|  |  |
| --- | --- |
| **Изображение выглядит как текст  Автоматически созданное описание** | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | |
| Институт информационных технологий | |
| Кафедра прикладной математики  **КУРСОВАЯ РАБОТА**  по дисциплине  Языки программирования для статистической обработки данных.  **Тема курсовой работы**: **Реализация перцептрона. Перцептрон в задачах бинарной классификации.**   |  |  | | --- | --- | | Студент группы ИМБО-01-19 | Ломовцев Павел Павлович | |  |  | |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) | |  |  | | Руководитель курсовой работы | Митина Ольга Алексеевна | |  |  | |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) | |  |  | | Работа представлена к защите | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. | |  |  | | Допущен к защите | « « \_\_\_\_\_\_\_\_\_\_\_\_2021 г. |   Москва 2021   |  |  | | --- | --- | | **Изображение выглядит как текст  Автоматически созданное описание** | | | МИНОБРНАУКИ РОССИИ | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | Институт информационных технологий | |   Кафедра прикладной математики  **ЗАДАНИЕ**  **на выполнение курсовой работы по дисциплине**  Языки программирования для статистической обработки данных   |  |  | | --- | --- | | Студент Ломовцев Павел Павлович | Группа ИМБО-01-19 |   **Тема работы:** Перцептрон в задачах бинарной классификации  **Перечень вопросов, подлежащих разработке, и обязательного графического материала:**   1. Код реализации перцептрона на языке программирования R. 2. Тестирование перцептрона. 3. Расчет метрик качества классификации.  |  |  |  |  | | --- | --- | --- | --- | | **Срок представления к защите курсовой работы:** | | до «\_\_\_» \_\_\_\_\_\_\_\_\_\_2021 г. | | |  | |  | | | **Задание на курсовую работу выдал** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | (Митина О.А.) | |  | | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2021 г. | | | **Задание на курсовую работу получил** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | (Ломовцев П. П.) | | |

ОГЛАВЛЕНИЕ

[Введение 3](#_Toc69827085)

[Теоретическая часть 4](#_Toc69827086)

[Понятие бинарной классификации и перцептрона 4](#_Toc69827087)

[Метрики качества классификации 6](#_Toc69827088)

[Практическая часть 11](#_Toc69827089)

[Реализация перцептрона на R 11](#_Toc69827090)

[Тестирование перцептрона 13](#_Toc69827091)

[Заключение 18](#_Toc69827092)

[Список используемой литературы 20](#_Toc69827093)

[Приложение 21](#_Toc69827094)

# Введение

С каждым годом объем производимой продукции растет, поэтому компаниям необходимо автоматизировать процесс сортировки товара. Одним из методов решения данной задачи является структура, называемая перцептроном. Это линейный классификатор, в котором заложена модель восприятия информации человеческим мозгом.

Перцептрон – один из наиболее популярных методов решения задачи классификации. Автоматизация процесса разделения товаров по их свойствам облегчит работу сотрудникам сортировочного центра и исключит возможность ошибки вследствие человеческого фактора.

Поэтому данная тема курсовой работы является актуальной.

Цель курсовой работы – реализовать перцептрон с помощью средств языка программирования высокого уровня R.

Задачи, решаемые в данной курсовой работе:

* изучение научной и методической литературы по проблеме бинарной классификации данных;
* предобработка данных;
* использование знаний математической статистики и линейной алгебры с использованием современных средств обработки данных: языка программирования высокого уровня R;
* реализация перцептрона;
* решение задачи бинарной классификации продуктов питания (цитрусов) с применением перцептрона;
* обучение качественному оформлению документации.

# Теоретическая часть

## Понятие бинарной классификации и перцептрона

Часто в жизни встречаются ситуации, когда необходимо разбить объекты на две группы. У каждого объекта есть свойства, которые его характеризуют. В процессе разбиения объектов на группы наш мозг старается выбрать похожие и занести их в одну группу. Те объекты, свойства которых значимо отличаются от свойств объектов первой группы, относятся к другой группе. В этом и заключается задача бинарной классификации.

**Бинарная классификация** – разбиение исходного множества объекта на **два** различных подмножества таким образом, чтобы характеристики объектов отличались друг от друга, но внутри каждого из подмножеств свойства были схожи [1].

Задача бинарной классификации – одна из задач **обучения с учителем** (англ. Supervised learning). Это означает, что для того, чтобы алгоритм работал правильно, ему нужны объективные данные. На каждом шаге результат работы алгоритма будет сравниваться с уже существующим единственно верным результатом и если он окажется неверным, необходимо корректировать параметры классификатора до тех пор, пока не получится верный результат.

Таким образом, задача бинарной классификация требует внимательного подхода к выбору алгоритма, а также к его обучению (корректировки параметров).

В этой работе мы будем использовать один из алгоритмов бинарной классификации – перцептрон, который моделирует работу человеческого мозга.

**Перцептрон** – математическая модель восприятия информации головным мозгом, предложенная Фрэнком Розенблаттом в 1957 году. Перцептрон стал одной из первых нейросетей в мире [2].

Перцептрон состоит из трёх типов элементов, а именно: поступающие от [датчиков](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D1%82%D1%87%D0%B8%D0%BA) [сигналы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB) передаются [ассоциативным](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C) элементам, а затем реагирующим элементам [2].

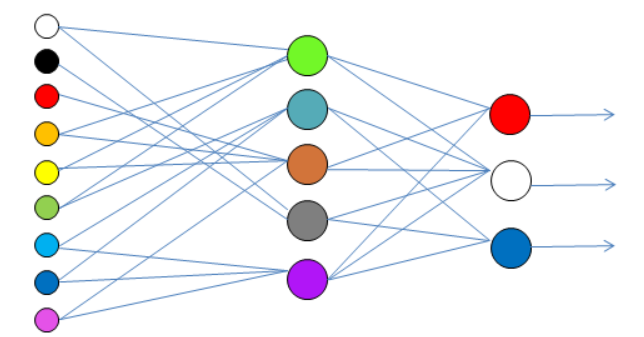


Рисунок 1 Логическое представление перцептрона с тремя уровнями

На рисунке 1 показана логическая схема перцептрона с тремя выходами, которая показывает, каким образом входные данные преобразуются в выходные, проходя обучение через один **А-слой**. Рассмотрим, в каких областях полезен перцептрон.

Перцептрон широко применяется в различных сферах: в медицине – в задачах классификации пациентов на больных и здоровых, в информационной безопасности – классификация пользователей на взломщиков и тех, кто не представляет опасности, в сфере сбыта – классификация товаров по качеству, в криминальном деле – детектор лжи определит, лжет человек или нет. В задачах распознавания изображений перцептрон является одной из наиболее популярных структур для быстрого и в то же время точного результата классификации.

Перцептрон решает, к какой группе отнести объект, основываясь на его атрибутах. Для этого алгоритм необходимо обучить, поскольку задача классификации относится к задаче обучения с учителем.

На вход подаются данные (атрибуты объектов), затем числовые значения атрибутов скалярно умножаются на веса, и их произведение подается на функцию активации. Это такая функция, которая позволяет отнести объект к определенному классу исходя из знака скалярного произведения признаков на веса и смещение (bias). Если знак произведения отличен от реального, то веса необходимо скорректировать. Процедура повторяется до тех пор, пока не получится верный результат или лимит на итерации не будет превышен. В таком случае все характеристики классификатора перестают изменяться и модель формируется окончательно. Ее можно тестировать на других данных для того, чтобы определить, насколько перцептрон хорошо различает объекты по группам. Для этого необходимо рассмотреть метрики качества бинарной классификации.

Таким образом, перцептроны позволяют создать набор [«ассоциаций»](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BF%D1%81%D0%B8%D1%85%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F)) между входными [стимулами](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D0%BC%D1%83%D0%BB) и необходимой [реакцией](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B0%D0%BA%D1%86%D0%B8%D1%8F_(%D0%B1%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%8F)) на выходе. В биологическом плане это соответствует преобразованию, например, зрительной информации в [физиологический ответ](https://ru.wikipedia.org/w/index.php?title=%D0%A4%D0%B8%D0%B7%D0%B8%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%80%D0%B5%D0%B0%D0%BA%D1%86%D0%B8%D1%8F&action=edit&redlink=1) от двигательных [нейронов](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D1%8B).

## Метрики качества классификации

Для оценки качества классификации необходимо рассчитать все необходимые метрики, которые позволят определить, насколько хорошо модель предсказывает класс очередного объекта.

**Метрикой** в задачах машинного обучения называется количественный показатель (в долях), который показывает ту или иную характеристику классификатора [4].

**Метод оценки качества классификации** – инструментарий для количественной оценки результатов классификации [3].

Принято выделять две группы методов оценки качества классификации:

* **внешние** (англ. *External*) меры основаны на сравнении результата классификации с априори известным разделением на классы;
* **внутренние** (англ. *Internal*) меры отображают качество кластеризации только по информации в данных [5].

Мы будем использовать **внешнюю** меру оценки качества классификации, т. е. по данным с априори известным реальным распределением данных.

Определим понятие **аккуратности** в задачах классификации. Введем следующие обозначения:

* (True Positive) – количество верно определенных классификатором объектов, относящихся к группе, обозначенной в объективных данных флагом 1;
* (False Positive) – количество неверно определенных классификатором объектов, относящихся к группе, обозначенной в объективных данных флагом «1»;
* (True Negative)– количество верно определенных классификатором объектов, относящихся к группе, обозначенной в объективных данных флагом «0»;
* (False Negative) – количество неверно определенных классификатором объектов, относящихся к группе, обозначенной в объективных данных флагом 0.

С помощью данных обозначений можно рассчитать все необходимые метрики для оценки качества классификации.

Рассмотрим основные **метрики** для оценки качества классификации:

**Аккуратность** (англ. **Accuracy) –** отношение количества верно определенных объектов к общему числу объектов [6].

|  |  |
| --- | --- |
|  | (1) |

Метрика показывает, работает ли наша модель или нет.

**Точность** (англ. **Precision**) – доля объектов, названных классификатором положительными и при этом действительно являющимися положительными [6].

|  |  |
| --- | --- |
|  | (2) |

Результат показывает, способна ли наша модель отличать классы друг от друга [6].

**Полнота** (англ. **Recall**) – доля объектов, действительно положительного класса, которую нашел алгоритм из всех объектов положительного класса, полученных алгоритмом [6].

|  |  |
| --- | --- |
|  | (3) |

Полнота показывает способность нашей модели обнаруживать класс в общем.

**F-мера –** среднее гармоническое точности и полноты. Эта метрика показывает в среднем, насколько полно и точно наша модель определяет отношение объекта к конкретному классу [6].

|  |  |
| --- | --- |
|  | (4) |

В настоящее время специалисты по машинному обучению используют данные метрики для оценки качества классификации. Принято считать, что модель должна быть сбалансированной, то есть метрики accuracy, precision, recall, F-мера должны быть не менее 0.7 на тестовом наборе данных – наборе, который модель видит впервые, иначе модель будет плохо классифицировать новые объекты.

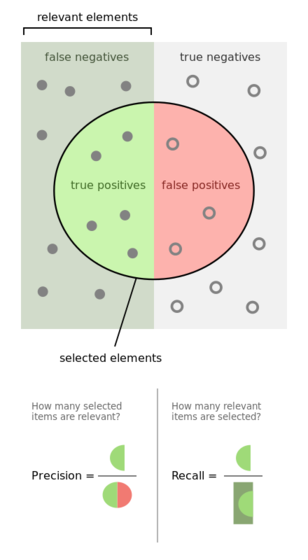


Рисунок 2 Графическое представление точности и полноты

Рисунок 2 поясняет, как влияют значения метрик на качество модели [6].

Низкий показатель обеих метрик (точности и полноты) указывает на то, что классификатор работает плохо – его необходимо переобучить, изменив входные параметры. Если после изменения модель все так же не работает, необходимо проверить реализацию модели. Стоит также обратить внимание на объем тренировочной выборки. Если он мал, то модель недообучается, что тоже может привести к низким рейтингам метрик.

Таким образом, задача бинарной классификации (разбиение объектов на две группы) – одна из наиболее актуальных проблем в различных сферах. Сегодня известно множество алгоритмов классификации. В работе мы используем разбиение на группы с помощью перцептрона – простейшей нейронной сети. Чтобы оценить качество кластеризации, необходимо сравнить результат, полученный в результате классификации с тем, что описывает «целевой столбец» и рассчитать соответствующие метрики – аккуратность, точность, полноту и F-меру. Именно эти показатели позволят нам оценить то, как хорошо модель предсказывает класс очередного объекта по его характеристикам.

# Практическая часть

## Реализация перцептрона на R

Для решения задача бинарной классификации необходимо определить алгоритм реализации перцептрона. Для создания модели необходимо получить данные о количестве итераций обучения, количестве признаков объекта и шагу обучения (learning rate). После инициализации перцептрона его необходимо обучить. Приведем алгоритм обучения перцептрона:

На вход подается матрица признаков объекта и целевой столбец, в котором отражено достоверное распределение предсказываемой переменной. Далее производятся шаги инициализации атрибутов класса «Перцептрон»:

1. Инициализируем вектор весов нулями.
2. Инициализируем bias параметр b = 0.
3. В начальный момент времени номер итерации t = 0.
4. Задаем шаг обучения learning rate = 0.1
5. Пока t меньше лимита итераций, выполнять:
   1. Случайно выбираем объект (x[i], y[i]) из тренировочной выборки, где x[i] – вектор признаков объекта, y[i] – объективный класс объекта (1 или -1).
   2. Если знаки результата предсказания перцептрона и y[i] различны, то это означает, что перцептрон дал неверный ответ, а значит необходимо изменить значения весов:

.

* 1. Обновляем счетчик итераций t = t + 1.

Таким образом, финальное значение ветора весов позволяют классифицировать новый объект . Если , то мы относим объект к классу +1, в противном случае мы относим объект к классу -1.

Для реализации перцептрона будем использовать язык программирования высокого уровня R.

**R** – язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом в рамках проекта [GNU](https://ru.wikipedia.org/wiki/GNU) [6].

В настоящее время язык R используется в крупных компаниях для анализа, визуализации данных, а также для сложных математических операций. Преимуществом этого языка программирования является то, что все операции векторизованы, что исключает явное использование циклов в программном коде.[7]

Перед началом реализации перцептрона необходимо импортировать необходимые пакеты.

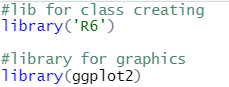


Рисунок 3 Импорт пакетов в среду R

На рисунке 3 изображены пакеты, которые будут использоваться в ходе реализации перцептрона. Пакет «R6» используется для создания класса, а «ggplot2» для визуализации результатов работы алгоритма.

Класс будет иметь 2 метода – fit (обучение) и predict (предсказание).

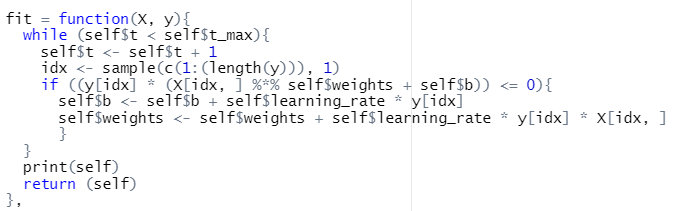


Рисунок 4 Реализация алгоритма обучения перцептрона

На рисунке 4 показана реализация описанного выше алгоритма обучения перцептрона на языке программирования высокого уровня R.

После того, как перцептрон обучится, его необходимо протестировать, ведь основное предназначение перцептрона – предсказывать класс очередного объекта по его характеристикам. Реализуем метод предсказания с учетом условия отнесения объекта к классу: если скалярное произведение признаков на веса больше нуля, то объект относится к классу +1, в противном случае -1.

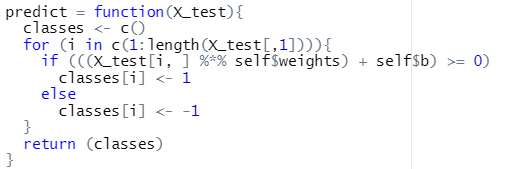


Рисунок 5 Реализация метода предсказания перцептрона

На рисунке 5 показана реализация алгоритма предсказания класса объекта по его признакам.

Таким образом был реализован класс «Перцептрон», который можно использовать для решения задачи классификации данных.

## Тестирование перцептрона

После реализации алгоритма необходимо протестировать его на данных, в которых присутствует бинарный признак. В данных, используемых для этой работы, такой столбец описывает наименование цитруса:

* «1» – апельсин;
* «-1» – грейпфрут.

Опишем данные, на которых будет производиться тестирование.

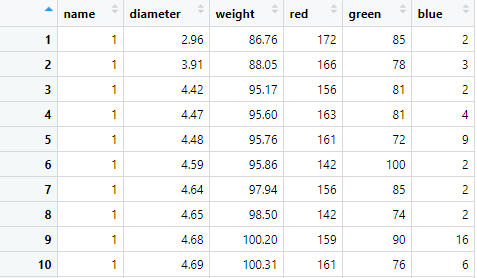


Рисунок 6 Описание тренировочных данных

На рисунке 6 показаны данные, которые будут использованы для тестирования перцептрона.

* name – целевой столбец факторного типа данных;
* diameter – диаметр фрукта, числового типа данных;
* weight – вес фрукта, числового типа данных;
* red – оттенок красного у фрукта, числового типа данных;
* green – оттенок зеленого у фрукта, числового типа данных;
* blue – оттенок синего у фрукта, числового типа данных. [8]

Импортируем данные в рабочую среду R. После этого необходимо разделить исходные данные на тренировочную и тестовую выборку. Тренировочная – для обучения модели, тестовая – для тестирования.

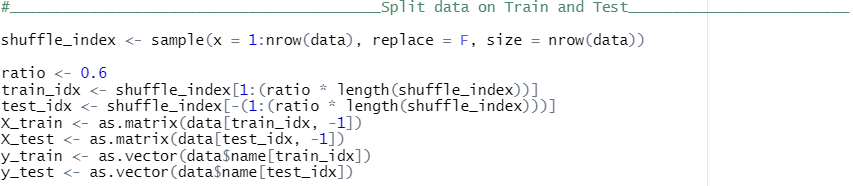


Рисунок 7 Разделение выборки на тренировочную и тестовую

На рисунке 7 показана реализация разбиения данных на тренировочную и тестовую выборки.

После импорта и разбиения данных необходимо создать экземпляр класса «Перцептрон» и приступить к его обучению.

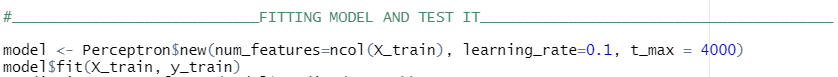


Рисунок 8 Создание экземпляра класса и обучение

На рисунке 8 отображен процесс создания перцептрона с шагом обучения 0.1, количеством итераций 4000 и количеством признаков, равным 5.

После создания объекта обучим перцептрон на тренировочной выборке. Подадим в качестве входных данных X\_train (матрица признаков) и y\_train (объективные значения целевого столбца). Предскажем классы объектов из тестовой.



Рисунок 9 Предсказание обученного перцептрона

На рисунке 9 показан программный код предсказания перцептроном классов. После предсказания классов необходимо рассчитать метрики качества классификации.

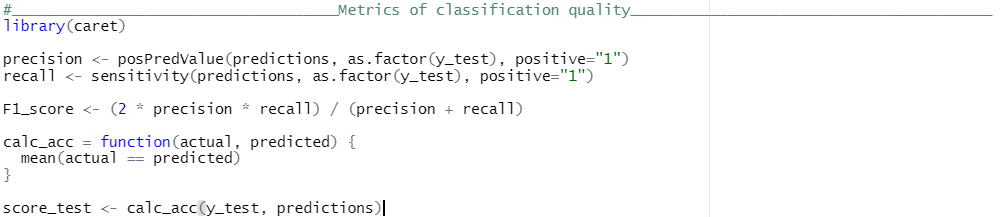


Рисунок 10 Расчет метрик качества классификации

На рисунке 10 показан расчет метрик качества классификации с помощью пакета «caret». Проверим результаты тестирования.

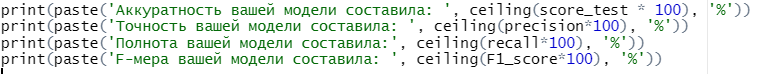


Рисунок 11 Код для вывода метрик

На рисунке 11 показан программный код вывода на экран рассчитанных метрик для перцептрона. Для наилучшей интерпретации выразим в процентах данные метрики.

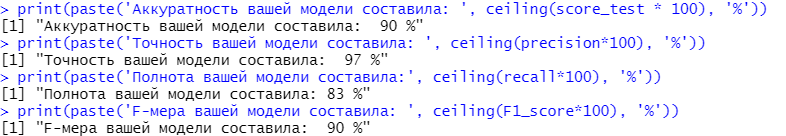


Рисунок 12 Вывод метрик качества классификации

На рисунке 12 показан результат тестирования перцептрона. Видно, что условие работы модели выполнены – все необходимые метрики более 70%, что означает, что модель работает правильно.

Необходимо проверить, соответствует ли предсказанное распределение объективному. Для этого результат необходимо визуализировать.

Создадим датафрейм, в котором будет 2 признака: предсказанные значения (predictions) и реальные значения (y\_test). Теперь построим гистограмму распределений каждого из признаков и убедимся в том, что они действительно схожи.

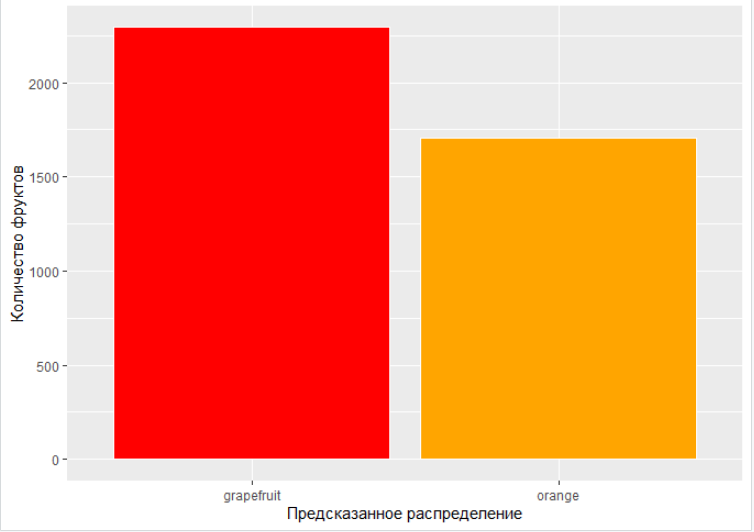


Рисунок 13 Предсказанное распределение объектов по классам

На рисунке 13 показана классификация тестовых данных с помощью перцептрона. Видно, что перцептрон определил грейпфрутов больше, чем апельсинов. Сравним результат с реальным.

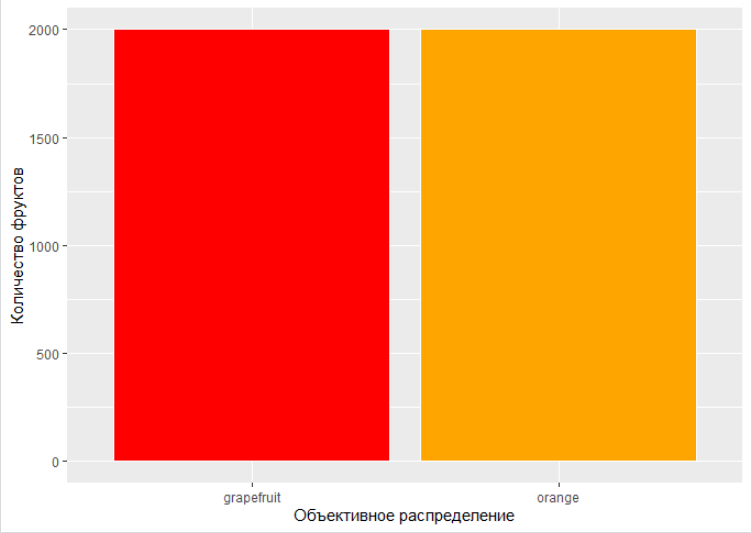


Рисунок 14 Реальное распределение объектов по классам

Рисунок 14 показывает достоверное распределение объектов по классам.

Код отрисовки гистограмм, а также программный код реализации перцептрона представлен в Приложении 1.

Таким образом, классификационный анализ цитрусов показал, что перцептрон справляется с задачей бинарной классификации. Данный алгоритм широко применяется в различных сферах. Перцептрон – один из наиболее популярных подходов для решения задач бинарной или мульти-классовой классификации. Перед работой алгоритм необходимо обучить, а затем протестировать и рассчитать метрики качества модели. Если все необходимые условия соблюдены, перцептрон готов к работе.

# Заключение

С течением времени в России все сильнее растет объем продукции, производимой предприятиями. Сегодня одной из серьезных проблем для российских предприятий является создание системы автоматического определения категории товара, позволяющей обеспечить безошибочную сортировку различных объектов. Очередной товар (объект) необходимо отнести к верному классу в зависимости от его характеристик.

В результате данного курсового проекта проведена реализация перцептрона на языке программирования высокого уровня R, решена задача бинарной классификации на примере цитрусов.

Обучение перцептрона зависит от параметров, которые задает пользователь – количества итераций и шага обучения (learning rate). На каждом шаге алгоритм проверяет, правильно ли определен класс объекта и, если получен отрицательный результат, то веса перцептрона корректируются до тех пор, пока верный результат не будет получен.

Цель – реализовать перцептрон на языке программирования высокого уровня R– достигнута.

В ходе данной курсовой работы выполнены следующие задачи:

* изучена научная и методическая литература по проблеме бинарной классификации данных;
* использованы знания математической статистики и линейной алгебры;
* реализован перцептрон;
* выполнена предобработка данных;
* обучение перцептрона;
* решена задача бинарной классификации на продуктов питания (цитрусов) с использованием перцептрона
* оценено качество обученной модели с помощью метрик качества;
* пройдено обучение качественному оформлению документации.

# Список используемой литературы

1. Горяинова, Е.И. Методы бинарной классификации объектов с номинальными показателями / Е.И. Горяинова // Журнал Новой экономической ассоциации № 2 (14), C. 27–49. – 2017. – С. 35
2. Warren S. McCulloch and Walter Pitts. [Логическое исчисление идей, относящихся к нервной активности](http://www.raai.org/library/books/mcculloch/mcculloch.pdf) = [A logical calculus of the ideas immanent in nervous activity](http://www.springerlink.com/content/61446605110620kg/fulltext.pdf) // Bulletin of Mathematical Biology. – New York: Springer New York, 1943. – Т. 5, № 4. – С. 115–133.
3. Жуков, Д.А. Анализ критериев классификации при диагностике функционирования технического объекта / Д.А. Жуков // Mathematical modelling. – 2018. – С. 13
4. Роберт Кабаков. R в действии. – [ДМК-Пресс](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%9C%D0%9A-%D0%9F%D1%80%D0%B5%D1%81%D1%81&action=edit&redlink=1), 2014. – 588 с.
5. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: Классификация и снижение размерности. – Москва: Финансы и статистика, 2019 – 571 с
6. Habr.com [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/ods/blog/328372/> – Дата доступа: 17.04.2021.
7. Мастицкий, С.Э. СТАТИСТИЧЕСКИЙ АНАЛИЗ И ВИЗУАЛИЗАЦИЯ ДАННЫХ С ПОМОЩЬЮ R / С.Э. Мастицкий. – Москва: 2017. – 172 с.
8. Kaggle.com [Электронный ресурс]. – Режим доступа: https://www.kaggle.com/joshmcadams/oranges-vs-grapefruit?select=citrus.csv. – Дата доступа: 17.04.2021.

# Приложение

Листинг программы:

#lib for class creating

library('R6')

#library for graphics

library(ggplot2)

# random seed

set.seed(42)

#Implementation of perceptron

Perceptron <- R6Class(classname = 'Perceptron',

public = list(

num\_features = NA,

learning\_rate = NA,

t\_max = NA,

weights = NA,

b = NA,

t = NA,

initialize = function(num\_features, learning\_rate, t\_max){

self$num\_features <- num\_features

self$learning\_rate <- learning\_rate

self$t\_max <- t\_max

self$weights <- rep(0, num\_features)

self$b <- 0

self$t <- 0

},

fit = function(X, y){

while (self$t < self$t\_max){

self$t <- self$t + 1

idx <- sample(c(1:(length(y))), 1)

if ((y[idx] \* (X[idx, ] %\*% self$weights + self$b)) <= 0){

self$b <- self$b + self$learning\_rate \* y[idx]

self$weights <- self$weights + self$learning\_rate \* y[idx] \* X[idx, ]

}

}

print(self)

return (self)

},

predict = function(X\_test){

classes <- c()

for (i in c(1:length(X\_test[,1]))){

if (((X\_test[i, ] %\*% self$weights) + self$b) >= 0)

classes[i] <- 1

else

classes[i] <- -1

}

return (classes)

}

))

#IN: Train Dataframe with 5 features and 1 binary target: 'orange' and 'grapefruit'

#OUT: Learned model "Perceptron" which classify Test data on 1 or -1 (orange or grapefruit).

data <- read.csv('C:/Users/plomo/Downloads/citrus.csv')

data$name <- ifelse(data$name == 'orange', 1, -1)

shuffle\_index <- sample(x = 1:nrow(data), replace = F, size = nrow(data))

ratio <- 0.6

train\_idx <- shuffle\_index[1:(ratio \* length(shuffle\_index))]

test\_idx <- shuffle\_index[-(1:(ratio \* length(shuffle\_index)))]

X\_train <- as.matrix(data[train\_idx, -1])

X\_test <- as.matrix(data[test\_idx, -1])

y\_train <- as.vector(data$name[train\_idx])

y\_test <- as.vector(data$name[test\_idx])

model <- Perceptron$new(num\_features=ncol(X\_train), learning\_rate=0.1, t\_max = 4000)

model$fit(X\_train, y\_train)

predictions <- as.factor(model$predict(X\_test))

library(caret)

precision <- posPredValue(predictions, as.factor(y\_test), positive="1")

recall <- sensitivity(predictions, as.factor(y\_test), positive="1")

F1\_score <- (2 \* precision \* recall) / (precision + recall)

calc\_acc = function(actual, predicted) {

mean(actual == predicted)

}

score\_test <- calc\_acc(y\_test, predictions)

print(paste('Аккуратность вашей модели составила: ', ceiling(score\_test \* 100), '%'))

print(paste('Точность вашей модели составила: ', ceiling(precision\*100), '%'))

print(paste('Полнота вашей модели составила:', ceiling(recall\*100), '%'))

print(paste('F-мера вашей модели составила: ', ceiling(F1\_score\*100), '%'))

pred\_data <- data.frame(predictions, y\_test)

ggplot(pred\_data, aes(predictions))+

geom\_bar(aes(ifelse(predictions == 1, 'orange', 'grapefruit')), col = 'white', fill = c('red', 'orange'))+

xlab("Предсказанное распределение")+

ylab('Количество фруктов')

ggplot(pred\_data, aes(y\_test))+

geom\_bar(aes(ifelse(y\_test == 1, 'orange', 'grapefruit')), col = 'white', fill = c('red', 'orange'))+

xlab("Объективное распределение")+

ylab('Количество фруктов')