k-NN.

Затем Томас Кавер и Питер Харт в 1967 году доказали верхнюю границу частоты ошибок при много классовых классификациях k-NN (в два раза превышающую частоту ошибок Байеса).

Открытие Cover & Hart проложило путь для ряда новых исследований отторжения, таких как:

в 1970 году Эдвард Хеллман исследовал “правило (k,k?) ближайшего соседа с возможностью отклонения“;

в 1975 году: Фукунага и Хостетлер внесли уточнения в отношении частоты байесовских ошибок;

в 1976 году: Дудани. В 1978 году Бейли и Джейн опубликовали подходы, взвешенные по расстоянию;

в 1983 году Адам Йозвик представил: Схему обучения для нечеткого правила k-NN;

в 1985 году Джеймс Келлер и др. разработали FKNN (нечеткий k-NN): Нечеткий алгоритм k-ближайшего соседа;

в 2000 году Бермехо и Кабестан опубликовали: Адаптивные мягкие k-классификаторы ближайших соседей;

в тех пор KNNS претерпела множество улучшений, и новые подходы продолжают появляться по сей день [4].

1.2. Описание алгоритма и приведение примера решения модельной задачи с описанием ее.

Описание библиотек Scikit-learn, Pandas, Seaborn, NumPy, Matplotlib

Scikit-learn – один из наиболее широко используемых пакетов Python для Data Science и Machine Learning. Он позволяет выполнять множество операций и предоставляет множество алгоритмов. Scikit-learn также предлагает отличную документацию о своих классах, методах и функциях, а также описание используемых алгоритмов [6].

В Scikit-learn входит:

1. Предварительная обработка данных;
2. Уменьшение размерности;
3. Выбор модели;
4. Регрессии;
5. Классификации;
6. Кластерный анализ;

Pandas – главная библиотека в Python для работы с данными. Её активно используют аналитики данных и дата-сайентисты. Библиотека была создана в 2008 году компанией AQR Capital, а в 2009 году она стала проектом с открытым исходным кодом с поддержкой большого комьюнити [7].

Seaborn – это библиотека для создания статистических графиков на Python. Она основывается на Matplotlib и тесно взаимодействует со структурами данных Pandas .

Архитектура Seaborn позволяет вам быстро изучить и понять свои данные. Seaborn захватывает целые фреймы данных или массивы, в которых содержатся все ваши данные, и выполняет все внутренние функции, нужные для семантического маппинга и статистической агрегации для преобразования данных в информативные графики [8].

Matplotlib – популярная Python-библиотека для визуализации данных. Она используется для создания любых видов графиков: линейных, круговых диаграмм, построчных гистограмм и других — в зависимости от задач [9].

NumPy – это открытая бесплатная Python-библиотека для работы с многомерными массивами, это аналог библиотеки Matplotlib. NumPy чаще всего используют в анализе данных и обучении нейронных сетей – в каждой из этих областей нужно проводить много вычислений с такими матрицами [10].

Задача

Напишите программу используя классификацию. Проведите анализ атрибутов, полученной модели. Для классификации используйте метод ближайших соседей и набор данных ирисов Шиффера.

Для начала импортируем библиотеки, которые будем использовать в программе (Рисунок 1).

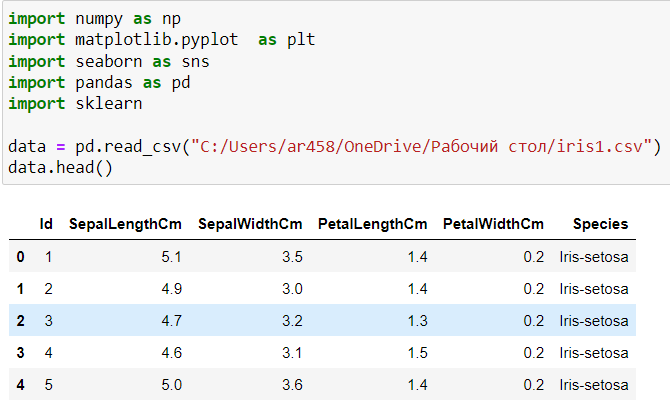


Рисунок 1 – Импортирование библиотек база данных

Далее получаем сводную информацию по датасету ирисов Шиффера (Рисунок 2).

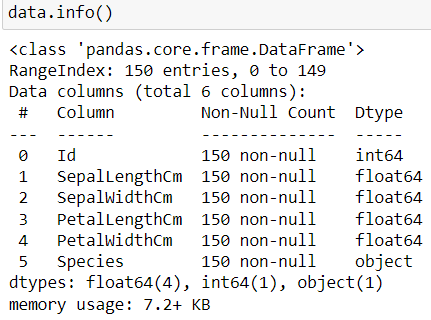


Рисунок 2 – Получение сводной информации

Изменяем название столбцов в таблице для удобства, а также выполняем разбивку данных (Рисунок 3).

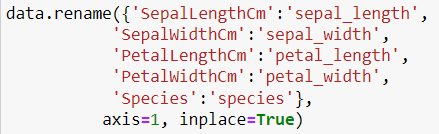




Рисунок 3 – Изменение столбцов и разбивка данных

Диаграмма рассеяния для данных, содержащих информацию о длине и ширине лепестков для трех видов ирисов: setosa, versicolor и virginica (Рисунок 4).

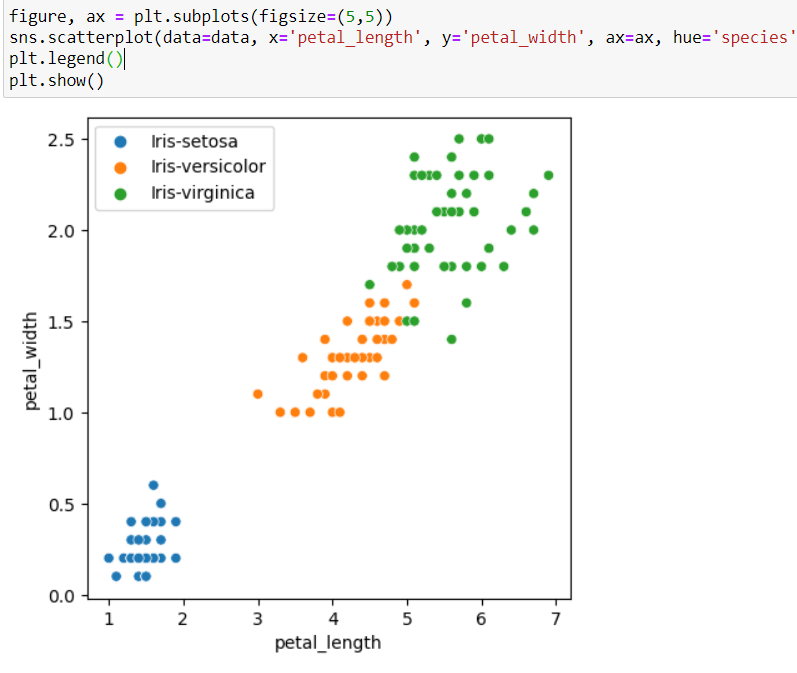


Рисунок 4 – Диаграмма рассеяния для длины и ширины лепестков

Выполним разбивку данных на обучающий и тестовый набор. Пусть тестовая выборка содержит 30% от общих данных (Рисунок 5).



Рисунок 5 – Разделение данных на тестовую и обучающую выборку

Далее при создании модели используем один параметр. n\_neighbors устанавливаем, как 3, что означает, что для классификации данной точки требуется 3 точки соседства (Рисунок 6).

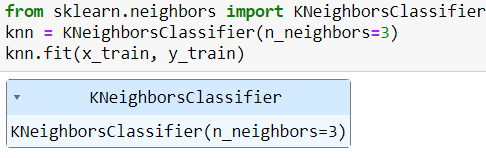


Рисунок 6 Количество ближайших соседей

Модель создана, теперь предскажем метки классов на тестовой выборке (Рисунок 7). Далее оценим точность тестовой выборки (Рисунок 8). Оценка точности классификации составила 0,97, что означает возможность применимости данного алгоритма.

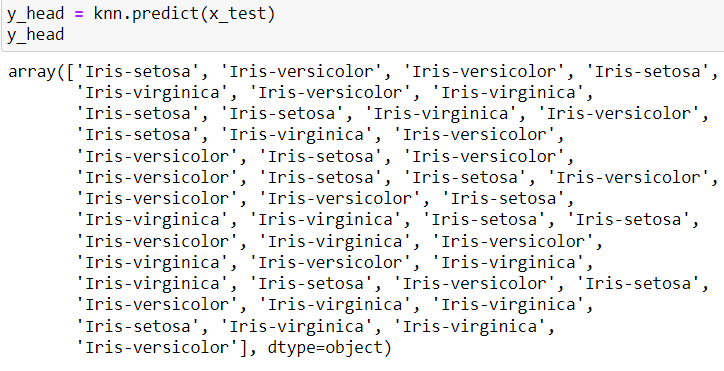


Рисунок 7 – Предсказание меток классов на тестовой выборке

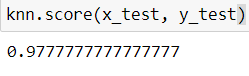


Рисунок 8 – Оценка точности

Теперь построим график зависимости точности модели на тестовой выборке от числа соседей k (Рисунок 9).

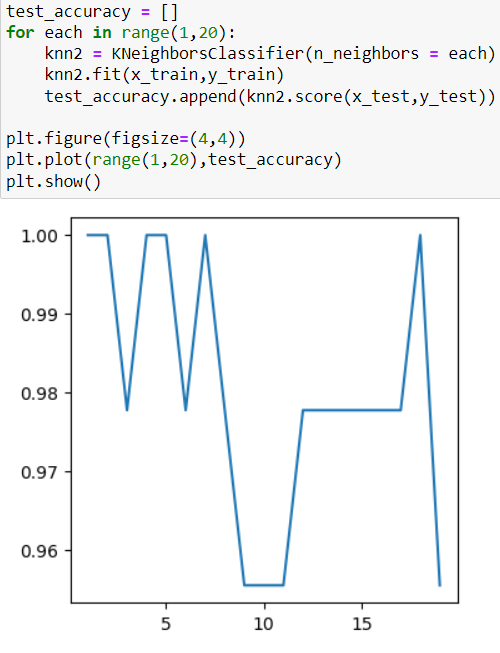


Рисунок 9 – График зависимости точности модели

2. Практическое применение реализации алгоритмов классификации данных.

2.1. Постановка задачи с помощью dataset (Набор данный о раке молочной железы Висконсин).

Задача

Написать программу с использованием классификации методом ближайших соседей на датасете «Набор данных о раке молочной железы Висконсин» и провести анализ графиков.

Поскольку данные имеют различную размерность необходимо их нормализовать. Для этого нам понадобится формула минмакс-нормализации. Выглядит она следующим образом:

,

где x – это x\_data, xmin – это np.min(x\_data) и xmax – это np.max(x\_data).

Также в работе было использована формула евклидова расстояния. Выгляди она следующим образом:

По алгоритму kNN результирующим тематическим классом классификатора является класс, имеющий максимальную оценку близости

,

при этом *Kmax* определяется по условию:

,

где *sim*(*D*, *Ki*) – оценка близости документа *D* к *Ki*.

2.2. Реализация задачи с помощью dataset (Набор данных о раке молочной железы Висконсин).

Для начала импортируем библиотеки, которые мы дем использовать в программе, а также базу данных (Рисунок 1).

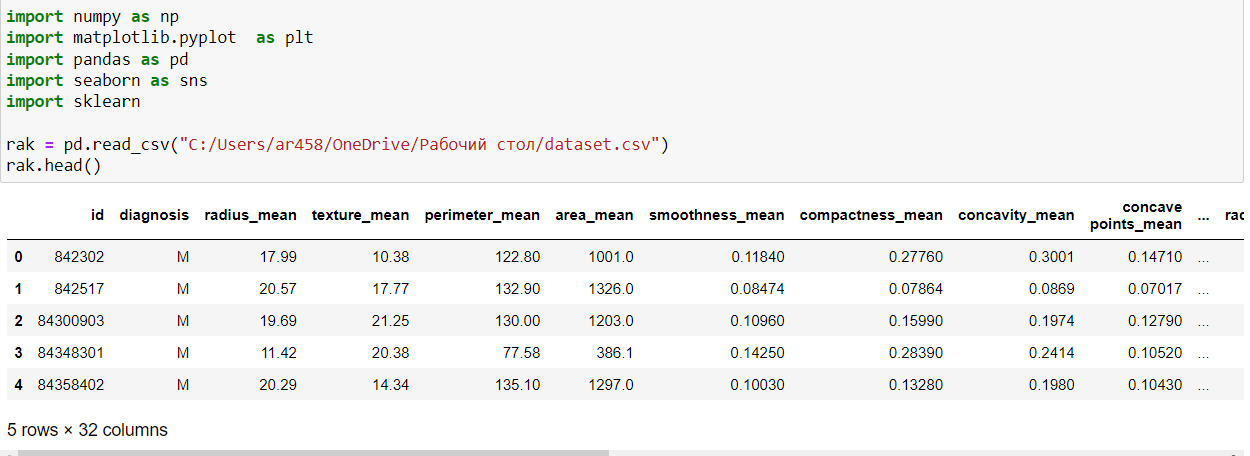


Рисунок 1 – Импортирование библиотек и датасета

Удалим не нужный для нас столбец “id” (Рисунок 2).



Рисунок 2 – Удаление столбца “id”

Далее выполним разбивку данных опухолей молочной железы на доброкачественные (B) и злокачественные (M) (Рисунок 3).



Рисунок 3 – Разбивка данных

Теперь построим диаграмму рассеяния для двух классов опухолей молочной железы (Рисунок 4.1).

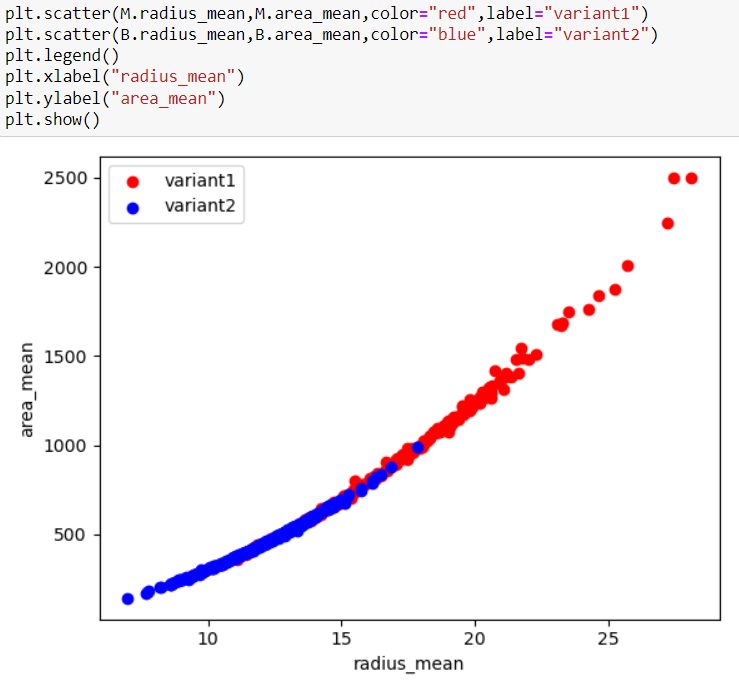


Рисунок 4.1 – Диаграмма рассеяния двух классов

Также строим диаграмму рассеяния для двух классов опухолей молочной железы по радиусу и текстуре (Рисунок 4.2).

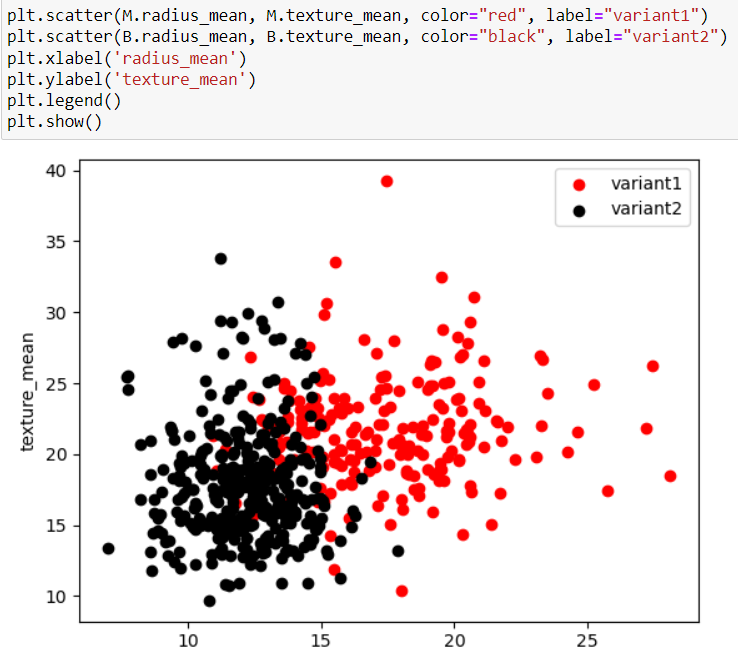


Рисунок 4.2 – Диаграмма рассеяния двух классов

Осуществляем предобработку данных для задачи бинарной классификации, где 1 соответствует диагнозу "M" (злокачественная опухоль), а 0 - диагнозу "B" (доброкачественная опухоль). Также используем формулу минмакс (Рисунок 5).

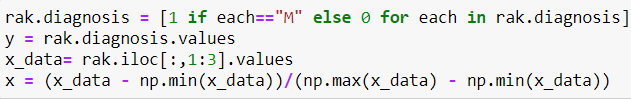


Рисунок 5 – Переработка данных

Выполним разбивку данных на обучающий и тестовый набор. Пусть тестовая выборка содержит 30% от общих данных (Рисунок 6).

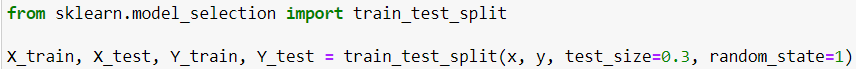


Рисунок 6 – Разбивка данных обучающую и тестовую выборку

Далее при создании модели используем один параметр. n\_neighbors устанавливаем, как 3, что означает, что для классификации данной точки требуется 3 точки соседства (Рисунок 7).

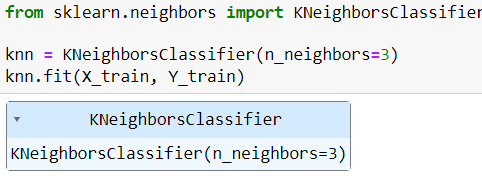


Рисунок 7 – Количество ближайших соседей

Модель создана, теперь предскажем метки классов на тестовой выборке (Рисунок 8). Далее оценим точность тестовой выборки (Рисунок 9). Оценка точности классификации составила 0,84, что означает возможность применимости данного алгоритма.

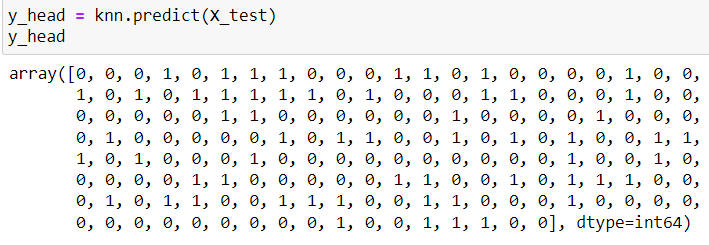


Рисунок 8 – Предсказание меток классов на тестовой выборке



Рисунок 9 – Оценка точности модели

Далее построим график зависимости точности модели на тестовой выборке от числа соседей k (Рисунок 10).

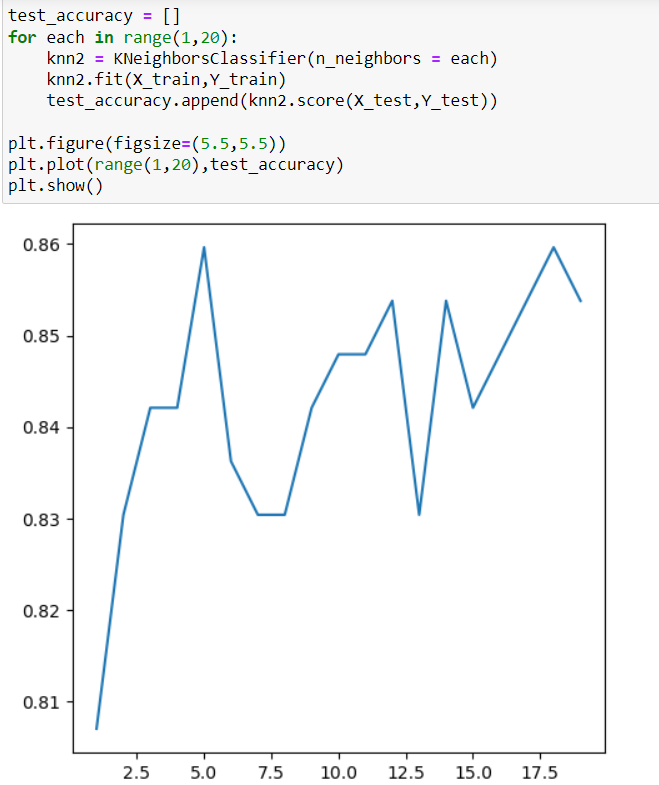


Рисунок 10 – График зависимости точности модели

Выполним перебор различных значений числа соседей k для модели k-ближайших соседей (k-NN) и оцениваем качество модели на тестовой выборке для каждого значения k. В нашем случае, оценка качества модели выполняется посредством вычисления средней ошибки (mean error) (Рисунок 11).

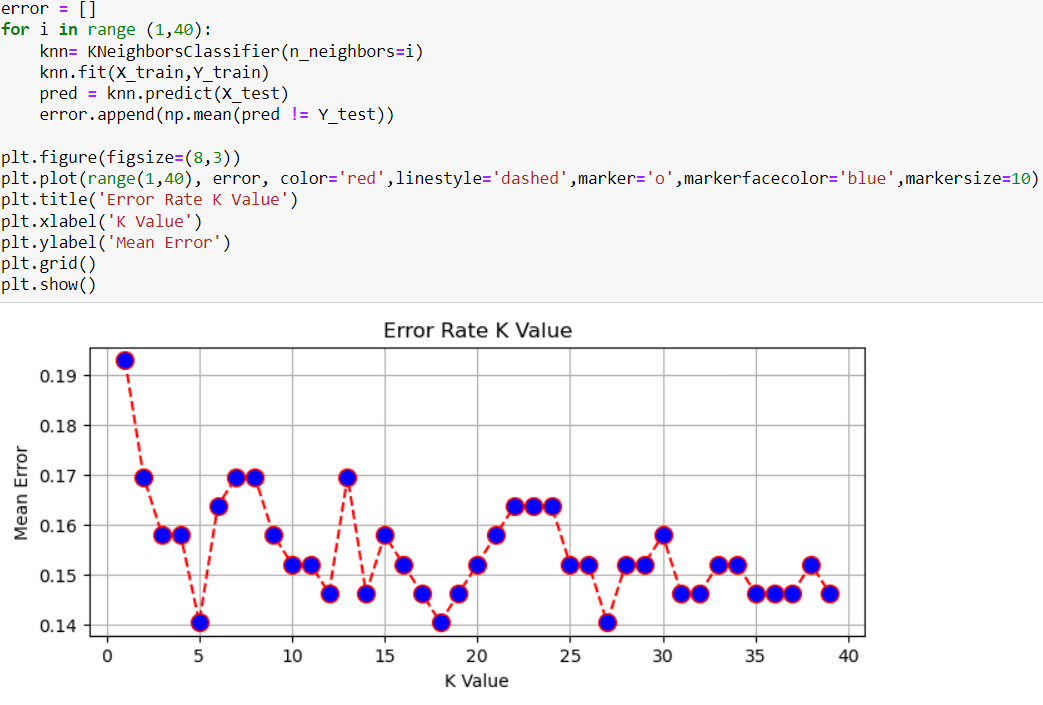


Рисунок 11 – Вычисления средней ошибки путем графика

Заключение

В данной курсовой работе мы рассмотрели тему «Реализация алгоритмов классификации на языке Python». При изучении данной темы мы познакомились с методом классификации, методом ближайших соседей, библиотеками NumPy, Pandas, Scikit-learn, Seaborn и Matplotlib, а также проанализировали dataset «Набор данных о раке молочной железы Висконсин» по которому мы написали программу.

Наиболее простой и популярный алгоритм классификации в машинном обучении – метод ближайших соседей (k-NN). Он основывается на том, что объекты с похожими признаками склонны к принадлежности к одному классу.

В Python существует множество библиотек, которые позволяют реализовать алгоритм k-NN. Одной из самых популярных библиотек для машинного обучения является scikit-learn. Библиотека scikit-learn предоставляет класс KNeighborsClassifier для реализации алгоритма k-NN. А также для реализации метода ближайших соседей используют библиотеку Matplotlib для графической отрисовки. Библиотека Matplotlib используется для создания любых видов графиков: линейных, круговых диаграмм, построчных гистограмм и других — в зависимости от задач [9].

Список использованных источников

1. Grigoriy\_Alekseev: Введение в машинное обучение. – URL: <https://habr.com/ru/articles/448892/> (17.04.2023)
2. Classification in Machine Learning: What it is & Classification Models – URL:<https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning> (17.04.2023)
3. How To Implement Classification In Machine Learning? – URL: <https://www.edureka.co/blog/classification-in-machine-learning/> (21.04.2023)
4. k Nearest Neighbor (kNN) History – URL: <https://holypython.com/knn/k-nearest-neighbor-knn-history/> (18.04.2023)
5. Александр Ланский, Обзор методов классификации в машинном обучении с помощью Scikit-Learn – URL: <https://tproger.ru/translations/scikit-learn-in-python/> (19.04.2023)
6. Что такое Scikit-learn – гайд по популярной библиотеке Python для начинающих – URL: <https://datastart.ru/blog/read/chto-takoe-scikit-learn-gayd-po-populyarnoy-biblioteke-python-dlya-nachinayuschih> (19.04.2023)
7. Работаем с Pandas: основные понятия и реальные данные – URL: <https://skillbox.ru/media/code/rabotaem-s-pandas-osnovnye-ponyatiya-i-realnye-dannye/> (20.04.2023)
8. Как наторить красивые графики на Python с Seaborn – URL: <https://habr.com/ru/companies/otus/articles/540526/> (20.04.2023)
9. Библиотека Matplotlib для построения графиков – URL: <https://skillbox.ru/media/code/biblioteka-matplotlib-dlya-postroeniya-grafikov/> (21.04.2023)
10. Библиотека Numpy: все что нужно знать новичку – URL: <https://skillbox.ru/media/code/biblioteka-numpy-vsye-chto-nuzhno-znat-novichku/> (23.04.2023)
11. Основы машинного обучения: учебное пособие / О.В. Лимановская, Т.И. Алферьева; Мин-во науки и высш. образования РФ. — Екатеринбург: Изд-во Урал. ун-та, 2020. URL: <https://elar.urfu.ru/bitstream/10995/88687/1/978-5-7996-3015-7_2020.pdf> (07.05.2023)
12. Бабаев А.М., Шемякина М.А. ОБЗОР КЛАССИЧЕСКИХ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В КОНТЕКСТЕ РЕШЕНИЯ ЗАДАЧ КЛАССИФИКАЦИИ // Форум молодых ученых. 2018. №11-1 (27). URL: <https://cyberleninka.ru/article/n/obzor-klassicheskih-metodov-mashinnogo-obucheniya-v-kontekste-resheniya-zadach-klassifikatsii> (08.05.2023).
13. Трясучкин Владислав Алексеевич, Синцева Марина МихайЛовна ИССЛЕДОВАНИЕ ОПТИМИЗАЦИИ ГИПЕРПАРАМЕТРОВ АЛГОРИТМА k‐БЛИЖАЙШИХ СОСЕДЕЙ // Вестник ПензГУ. 2019. №2 (26). URL: <https://cyberleninka.ru/article/n/issledovanie-optimizatsii-giperparametrov-algoritma-k-blizhai-shih-sosedei> (08.05.2023).

Приложение 1.   
Листинг программы

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import seaborn as sns

import sklearn

rak = pd.read\_csv("C:/Users/ar458/OneDrive/Рабочий стол/dataset.csv")

rak.head()

rak.drop(['id'], axis=1, inplace=True)

M = rak[rak.diagnosis=="M"]

B = rak[rak.diagnosis=="B"]

plt.scatter(M.radius\_mean,M.area\_mean,color="red",label="variant1")

plt.scatter(B.radius\_mean,B.area\_mean,color="blue",label="variant2")

plt.legend()

plt.xlabel("radius\_mean")

plt.ylabel("area\_mean")

plt.show()

plt.scatter(M.radius\_mean, M.texture\_mean, color="red", label="variant1")

plt.scatter(B.radius\_mean, B.texture\_mean, color="black", label="variant2")

plt.xlabel('radius\_mean')

plt.ylabel('texture\_mean')

plt.legend()

plt.show()

rak.diagnosis = [1 if each=="M" else 0 for each in rak.diagnosis]

y = rak.diagnosis.values

x\_data= rak.iloc[:,1:3].values

x = (x\_data - np.min(x\_data))/(np.max(x\_data) - np.min(x\_data))

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=1)

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n\_neighbors=3)

knn.fit(X\_train, Y\_train)

y\_head = knn.predict(X\_test)

y\_head

knn.score(X\_test, Y\_test)

test\_accuracy = []

for each in range(1,20):

knn2 = KNeighborsClassifier(n\_neighbors = each)

knn2.fit(X\_train,Y\_train)

test\_accuracy.append(knn2.score(X\_test,Y\_test))

plt.figure(figsize=(4,4))

plt.plot(range(1,20),test\_accuracy)

plt.show()

error = []

for i in range (1,40):

knn= KNeighborsClassifier(n\_neighbors=i)

knn.fit(X\_train,Y\_train)

pred = knn.predict(X\_test)

error.append(np.mean(pred != Y\_test))

plt.figure(figsize=(10,5))

plt.plot(range(1,40), error, color='red',linestyle='dashed',marker='o',markerfacecolor='blue',markersize=10)

plt.title('Error Rate K Value')

plt.xlabel('K Value')

plt.ylabel('Mean Error')

plt.grid()

plt.show()