Московский Авиационный Институт

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №2 по курсу «Искусственный Интеллект»

Студент: В. П. Будникова

Преподаватель:

Группа: М80-307Б-19

Дата:

Оценка:

Подпись:

Москва, 2022

**Задание:**

1. Используя модели которые вы реализовали в предыдущей лабораторной работе, реализовать два подхода для построения ансамблей: жесткое и мягкое голосование

2. Реализовать дерево решений

4. Реализовать случайный лес

3. Воспользоваться готовой коробочной реализацией градиентного бустинга для решения вашей задачи

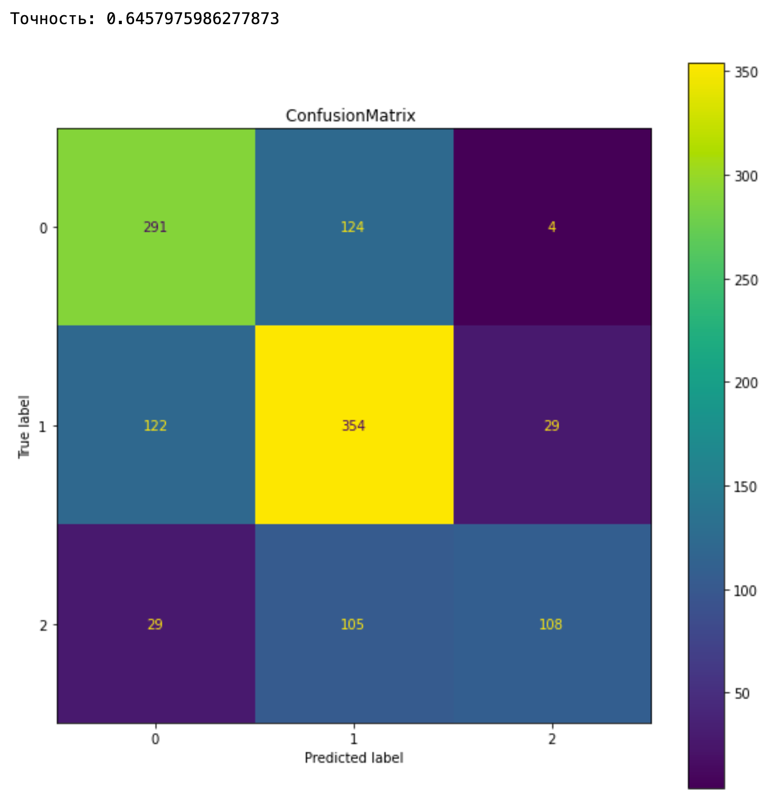
**Оборудование:**

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

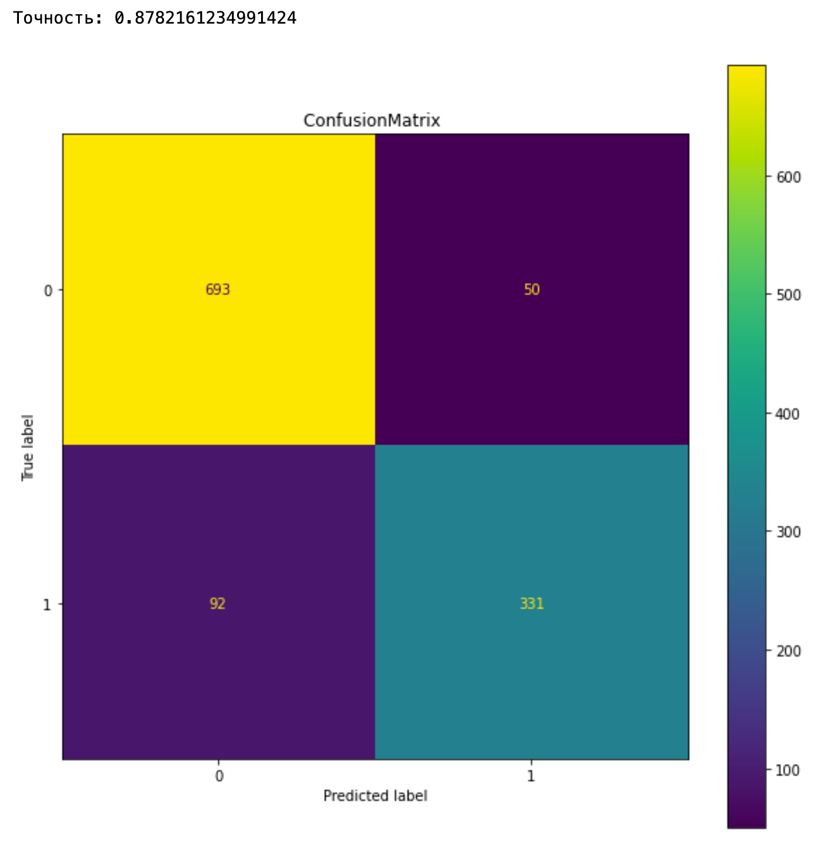
**Ход работы:**

Жесткое голосование

Для многоклассовой классификации:

****

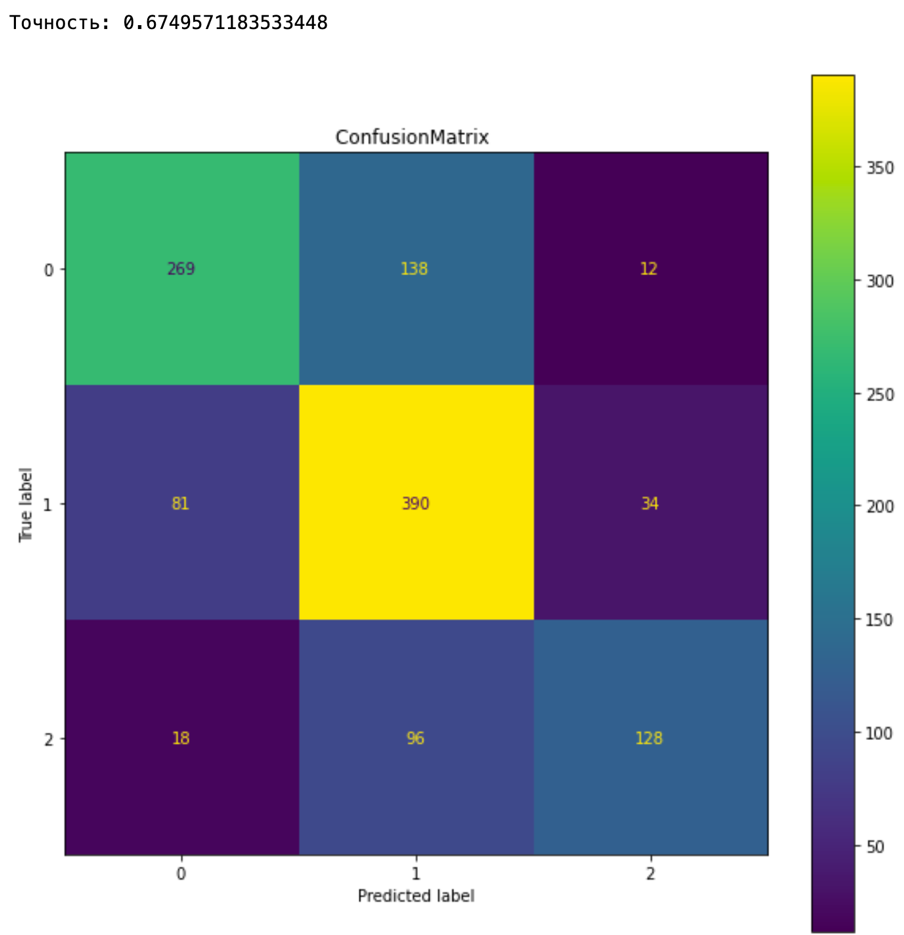
Для бинарной классификации:

****

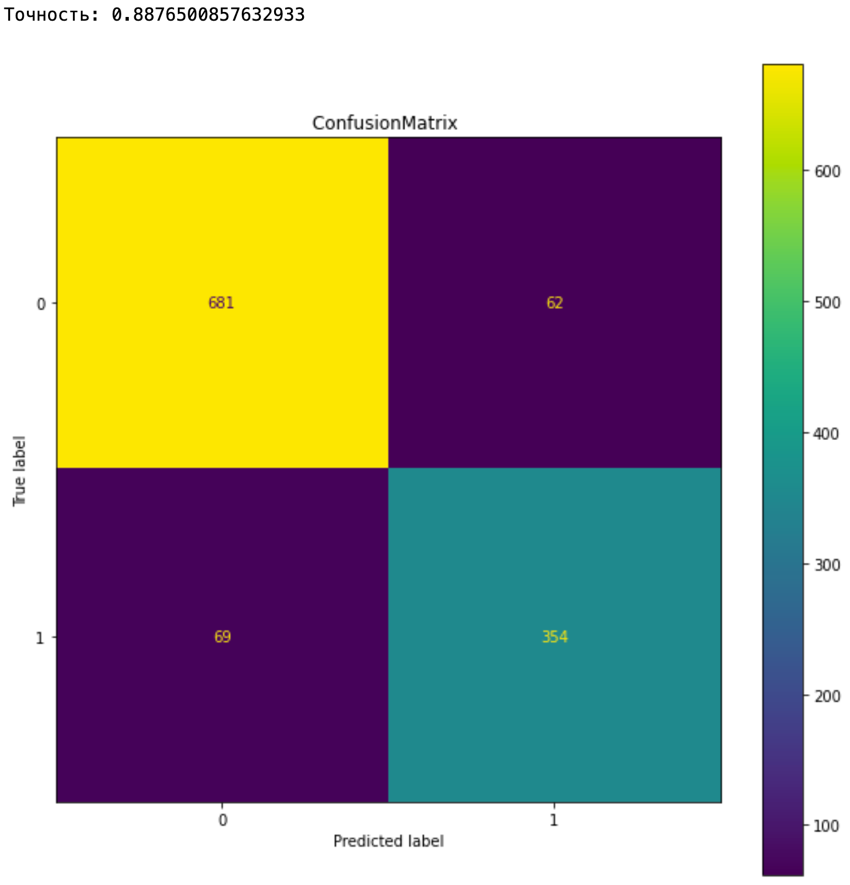
Анализируя точность и ConfusionMatrix, можно подтвердить выводы прошлой лабораторной работы, что на два «грубых» класса данные разделяются лучше, чем на три. Но алгоритм жесткого голосования дал точность выше, чем большинство алгоритмов, работавшие в одиночку при мультиклассовой классификации.

Мягкое голосование

Для многоклассовой классификации:

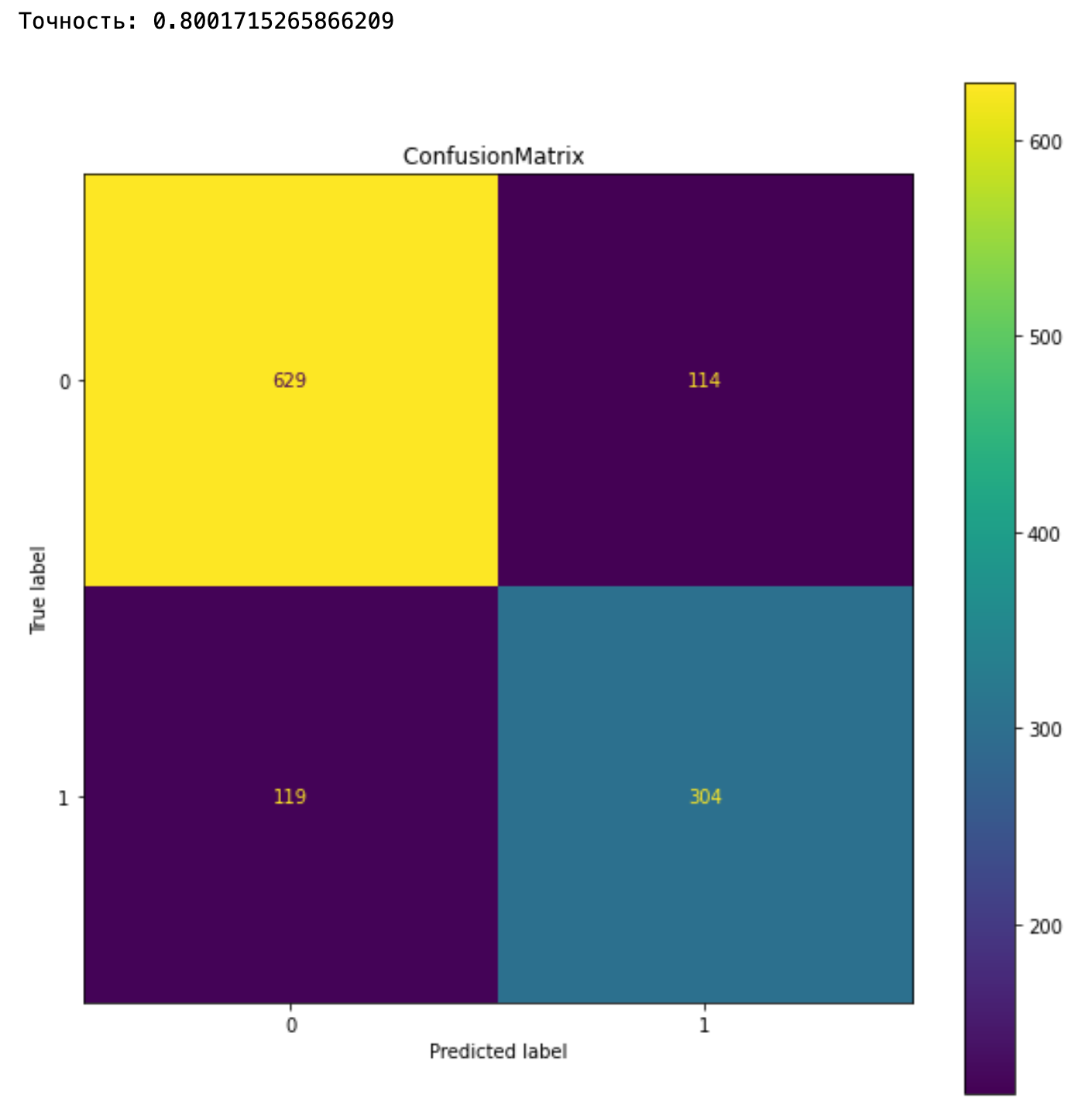


Для бинарной классификации:



При мягком голосовании были получены точности выше, чем при жестком голосовании. При мягком голосовании, учитываются не сами значения, выданные алгоритмами, участвующими в голосовании, а оцениваются вероятности, полученные для каждого класса в каждом алгоритме, что позволяет не получать грубых ошибок.

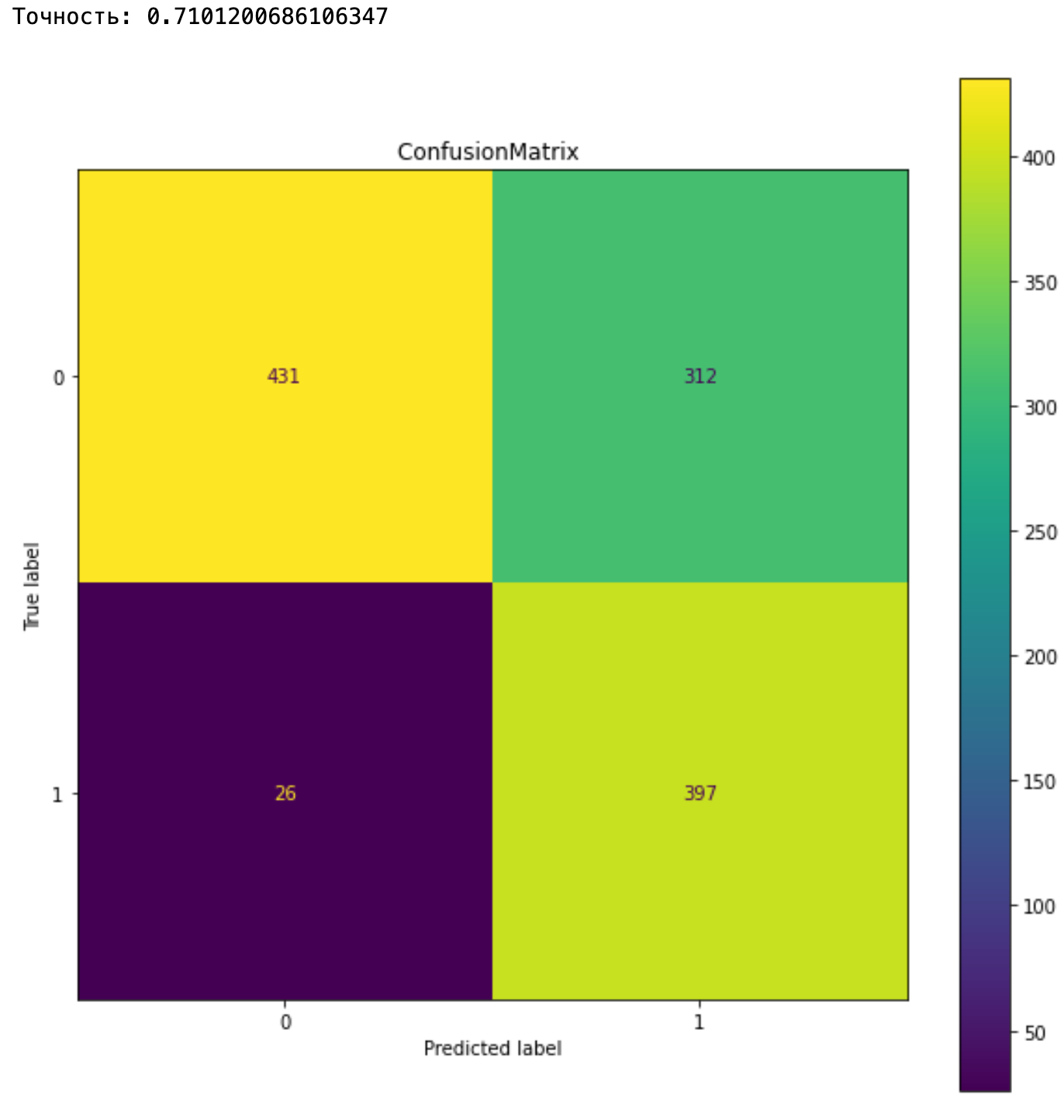
Дерево решений



Дерево решений дало хорошую точность – 0.8

При реализации и тестировании были проверены разные параметры глубины дерева, если дерево сделать слишком глубоким, то оно начинает переобучаться, так как подстраивается под тренировочную выборку. Если глубину дерева совсем не ограничивать, то получится так, что для почти каждого варианта тренировочной выборки будет существовать свой путь в дереве и свой лист, что плохо скажется при тестировании на отличающихся тестовых данных.

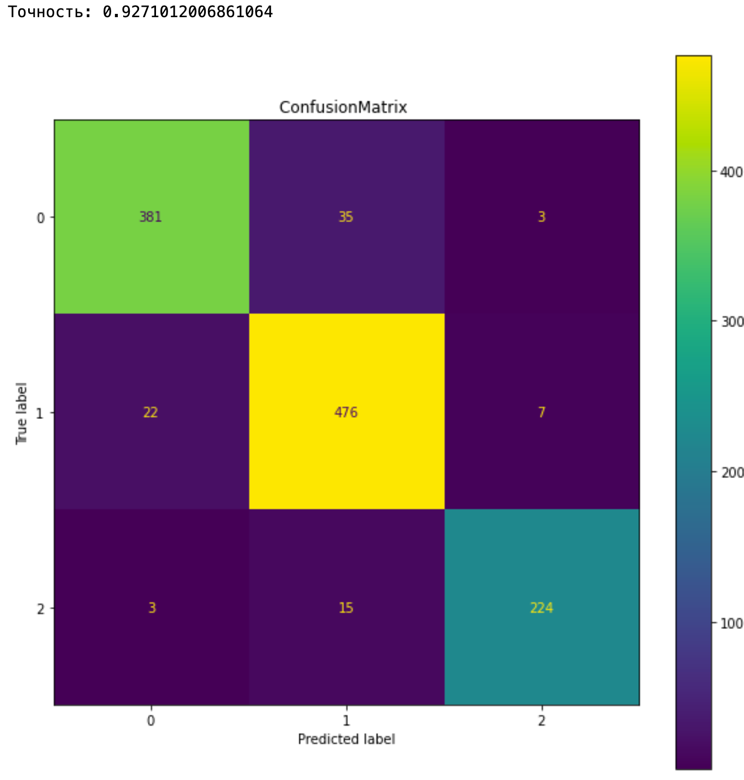
Случайный лес



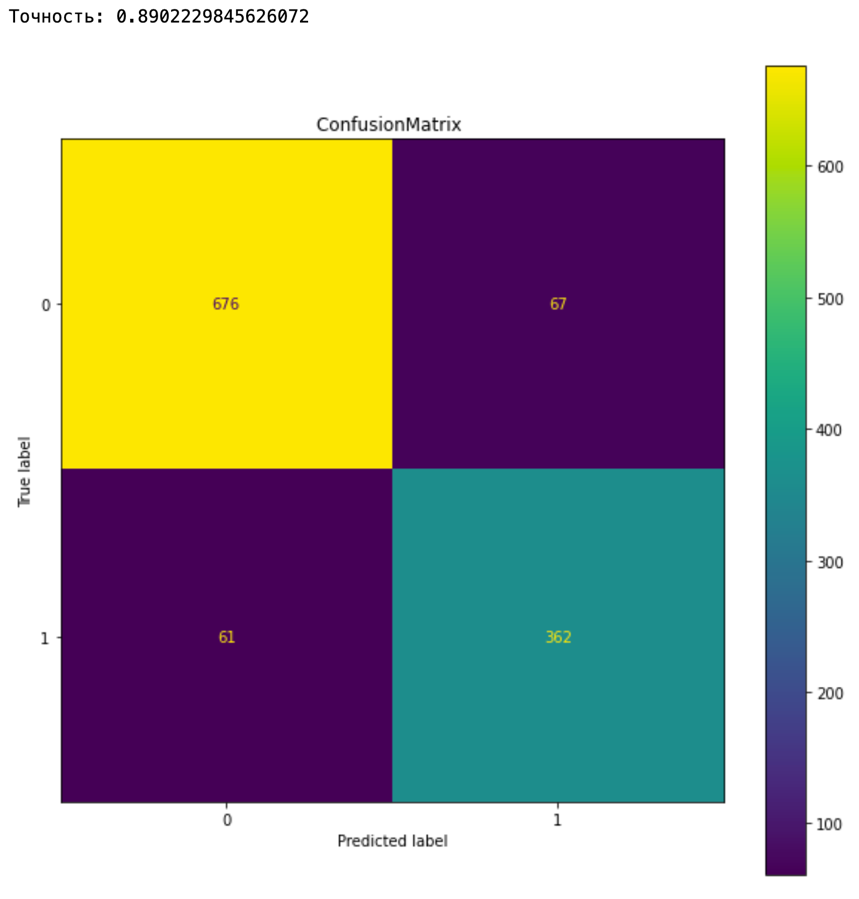
Случайный лес показал результат хуже, чем дерево решений. При тестировании случайного леса точность алгоритма могла как быть и очень близко к точности одного случайного дерева, так и мыть меньше нее. По моему мнению это происходит потому, что при построении деревьев используются случайные выборки из датасета, а также случайные индексы, по которым элементы будут сравниваться для разделения, что может приводить к понижению точности. Но данный алгоритм будет хорошо себя проявлять при наличии зашумленных данных, а также ему будет сложнее переобучится, чем одному дереву решений, даже при максимальной глубине каждого дерева.

Градиентный бустинг

Для многоклассовой классификации:



Для бинарной классификации:



Коробочный градиентный бустинг показал наилучший результат для данного датасета в данной лабораторной работе

**Вывод:**

В данной лабораторной работе я изучила разные ансамбли: жесткое и мягкое голосование, а также случайный лес. Я реализовала данные алгоритмы, а также алгоритм дерева решений для моего датасета. Также была использована коробочная реализация градиентного бустинга и сравнены результаты каждого из алгоритмов

Ссылка на GitHub: https://github.com/Ler-B/Al