МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени
Федеральное Государственное бюджетное образовательное учреждение высшего образования
«Московский Технический Университет Связи и Информатики»

Кафедра «Информационная безопасность»

Дисциплина «Интеллектуальные технологии информационной безопасности»

Лабораторные работы

Выполнил студент
группы М102201(70)
Пальчун Д.А.

Преподаватель:
Раковский Д.И.

Москва,
2022

Содержание

Лабораторная работа 1. «Статистический анализ наборов данных»

Цель работы: усвоить основные способы статистической оценки данных перед их обработкой интеллектуальными алгоритмами анализа данных.

Теоретическая часть

Говорят, что компьютерная программа обучается на опыте E относительно некоторого класса задач T и меры качества P, если ее качество на задачах, принадлежащих T, измеренное в соответствии с P, улучшается с увеличением опыта E [1].

Таким образом, существует много видов машинного обучения, зависящих от природы задачи T, решению которой необходимо обучить систему, природы меры качества P, используемой для оценки работы системы, и природы обучающего сигнала, или опыта E, на котором обучается система [2].

Перед применением инструментов машинного обучения (МО) к какой-либо задаче, необходимо провести разведочных анализ данных с целью выяснения наиболее очевидных закономерностей в наборе данных. Для первичного анализа данных потребуется знание основ математического анализа, математической статистики, теории информации, дискретной математики

График парных отношений.

Для табличных данных с небольшим числом признаков часто строят график парных отношений, на котором каждый подграф (i, j) содержит диаграмму рассеяния величин i и j, а диагональные элементы (i, i) показывают маргинальное распределение величины i. Обобщенная запись маргинального распределения для случайной величины записывается как:

$$p(X = x) = \sum p(X = x, Y = y) \qquad (1.1)$$

где суммирование производится по всем возможным значениям Y.

Ранее в русскоязычной литературе термин marginal distribution переводился как «частное распределение», сейчас более употребителен перевод «маргинальное распределение»); дополнительно все графики могут быть маркированы цветом, ассоциированным с меткой класса. Визуализация графика парных отношений приведена на рис. 1.1.

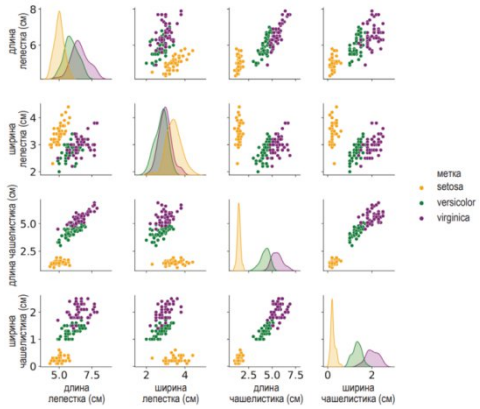

Рисунок 1.1 Визуализация построения попарной диаграммы рассеяния на наборе Iris.

Частотная гистограмма распределения классов.

Частотное распределение представляет собой столбчатую диаграмму для отображения частотности попадания наблюдаемых значений в определенные интервалы или классы. Пример построенной частотной гистограммы распределения классов для бинарного случая приведен на рис. 1.2.

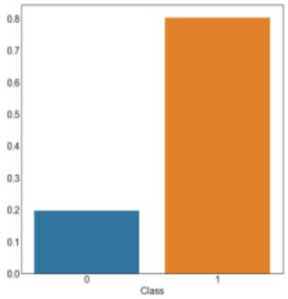

Рисунок 1.2 Визуализация построения частотной гистограммы распределения классов.

Корреляционная диаграмма.

Для оценки значимости и оценки количества атрибутов используется построение корреляционной диаграммы на основе оценок коэффициентов корреляции атрибутов методами Пирсона, Кендалла и Спирмена. Коэффициент корреляции Пирсона характеризует существование линейной зависимости между значениями двух выборок одинаковой размерности.

Ранговые корреляции Спирмена и Кенделла отличаются различной интерпретацией. Так если коэффициент корреляции Спирмена может рассматриваться как прямой аналог коэффициента корреляции Пирсона, вычисленный по рангам, то при вычислении коэффициента корреляции Кендалла проверяется наличие различий между вероятностями порядка (ранга) расположения наблюдаемых данных. Следует отметить, что в корреляции методом Спирмена инверсиям придаются дополнительные веса. В результате коэффициент корреляции Спирмена значительнее реагирует на несогласие выборок, чем коэффициент корреляции Кендалла. В большинстве случаев $|\rho| > |\tau|$. Пример построения трех корреляционных диаграмм для одного и того же набора данных приведен на рис. 1.3.

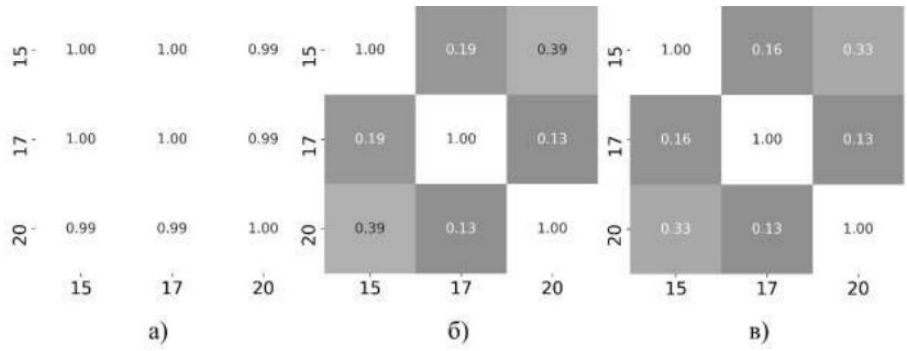

Рисунок 1.3 Визуализация построения корреляционной диаграммы. Корреляционные диаграммы, выполненные для атрибутов №15, №17, №20 методом: а) Пирсона; б) Спирмена; в) Кендалла.

Ход работы:

Dataset: IMDB Video Games

(https://www.kaggle.com/datasets/muhammadadiltalay/imdb-video-games)

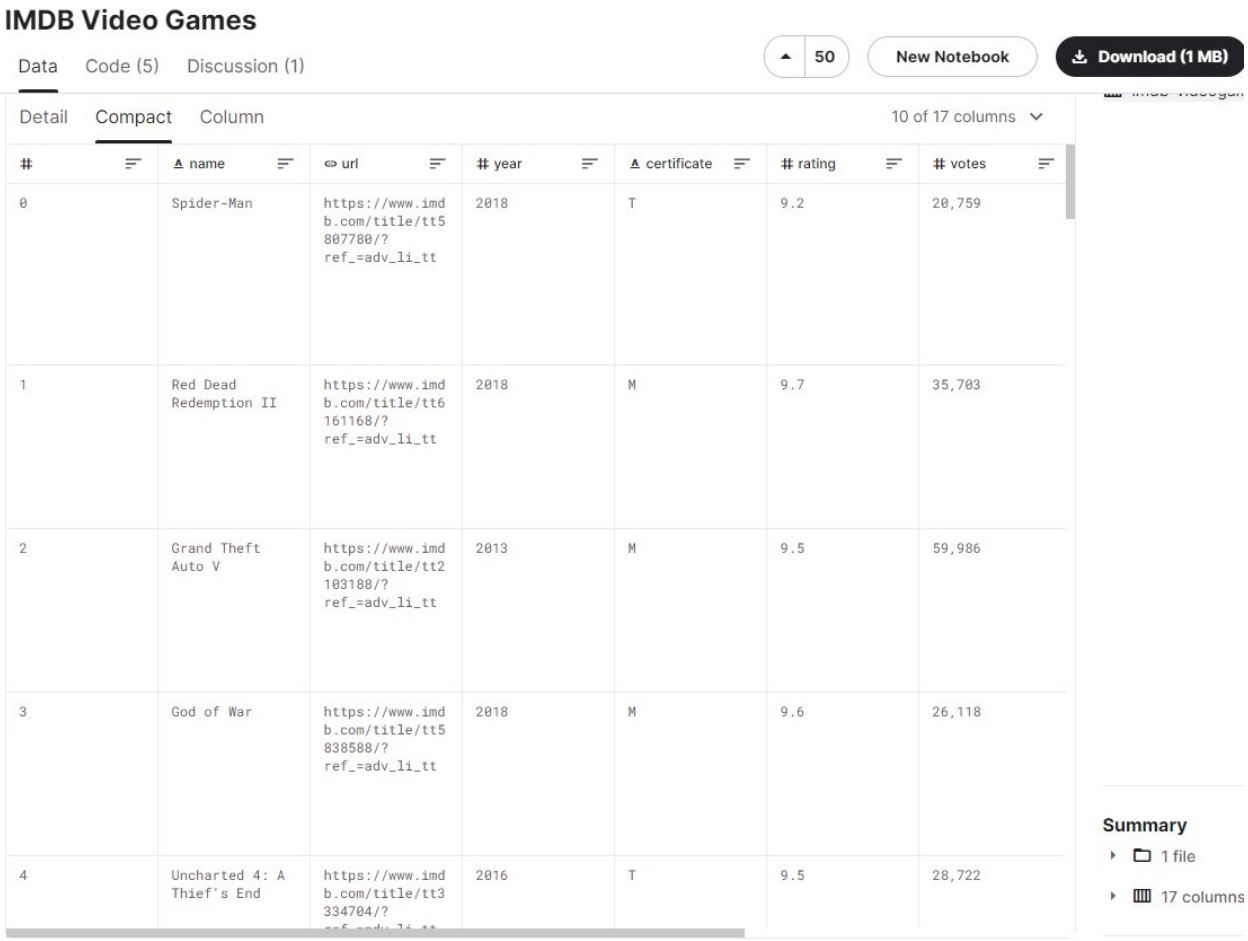

Рисунок 1.3 Dataset: IMDB Video Games

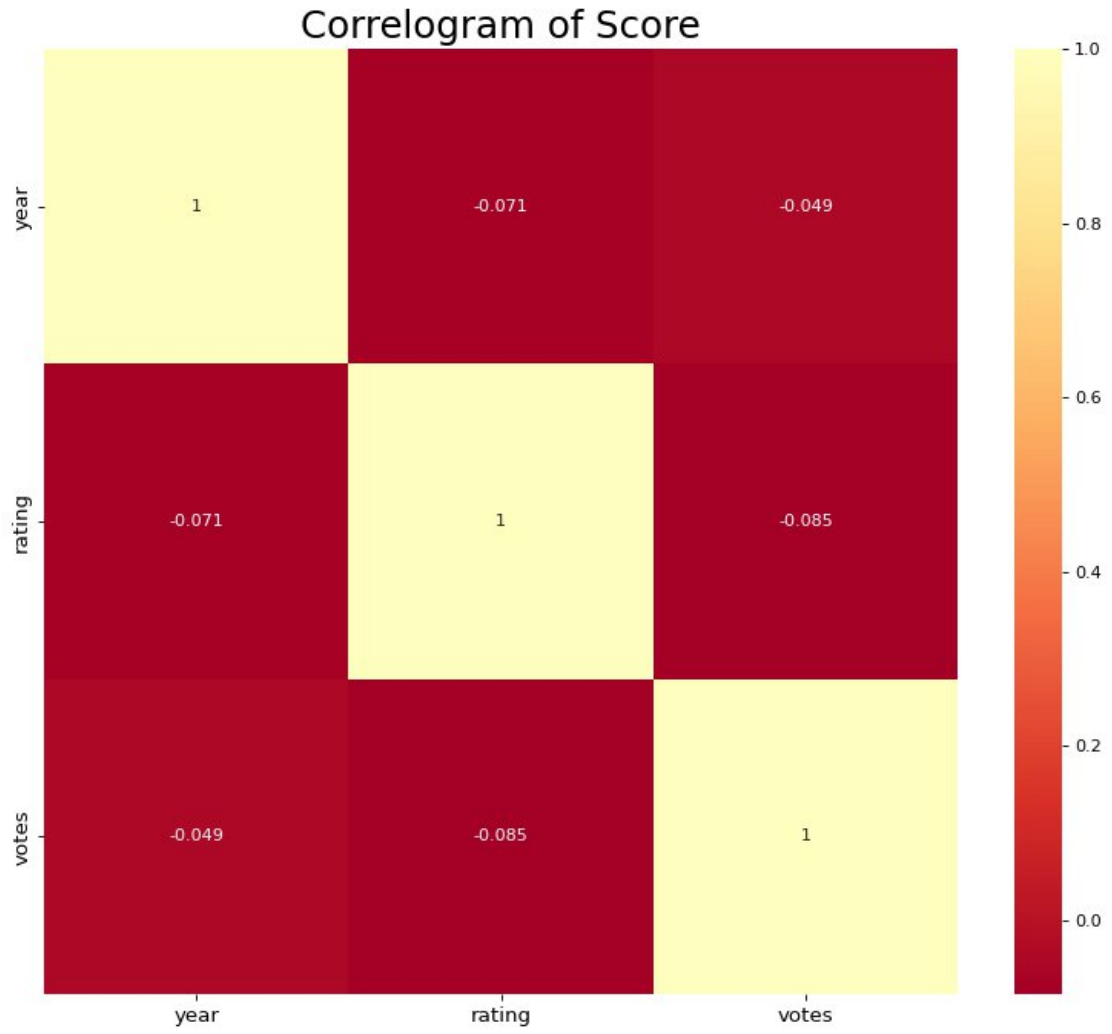

Рисунок 1.4 Корреляционная диаграмма

В данном наборе данных (рисунок 1.4) очевидно, что между годами выхода игр, количеством голосов на игру и рейтингом вышедших игр нет корреляции. Таким образом, мы можем утверждать, что заявление, которое не подкрепляется дополнительными аргументами, о том, что раньше игры делали лучше - ложно, также ложно утверждение, что нужно ориентироваться на качество игры исходя только из того, что за нее обращает внимание большое количество людей.

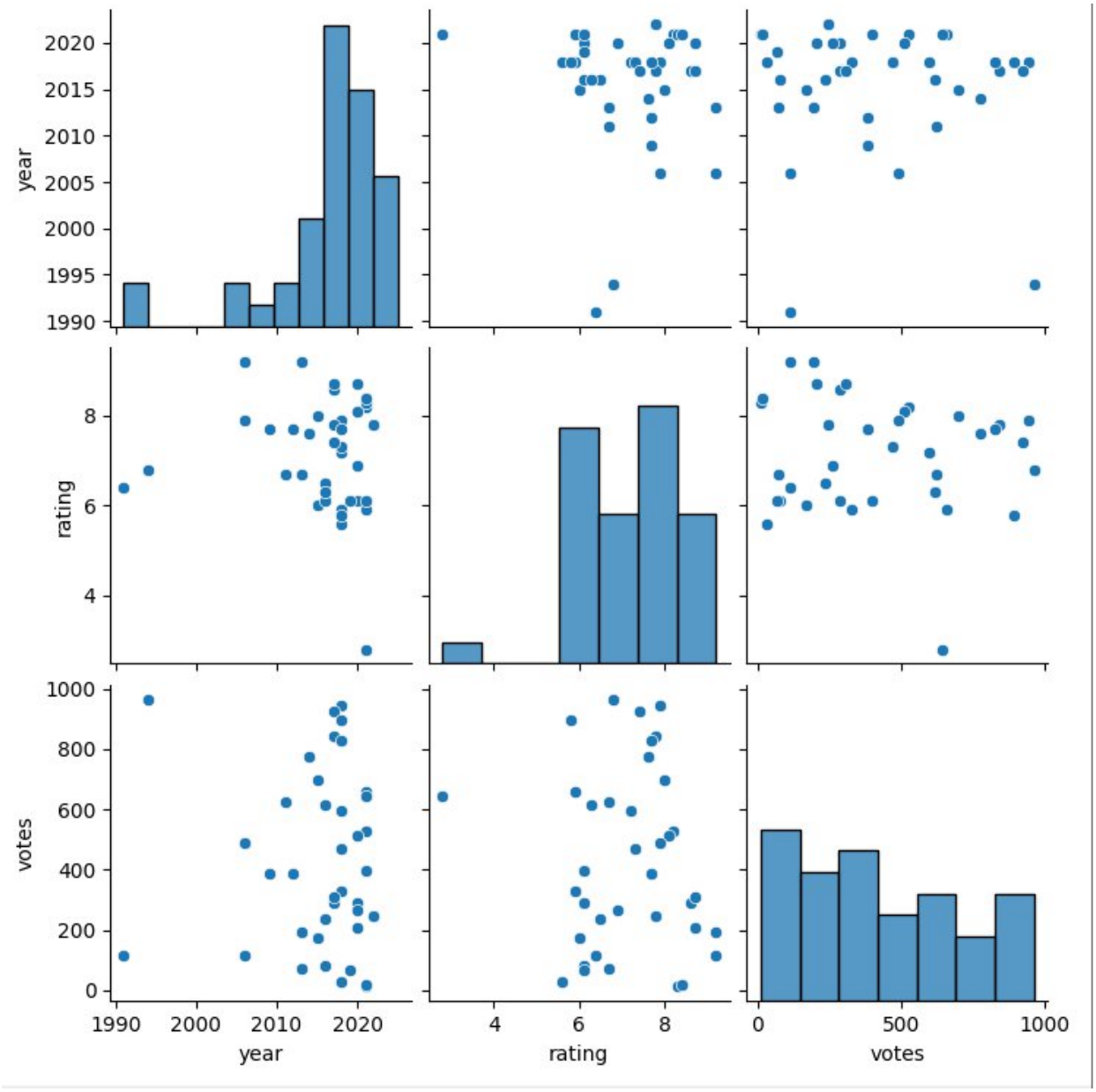

Рисунок 1.5 График парных отношений

На графике парных отношений (рисунок 1.5) мы можем наблюдать, что с 2015 года количество голосующих за рейтинг игр увеличилось, также видна тенденция того, что с течением лет количество оценок игр увеличивается, но высота оценки уменьшается. Также, видно, что люди неохотно ставят играм низкие оценки.

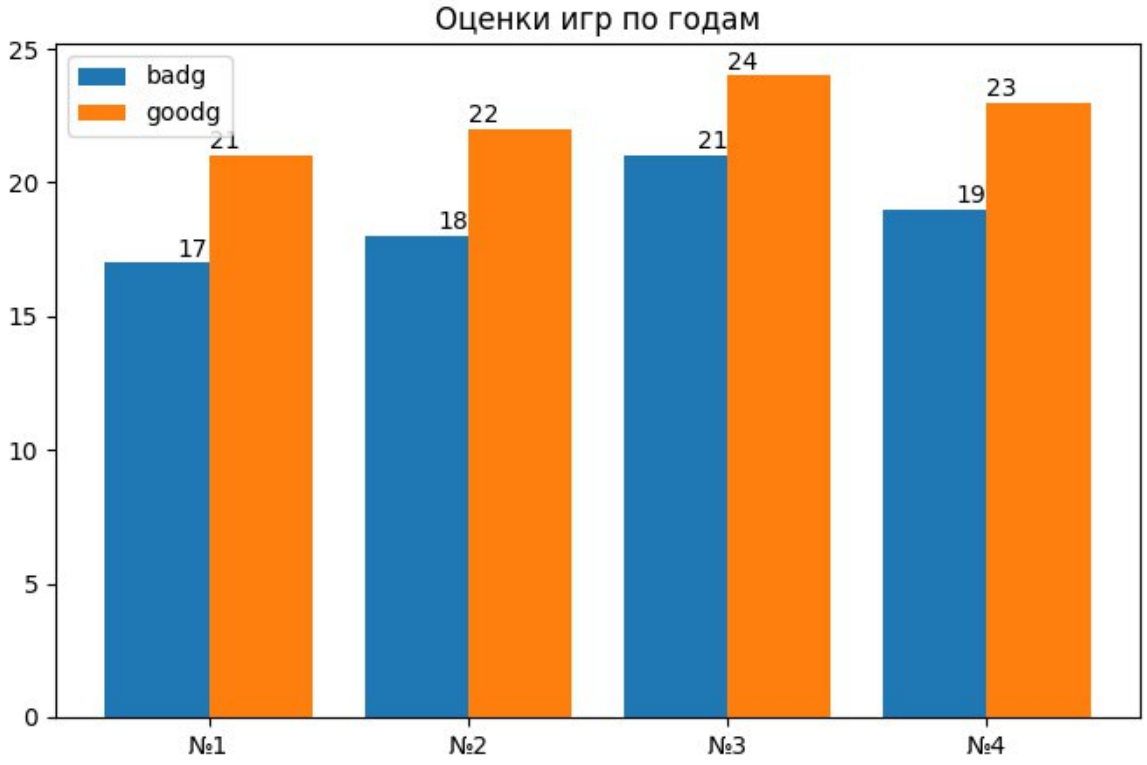

Рисунок 1.6 Частотная гистограмма распределения классов. №1 – Общий результат трех лет. №2 – Результат за 2013. №3 – результаты за 2018. №4 – результаты за 2020.

На рисунке 1.6 продемонстрирована гистограмма общего результата оценок за 3 года. Здесь мы видим, что количество игр, набравших более 7 баллов с течением лет увеличивается, как и общее количество игр.

Вывод

По результатам выполнения лабораторной работы были усвоены основные способы статистической оценки данных. Был проведен разведочный анализ данных с целью выяснения наиболее очевидных закономерностей в наборе данных. Проанализированы: график парных отношений, частотная гистограмма распределения классов и корреляционная диаграмма.

Лабораторная работа 2. «Исследование однослойных нейронных сетей на примере моделирования булевых выражений»

Цель работы: исследовать функционирование простейшей нейронной сети (НС) на базе нейрона с нелинейной функцией активации посредством обучения ее по правилу Видроу-Хоффа.

Постановка задачи: получить модель булевой функции (БФ) на основе однослойной НС (единичный нейрон = персептрон (рис. 1)) с двоичными входами $x1,x2,x3,x4 \in \{0,1\}$, единичным входом смещения $x0=1$, синаптическими весами $w0$, $w1$, $w2$, $w3$, $w4$, двоичным выходом $y \in \{0,1\}$ и заданной нелинейной функцией активации (ФА) $f: R \to (0,1)$.

Для заданной БФ реализовать обучение НС с использованием:

- всех комбинаций переменных $x1,x2,x3,x4$;
- части возможных комбинаций переменных $x1,x2,x3,x4$, остальные комбинации являются тестовыми.

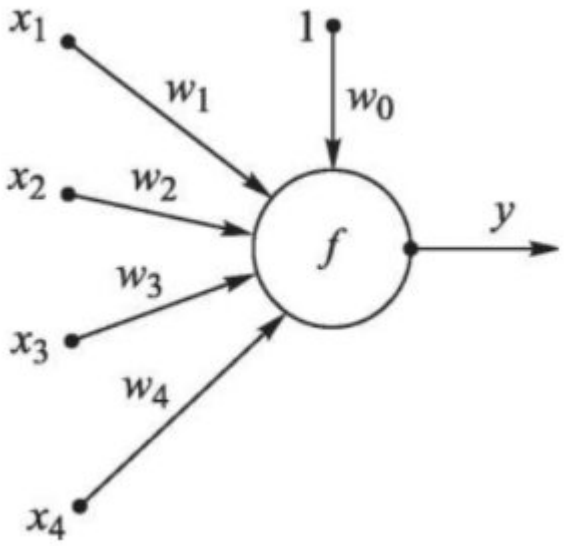

Рисунок 2.1. Однослойная нейронная сеть (персептрон).

Режимы работы НС: рабочий режим и режим обучения.

Рабочий режим: функционирование НС с пороговой ФА имеет вид:

$$net = \sum_{i=1}^{4} w_i x_i + w_0$$

$$y(net) = \{1,\ net \geq 0; 0,\ net < 0$$

Функционирование НС с логистической ФА имеет вид:

$$net = \sum_{i=1}^{4} w_i x_i + w_0$$

$$out = f(net)$$

$$y(out) = \{1,\ net \geq 0\ 0,\ net < 0$$

где out — сетевой (недискретизированный) выход НС.

Режим обучения: Для необученной НС её реальный выход y в общем случае отличается от целевого выхода – *t* - булевой функции нескольких переменных $F(x_1,x_2,x_3,x_4):[0,1]^4 \rightarrow \{0,1\}$. Имеется хотя бы один набор сигналов $(x1,x2,x3,x4)$, для которого ошибка $\delta = t - y \neq 0$.

Каждая эпоха обучения k = 1, 2, … включает в себя цикл последовательного предъявления всех образцов обучающей выборки на вход НС. Предъявление одного обучающего образца внутри эпохи является элементарным шагом обучения l = 0, 1, 2, … , во время которого вектор весовых коэффициентов $\mathbf{w} = (wi)i$ = 0, …, 4 корректируется согласно правилу Видроу — Хоффа (дельта-правило):

$$w_i^{(l+1)} = w_i^{(l)} + \Delta w_i^{(l)}$$

$$\Delta w_i^{(l)} = \eta \delta^{(l)} \frac{df(net)}{dnet} x_i^{(l)}$$

$\Delta w_i^{(l)}$ – коррекция веса на l-том шаге;

$\eta \in (0;1]$ – норма обучения;

$\delta^{(l)}$- ошибка.

На каждой эпохе k суммарная квадратичная ошибка E(k) равна расстоянию Хэмминга между векторами целевого и реального выходов.

Исходные данные:

№ варианта	Моделируемая БФ	ФА*
13	$\left(\underline{x_1} + \underline{x_2} + \underline{x_3}\right)\left(\underline{x_2} + \underline{x_3} + x_4\right)$	1, 2
		$1).\ f_1(net) = \{1,\ net \geq 0,\ 0,\ net < 0;$ $2).\ f(net) = \frac{1}{2}\left(\frac{net}{1+\lvert net\rvert} + 1\right);$

Ход решения

Построим таблицу истинности заданной БФ:

№	X_1	X_2	X_3	X_4	$\neg X_1$	$\neg X_2$	$\neg X_3$	$\neg X_1 \vee \neg X_2 \vee \neg X_3$	$\neg X_2 \vee \neg X_3 \vee X_4$	$(\neg X_1 \vee \neg X_2 \vee \neg X_3) \wedge (\neg X_2 \vee \neg X_3 \vee X_4)$
0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1
3	0	0	1	1	1	1	0	1	1	1
4	0	1	0	0	1	0	1	1	1	1
5	0	1	0	1	1	0	1	1	1	1
6	0	1	1	0	1	0	0	1	0	0
7	0	1	1	1	1	0	0	1	1	1
8	1	0	0	0	0	1	1	1	1	1
9	1	0	0	1	0	1	1	1	1	1
10	1	0	1	0	0	1	0	1	1	1
11	1	0	1	1	0	1	0	1	1	1
12	1	1	0	0	0	0	1	1	1	1
13	1	1	0	1	0	0	1	1	1	1
14	1	1	1	0	0	0	0	0	0	0
15	1	1	1	1	0	0	0	0	1	0

Практическое решение исполнено при помощи Microsoft Visual Studio 2017 на языке C++.

Вектор начальных весов взят равным (l=0; k=0):

$$w_0^{(0)} = w_1^{(0)} = w_2^{(0)} = w_3^{(0)} = w_4^{(0)} = 0$$

Используем пороговую функцию активации $f_1(net)$. Норма обучения взята $\eta = 0.3$. Динамика персептрона отражена на рисунках 2.2-2.6. График суммарной ошибки отражен на рисунке 2.7.

w0	w1	w2	w3	w4	net	y	t
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	0
-0.6	-0.3	-0.6	-0.6	0	-2	0	0

Эпоха: 1
Ошибок всего: 2

Рисунок 2.2. Параметры НС на эпохе 1

w0	w1	w2	w3	w4	net	y	t
-0.6	-0.3	-0.6	-0.6	0	0	1	1
-0.6	-0.3	-0.6	-0.6	0	0	1	1
-0.6	-0.3	-0.6	-0.6	0	-1	0	1
-0.3	-0.3	-0.6	-0.3	0	0	1	1
-0.3	-0.3	-0.6	-0.3	0	0	1	1
-0.3	-0.3	-0.6	-0.3	0	0	1	1
-0.3	-0.3	-0.6	-0.3	0	-1	0	0
-0.3	-0.3	-0.6	-0.3	0	-1	0	1
0	-0.3	-0.3	0	0.3	0	1	1
0	-0.3	-0.3	0	0.3	0	1	1
0	-0.3	-0.3	0	0.3	0	1	1
0	-0.3	-0.3	0	0.3	0	1	1
0	-0.3	-0.3	0	0.3	0	1	1
0	-0.3	-0.3	0	0.3	0	1	1
0	-0.3	-0.3	0	0.3	0	1	0
-0.3	-0.6	-0.6	-0.3	0.3	-1	0	0

Эпоха: 2
Ошибок всего: 3

Рисунок 2.3. Параметры НС на эпохе 2

w0	w1	w2	w3	w4	net	y	t
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	-1	0	0
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	0	1	1
-0.3	-0.6	-0.6	-0.3	0.3	-1	0	1
0	-0.3	-0.6	0	0.3	0	1	1
0	-0.3	-0.6	0	0.3	0	1	1
0	-0.3	-0.6	0	0.3	0	1	1
0	-0.3	-0.6	0	0.3	0	1	0
-0.3	-0.6	-0.9	-0.3	0.3	-1	0	0

Эпоха: 3
Ошибок всего: 2

w0	w1	w2	w3	w4	net	y	t
1.8	-1.2	-1.5	-1.2	0.9	1	1	1
1.8	-1.2	-1.5	-1.2	0.9	2	1	1
1.8	-1.2	-1.5	-1.2	0.9	0	1	1
1.8	-1.2	-1.5	-1.2	0.9	1	1	1
1.8	-1.2	-1.5	-1.2	0.9	0	1	1
1.8	-1.2	-1.5	-1.2	0.9	1	1	1
1.8	-1.2	-1.5	-1.2	0.9	0	1	0
1.5	-1.2	-1.8	-1.5	0.9	0	1	1
1.5	-1.2	-1.8	-1.5	0.9	0	1	1
1.5	-1.2	-1.8	-1.5	0.9	1	1	1
1.5	-1.2	-1.8	-1.5	0.9	-1	0	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	-2	0	0
1.8	-0.9	-1.8	-1.2	0.9	-1	0	0

Эпоха: 29
Ошибок всего: 2

Рисунок 2.4. Параметры НС на эпохе 3 Рисунок 2.5. Параметры НС на эпохе 29

w0	w1	w2	w3	w4	net	y	t
1.8	-0.9	-1.8	-1.2	0.9	1	1	1
1.8	-0.9	-1.8	-1.2	0.9	2	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	1	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	-1	0	0
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	1	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	0	1	1
1.8	-0.9	-1.8	-1.2	0.9	-2	0	0
1.8	-0.9	-1.8	-1.2	0.9	-1	0	0
Эпоха: 30							
Ошибок всего: 0							

Рисунок 2.6. Параметры НС на эпохе 30

Итоговый весовой вектор:

$$\vec{w} = (w_0; w_1; w_2; w_3; w_4) = (1.8; -0.9; -1.8; -1.2; 0.9)$$

Рисунок 7. Суммарная ошибка при пороговой ФА ($f_1(net)$)

Аналогичные таблицы (Рисунок. 2.8-2.10) построим для функции $f_2(net)$. График суммарной ошибки отражен на рисунке 2.11.

w0	w1	w2	w3	w4	net	y	t
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0
-0.15	0	-0.15	-0.15	0	0	1	1
-0.15	0	-0.15	-0.15	0	0	1	1
-0.15	0	-0.15	-0.15	0	0	1	1
-0.15	0	-0.15	-0.15	0	0	1	1
-0.15	0	-0.15	-0.15	0	0	1	1
-0.15	0	-0.15	-0.15	0	0	1	1
-0.15	0	-0.15	-0.15	0	0	1	1
-0.15	0	-0.15	-0.15	0	0	1	0
-0.3	-0.15	-0.3	-0.3	0	-1	0	0

Эпоха: 1
Ошибок всего: 2

Рисунок 2.8. Параметры НС на эпохе 1

w0	w1	w2	w3	w4	net	y	t
-0.3	-0.15	-0.3	-0.3	0	0	1	1
-0.3	-0.15	-0.3	-0.3	0	0	1	1
-0.3	-0.15	-0.3	-0.3	0	0	1	1
-0.3	-0.15	-0.3	-0.3	0	0	1	1
-0.3	-0.15	-0.3	-0.3	0	0	1	1
-0.3	-0.15	-0.3	-0.3	0	0	1	1
-0.3	-0.15	-0.3	-0.3	0	0	1	0
-0.45	-0.15	-0.45	-0.45	0	-1	0	1
-0.4125	-0.15	-0.4125	-0.4125	0.0375	0	1	1
-0.4125	-0.15	-0.4125	-0.4125	0.0375	0	1	1
-0.4125	-0.15	-0.4125	-0.4125	0.0375	0	1	1
-0.4125	-0.15	-0.4125	-0.4125	0.0375	0	1	1
-0.4125	-0.15	-0.4125	-0.4125	0.0375	0	1	1
-0.4125	-0.15	-0.4125	-0.4125	0.0375	0	1	1
-0.4125	-0.15	-0.4125	-0.4125	0.0375	-1	0	0
-0.4125	-0.15	-0.4125	-0.4125	0.0375	-1	0	0

Эпоха: 2
Ошибок всего: 2

Рисунок 2.9. Параметры НС на эпохе 2

w0	w1	w2	w3	w4	net	y	t
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	-1	0	0
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	0	1	1
-0.3375	-0.15	-0.3375	-0.3375	0.1125	-1	0	0
-0.3375	-0.15	-0.3375	-0.3375	0.1125	-1	0	0

Эпоха: 5
Ошибок всего: 0

Рисунок 2.10. Параметры НС на эпохе 5

Итоговый весовой вектор:

$$\vec{w} = (w_0; w_1; w_2; w_3; w_4) = (-0.3375;\ -0.15; -0.3375; -0.3375; 0.1125)$$

Рисунок 2.11. Суммарная ошибка при логистической ФА ($f_2(net)$)

Для пороговой функции уменьшим размер обучающей выборки при сохранении нулевой ошибки с целью уменьшения количества эпох, необходимых для обучения. Для функции:

$$f = \left(\underline{x_1} + \underline{x_2} + \underline{x_3}\right)\left(\underline{x_2} + \underline{x_3} + x_4\right)$$

Был обнаружен следующий набор выборки и изменена его последовательность, при которой сохраняется нулевая ошибка и уменьшается время обучения:

$$x^{(1)} = (0,1,0,1)$$

$$x^{(2)} = (0,1,1,0)$$

$$x^{(3)} = (0,1,1,1)$$

$$x^{(4)} = (1,0,0,0)$$

$$x^{(5)} = (1,0,0,1)$$

При этом время обучения удалось сократить до 6 эпох. Эпохи отражены на рисунках 2.12-2.15. Рисунок 2.16 отражает график суммарной ошибки. Рисунок 2.17 отражает результаты проверки.

w0	w1	w2	w3	w4	net	y	t
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	1
-0.3	0	-0.3	-0.3	0	0	1	0
-0.6	-0.3	-0.6	-0.6	0	-1	0	1
-0.3	0	-0.3	-0.6	0	0	1	1
-0.3	0	-0.3	-0.6	0	-1	0	0

Эпоха: 1
Ошибок всего: 3

Рисунок 2.12. Параметры НС на эпохе 1

w0	w1	w2	w3	w4	net	y	t
-0.3	0	-0.3	-0.6	0	-1	0	1
0	0	0	-0.3	0.3	0	1	1
0	0	0	-0.3	0.3	0	1	0
-0.3	0	-0.3	-0.6	0.3	0	1	1
-0.3	0	-0.3	-0.6	0.3	0	1	1
-0.3	0	-0.3	-0.6	0.3	0	1	1
-0.3	0	-0.3	-0.6	0.3	-1	0	0
-0.3	0	-0.3	-0.6	0.3	0	1	1
-0.3	0	-0.3	-0.6	0.3	0	1	1
-0.3	0	-0.3	-0.6	0.3	0	1	0

Эпоха: 2
Ошибок всего: 3

Рисунок 2.13. Параметры НС на эпохе 2

w0	w1	w2	w3	w4	net	y	t
-0.3	-0.3	-0.9	-0.9	0	-2	0	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	-1	0	0
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	-1	0	0
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	-1	0	0

Эпоха: 5
Ошибок всего: 1

Рисунок 2.14. Параметры НС на эпохе 5

w0	w1	w2	w3	w4	net	y	t
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	-1	0	0
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	-1	0	0
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	-1	0	0

Эпоха: 6
Ошибок всего: 0

Рисунок 2.15. Параметры НС на эпохе 6

Итоговый весовой вектор:

$$\vec{w} = (w_0; w_1; w_2; w_3; w_4) = (0;\ -0.3,\ -0.6,\ -0.6, 0.3)$$

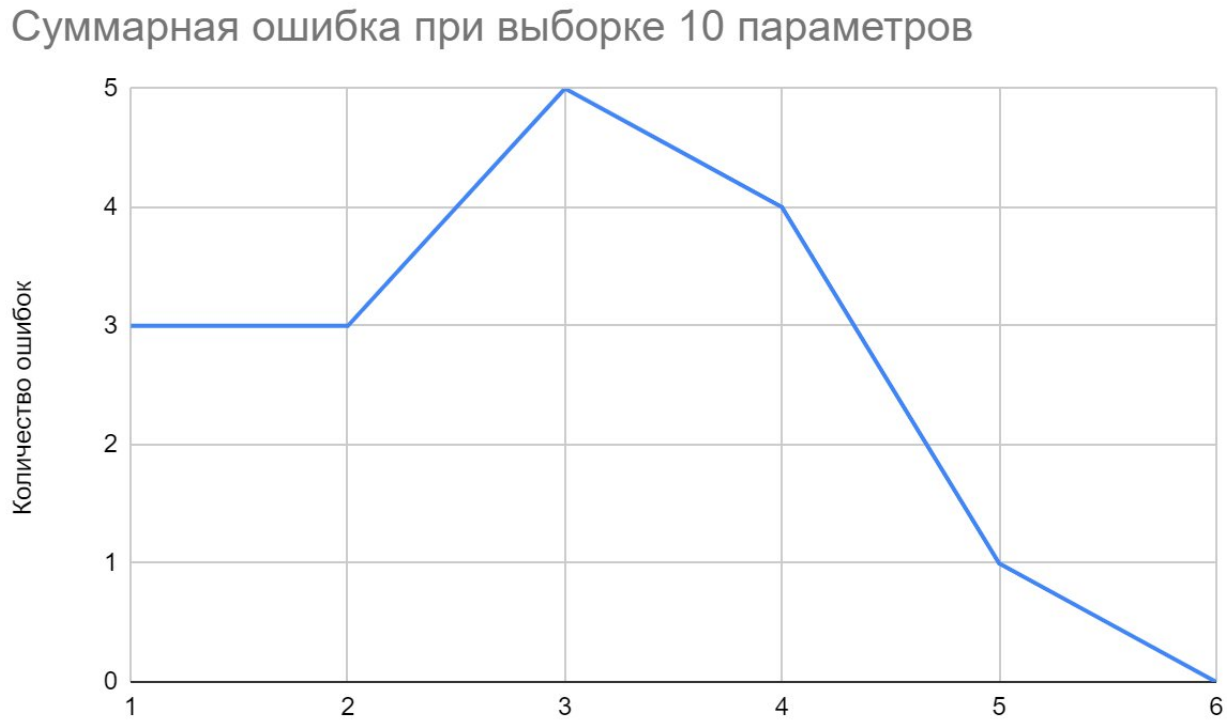

Рисунок 2.16. Суммарная ошибка НС при длине обучающей выборки равной 10

w0	w1	w2	w3	w4	net	y	t
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	-1	0	0
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	0	1	1
0	-0.3	-0.6	-0.6	0.3	-1	0	0
0	-0.3	-0.6	-0.6	0.3	-1	0	0

Эпоха: 1
Ошибок всего: 0

Рисунок 2.17. Результаты тестирования весового вектора на полной БФ.

Вывод

При использовании пороговой функции нейронная сеть обучалась достаточно долго, по сравнению с логистической. При попытке оптимизации методом выборки векторов было уменьшено время обучения на 80%, при тестировании полученного набора весов нейронная сеть не допустила ни одной ошибки. Из этого можно сделать вывод, что используя меньшее количество входных данных и определенную их последовательность, возможно сократить время обучения.

Таким образом, порядок начальных обучающих векторов определяет траекторию обучения НС.

Лабораторная работа 3. Оценка результатов классификации при помощи метрик TP, FP, FN, TN, Accuracy, Precision, AUC, F-мера, матрица ошибок и лабораторная работа 4: «Алгоритмы машинного обучения с учителем для решения задач классификации»

Цель работы: исследовать функционирование алгоритмов машинного обучения с учителем для решения задач классификации. Освоить основные методы оценки метрик точности алгоритмов классификации.

Теоретическая часть

Классификация является одной из трех основных задач машинного обучения. Задача классификации подразумевает отнесение объекта к одной из категорий на основании его признаков.

Важнейшим аспектом задачи классификации, отличающей ее от задачи кластеризации, является известность признаков объекта (в том числе и меток/маркеров класса).

Для решений задачи классификации разработано большое количество методов машинного обучения. Один из алгоритмов, относящихся к методам вычислительного интеллекта, рассматривался в лабораторной работе №2 – однослойный персептрон (однослойная НС).

Рассмотрим один из алгоритмов метода машинного обучения – деревья решений. Основная идея алгоритмов, относящихся к данной категории – использование «древовидной» вложенной структуры вида if…else, позволяющей классифицировать данные по набору примитивных правил.

Для оценки качества построенного дерева вводится понятие энтропии.

Энтропия - это среднее количество информации, приходящееся на одно сообщение, символ, слово источника информации.

Энтропия характеризует также среднюю неопределённость ситуации. Чем больше энтропия, тем больше неопределённость ситуации и,

следовательно, тем больше информации получает лицо, принимающее сообщение, устраняющее неопределённость.

Чаще всего дерево решений строится по принципу жадной максимизации прироста информации - на каждом шаге выбирается тот признак, при разделении по которому прирост информации оказывается наибольшим. Дальше процедура повторяется рекурсивно, пока энтропия не окажется равной нулю или какой-то малой величине. В разных алгоритмах применяются разные эвристики для "ранней остановки" или "отсечения", чтобы избежать построения переобученного дерева.

Ход работы:

Текущий набор данных представляет собой сведения о молоке и его запахе. В него входят такие параметры как: Температура, Вкус, Жирность, Мутность, Цвет и т.д.

Составим дерево решений, по которому можно будет определить тип гриба (рисунок 3.1).

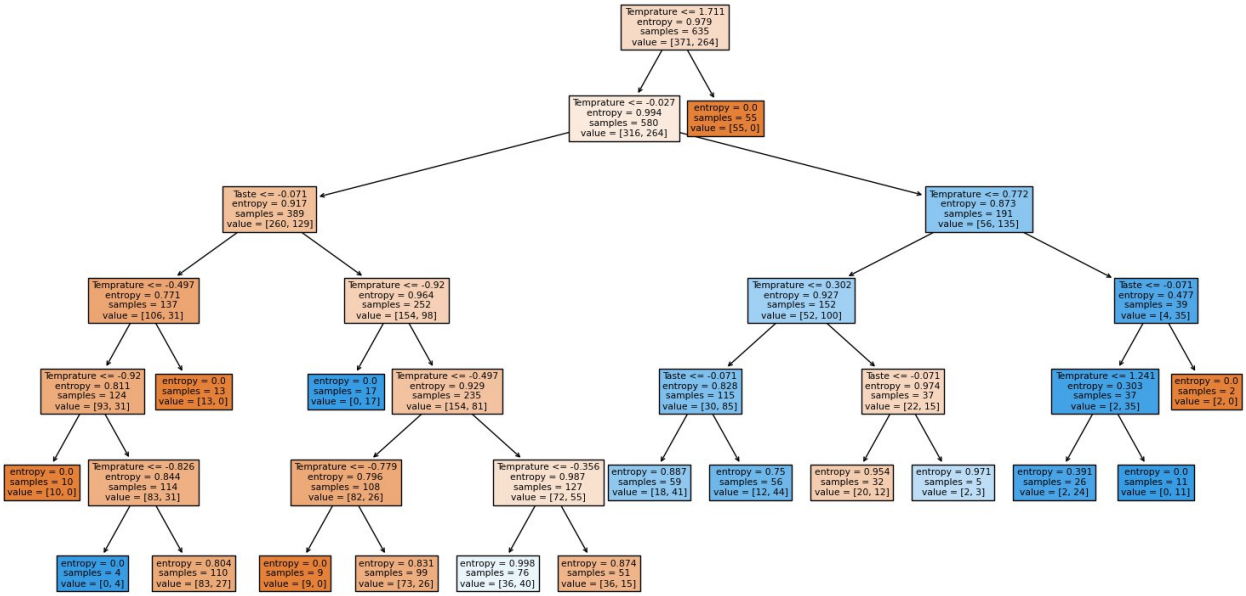

Рисунок 3.1. Дерево решений

Для построения дерева решений были выбраны следующие параметры:

1. Temperature	1. Температура
2. Taste	2. Вкус
3. Fat	3. Жирность
4. Turbidity	4. Мутность
5. Colour	5. Цвет

Оранжевый цвет - хороший запах, синий – плохой. Все данные сыграли примерно одинаково полезную роль при классификации и дали хороший результат.

Проведем анализ результатов обучения (Рис. 3.2-3.4).

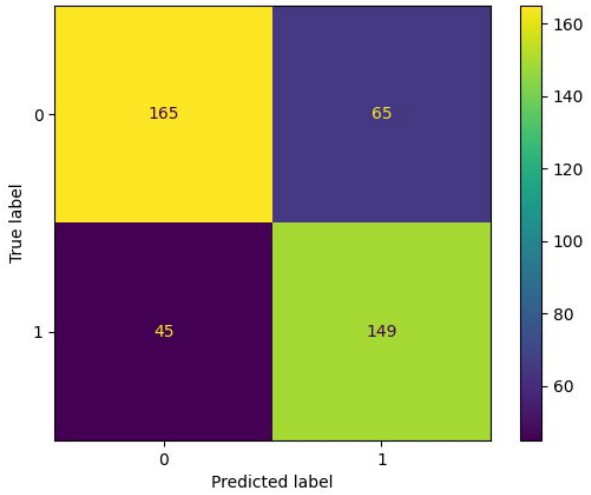

Рисунок 3.2. Матрица ошибок

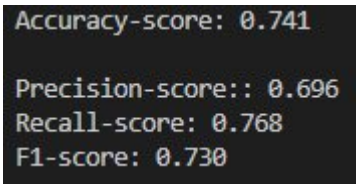

Рисунок 3.3. Метрики

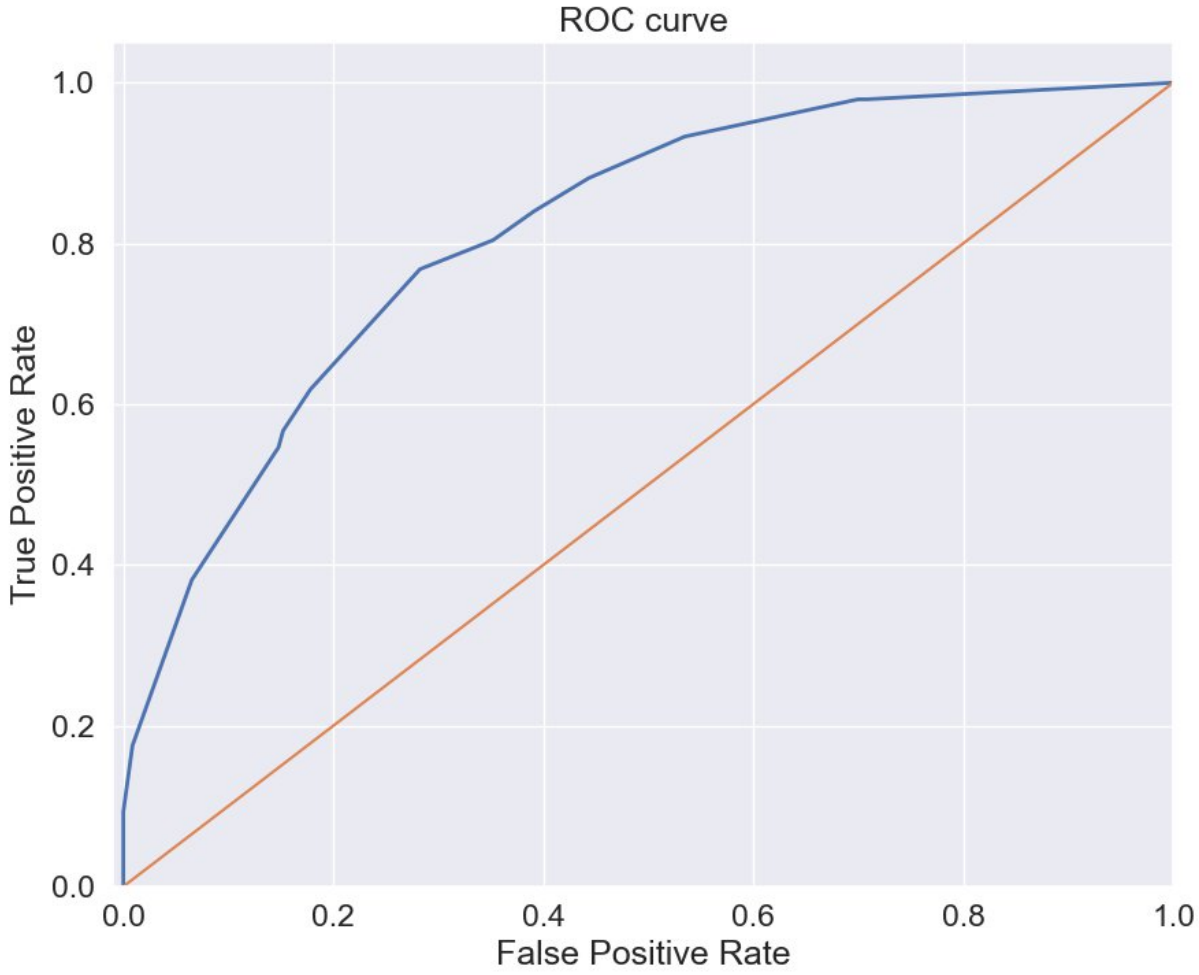

Рисунок 3.4. ROC кривая

На рисунке 3.2. (0 – хороший запах, 1 – плохой запах) видно, что доля неудовлетворительных ответов находиться на среднем уровне. Как можно заметить на рисунке 3.3. доля правильных ответов составляет 74%, что является хорошим результатом.

Вывод

По результатам выполнения лабораторных работ №3 и №4, был реализован алгоритм машинного обучения с учителем для решения задач классификации и была проведена оценка результатов классификации при помощи метрик TP, FP, FN, TN, Accuracy, Precision, AUC, F-мера, матрица ошибок. Алгоритм способен выдать в 74% верный результат, следовательно, есть 26% риск найти плохое молоко.

Лабораторная работа 8. Снижение размерности входных данных на примере алгоритмов PCA, SVD

Цель работы: с использованием библиотеки scikit-learn изучить принципы работы алгоритмов сокращения размерности SVD и PCA.

Теоретическая часть

При работе с данными высокой размерности часто бывает полезно понизить размерность путем проецирования на подпространство меньшей размерности, которое улавливает основные особенности данных. Один из подходов к этой проблеме – предположить, что все наблюдаемые многомерные выходы $xn \in \mathbb{R}D$ порождены множеством скрытых, т. е. ненаблюдаемых латентных факторов низкой размерности $zn \in \mathbb{R}K$.

Простейшим примером является применение факторного анализа. В частном случае, факторный анализ сводится к модели вероятностного метода главных компонент (principal components analysis — PCA

Сингулярное разложение (SVD), является обобщением спектрального разложения на прямоугольные матрицы. Любую вещественную матрицу A размера m´n можно разложить в произведение:

$$\boldsymbol{A = USV^T} = \sigma_1 \begin{pmatrix} | \\ \boldsymbol{u}_1 \\ | \end{pmatrix} (- \quad \boldsymbol{v}_1^T \quad -) + \cdots + \sigma_r \begin{pmatrix} | \\ \boldsymbol{u}_r \\ | \end{pmatrix} (- \quad \boldsymbol{v}_1^T \quad -),$$

где U – матрица $m \times m$ с ортонормированными столбцами (т. е. $U^T\boldsymbol{U} = \boldsymbol{Im}$), V – матрица $n \times n$ с ортонормированными строками и столбцами (т. е. $V^T\boldsymbol{V} = \boldsymbol{V}V^T = \boldsymbol{In}$), а S – матрица $m \times n$, содержащая r = min(m, n) сингулярных чисел $\sigma i \geq 0$ на главной диагонали и нули во всех остальных позициях. Столбцы U называются левыми сингулярными векторами, а столбцы V – правыми сингулярными векторами.

Ход работы:

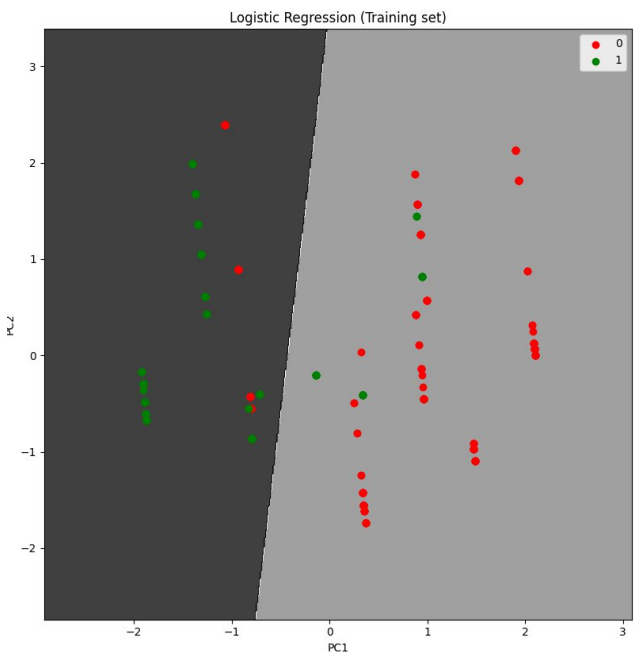

Рисунок 4.1. Тренировочный набор

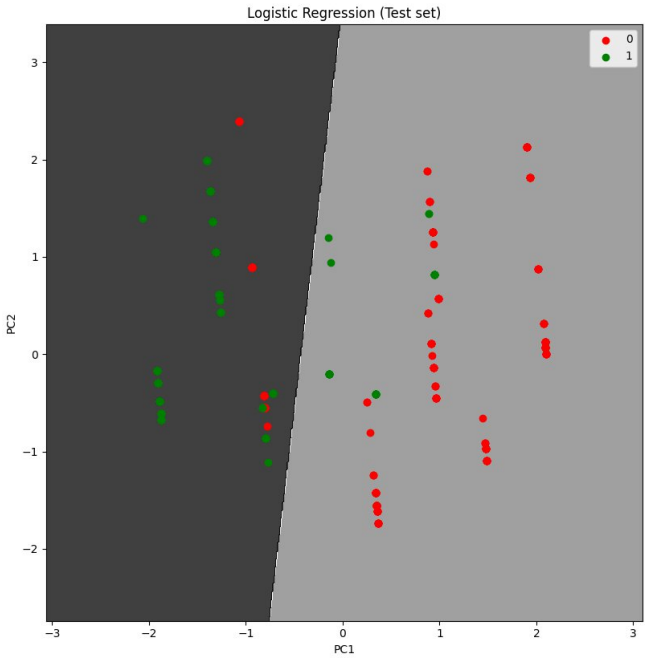

Рисунок 4.2. Тестовый набор

На рисунках 4.1-4.2 видно, что часть плохого молока (1) находятся в зоне хорошего (0), а часть хорошего на стороне плохого.

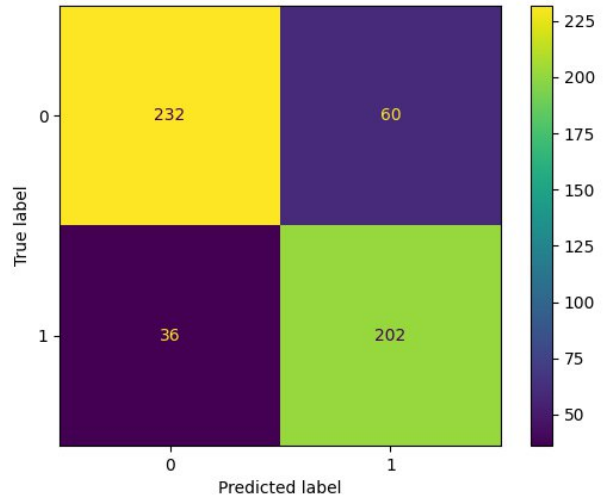

Рисунок 4.3. Метрики

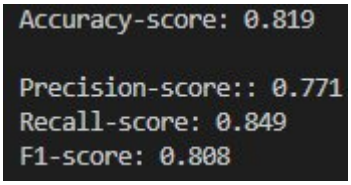

Рисунок 4.4. Метрики

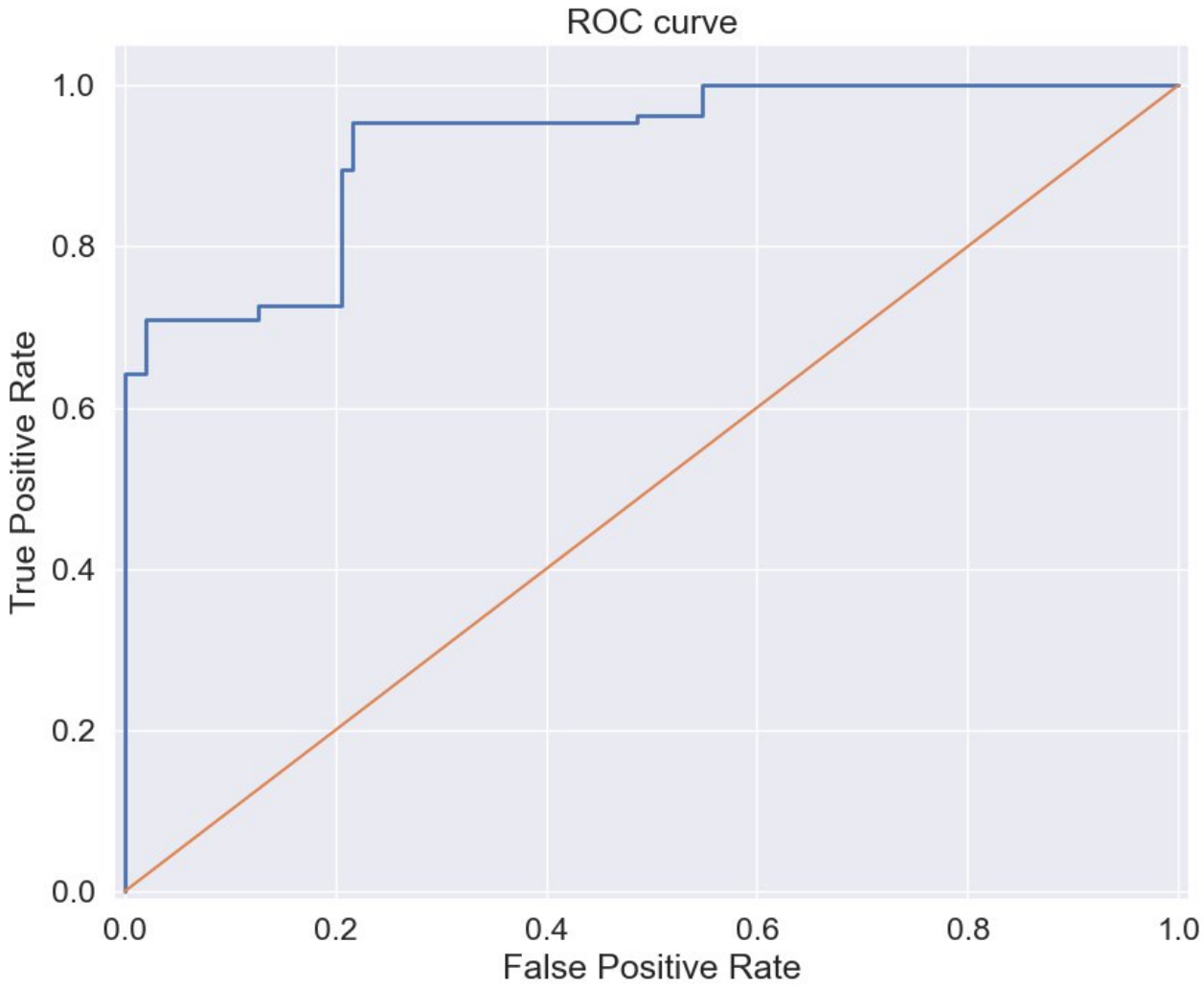

Рисунок 4.5. ROC-кривая

Анализируя метрики можно сделать вывод, что алгоритм классификации работает корректно, точность составляет 82% .

Вывод

В общем случае по данным результатам следует сказать следующее. При возникновении сомнений по поводу съедобности молока стоит его пропустить. Количество неверно классифицированного молока, а именно пропавшего, среднее. Следует отметить, что употребление пропавшего молока с таким процентом его выявления не повлечет для человека серьезных последствий, следовательно, наличие данного алгоритма лучше, чем его отсутствие.