**Федеральное агентство по образованию**Государственное образовательное учреждение высшего профессионального   
образования **«Тихоокеанский Государственный университет»**

Факультет компьютерных и фундаментальных наук

Кафедра ПОВТАС

**Лабораторная работа №1**

по дисциплине: «Методы машинного обучения»

на тему: «Расчет коэффициентов разделяющей линии и вычисление отступа (margin) для объектов разных классов»  
Вариант №4

Выполнил: студент группы ПИИ(м)-21

Латынцев А.В.

Проверил: преподаватель кафедры ПОВТАС

Тормозов В.С.

# Постановка задачи

**Цель работы**: научиться вычислять коэффициенты разделяющей линии ивеличину отступа (margin) при бинарной классификации объектов.

**Задания на лабораторную работу (8 вариант)**

1. Используя рисунок своего варианта, необходимо вычислить коэффициенты разделяющей линии, которая определяется выражением:
2. Вычислить отступы (margin) для зеленых точек (с меткой класса +1) и синих точек (с меткой класса -1). Напомню, что отступ вычисляется по формуле:

где - метка класса образа (точки) ; - скалярное произведение векторов.

Вектор должен быть подобран так, чтобы для наиболее удаленных точек от разделяющей линии отступ был положительным, а для ближних - отрицательным.

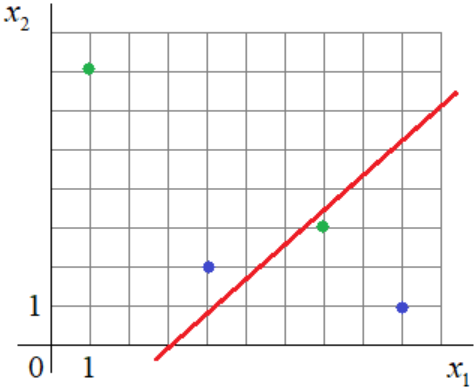


Рис. . График для 8 варианта

# Краткая теория

Прогноз линейной регрессионной модели определяется уравнением:

где – спрогнозированное значение, – количество признаков, – значение -того признака, – -тый параметр модели (включая член смещения и веса признаков ).

Обучение модели означает установку её параметров так, чтобы модель была наилучшим образом подогнана к обучающему набору. Для этой цели первым делом нужна мера того, насколько хорошо (или плохо) модель подогнана к обучающим данным. Самой распространенной мерой производительности регрессионной модели является квадратный корень из среднеквадратической ошибки (RMSE). Однако в данной лабораторной работе не стоит задача определения наилучших параметров, поэтому мы не будем вычислять RMSE.

Подбор гиперпараметров будет осуществляться с использованием ступенчатой функции Хевисайда, хорошо подходящей для задач бинарной классификации:

На Рис. 2 для модели хорошо видно, что радиус-векторы точек для класса образуют острый угол с вектором . Т.к. скалярное произведение – это, фактически, вычисление косинуса угла между векторами, то для острых углов будут получены положительные значения. Точки же класса образуют тупые углы с вектором . Следовательно, для будут получены отрицательные значения.

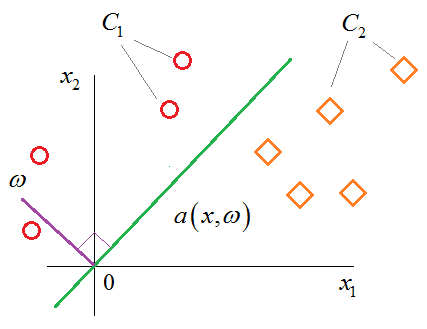


Рис. . Демонстрация принципа классификации функцией Хевисайда

В итоге, алгоритм классификации образов с помощью модели можно записать в виде:

Здесь – знаковая функция, которая возвращает +1 для положительных чисел и -1 – для отрицательных:

# Результаты работы

Работа была выполнена на языке программирования Python 3 с использованием Jupyter Notebook.

Первым делом (Листинг 1) импортировал необходимые библиотеки, определил точки в соответствии с вариантом, а также размер выборки:

|  |
| --- |
| **import** **numpy** **as** **np** # 1.23.4  **import** **ipywidgets** **as** **widgets** # 8.0.2  **import** **matplotlib.pyplot** **as** **plt** # 3.6.1  # использовать системное приложение для взаимодействия с графиками  # в моем случае - TkAgg  %matplotlib  points = {  'green': [[**1**, **7**], [**7**, **3**]],  'blue': [[**4**, **2**], [**9**, **1**]]  }  count\_of\_points = **0** # размер обучающей выборки  **for** point\_class **in** points:  count\_of\_points += len(points[point\_class]) |

Листинг 1. Первоначальная инициализация

Следующий шаг – определение функций Хевисайда и цикличный подбор оптимальных гиперпараметров:

|  |
| --- |
| w = [**0**, -**1**] # нач. значение вектора w  a = **lambda** x: np.sign(x[**0**]\*w[**0**] + x[**1**]\*w[**1**]) # решающее правило  N = **50** # максимальное число итераций  L = **0.1** # шаг изменения веса  e = -**0.05** # небольшая добавка w0 для зазора м/ду раздел. линией и классами  last\_err\_index = **None** # индекс последнего ошибоч. наблюдения  **for** n **in** range(N):  count\_of\_right\_answer = **0**  **for** point\_class **in** points:  multiplier = **1**  **if** point\_class == 'green':  multiplier = -**1**  **for** point **in** points[point\_class]:  **if** multiplier \* a(point) < **0**:  w[**0**] = w[**0**] + L \* multiplier  last\_err\_index = multiplier  **else**:  count\_of\_right\_answer += **1**  **if** count\_of\_right\_answer == count\_of\_points:  **break** |

Листинг 2. Расчет гиперпараметров

Листинг 3 определяет небольшое смещение для гиперпараметра , чтобы линия разделения не лежала точно на точке одного из классов:

|  |
| --- |
| w\_0 = **0** # смещение  **if** last\_err\_index:  w[**0**] += e \* last\_err\_index + w\_0 |

Листинг . Смещение для

В Листинг 4 определяются координаты разделяющей линии:

|  |
| --- |
| line\_x = [xy[**0**] **for** point\_class **in** points **for** xy **in** points[point\_class]]  line\_x\_coords = min(line\_x), max(line\_x)  line\_y\_coords = w[**0**]\*line\_x\_coords[**0**], w[**0**]\*line\_x\_coords[**1**] |

Листинг . Разделяющая линия

Ниже приведен код для вычисления отступов. Отступы характеризуют удаленность точки от линии разделения (длина перпендикуляра):

|  |
| --- |
| point\_margin = {  'green': [],  'blue': []  }  **for** point\_class **in** points:  multiplier = **1**  **if** point\_class == 'green':  multiplier = -**1**  **for** point **in** points[point\_class]:  point\_margin[point\_class].append(  round(multiplier \* (w[**0**] \* point[**0**] + w[**1**] \* point[**1**]), **2**)  ) |

Листинг . Отступы

Построение графика осуществляется с помощью matplotlib:

|  |
| --- |
| fig, ax = plt.subplots(nrows=**1**, ncols=**1**, figsize=(**8**, **8**))  ax.set\_title(print\_w, color='black')  ax.plot(line\_x\_coords, line\_y\_coords, color='red',  label=f'Разделяющая линия, {w[0]} \* x + {w[1]} \* y + {w\_0} = 0'  )  **for** point\_class **in** points:  ax.scatter(  list(map(**lambda** x: x[**0**], points[point\_class])),  list(map(**lambda** x: x[**1**], points[point\_class])),  color=point\_class,  label=f"Образы {point\_class}, Отступы: {point\_margin[point\_class]}"  )  ax.tick\_params(labelcolor='indigo')  ax.legend()  ax.grid()  plt.show() |

Листинг 6. Построение графика

Результат работы программы приведен на Рис. 3.

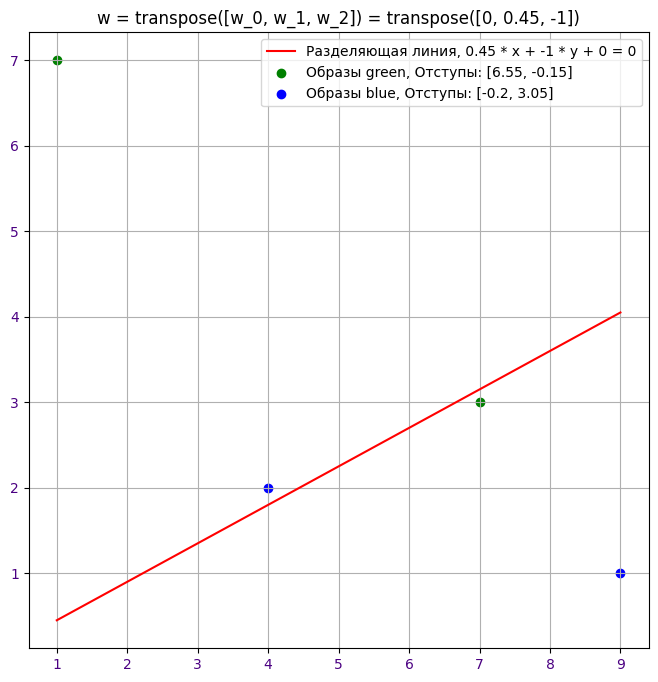


Рис. . Полученный график

# Вывод

В ходе лабораторной работы я определил коэффициенты разделяющей линии и отступы для зеленых и синих точек.