Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»
Дисциплина «Искусственный Интеллект»

Лабораторная работа №2

Тема: Ансамбли и деревья решений

Студент: Глушатов И.С.
Группа: М8О-307Б-19
Преподаватель: Ахмед Самир. Х.
Дата:
Оценка:

Москва, 2022

Цель работы: научиться реализовывать дерево решений, различные типы ансамблей (бэггинг, пастинг, бустинг, стэкинг), случайный лес. Провести оценку по выбранному датасету.

Задание:

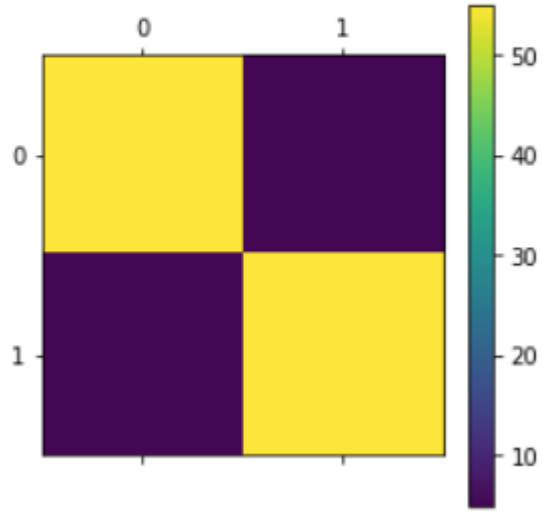
Вы построили базовые (слабые) модели машинного обучения под вашу задачу. Некоторые задачи показали себя не очень, некоторые показали себя хорошо. Как выяснилось, вашим инвесторам показалось этого мало, и они хотят, чтобы вы построили модели посерьезней и поточнее. Вы вспомнили, что когда-то вы проходили курс машинного обучения и слышали что есть способ улучшить результаты вашей задачи: ансамбли: беггинг, пастинг, бустинг и стекинг, а также классификация путем жесткого и мягкого голосования и вы решили это опробовать.

Ход работы

На основе уже реализованных в прошлой лабораторной работы простых линейных моделей бинарной классификации, нужно было использовать их для построение ансамблей различных типов.

Я начал с самого простого типа ансамбля – стэкинга, суть которого заключалась в классификации мнением большинства из различных простых алгоритмов классификации.

```
Accuracy: 0.9166666666666666
Recall: 0.9166666666666666
Precision: 0.9166666666666666
```

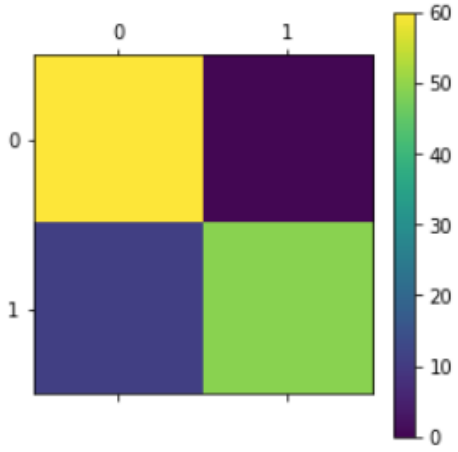


Слева представлены метрики по данному типу ансамбля для алгоритмов KNN с 3 и 6 соседями и логистической регрессии.

Далее я реализовал ансамбль типа бэггинга, который в свою очередь разделяется на два алгоритма – когда из тренировочной выборки данные берутся единожды (пастинг) и когда любое количество раз.

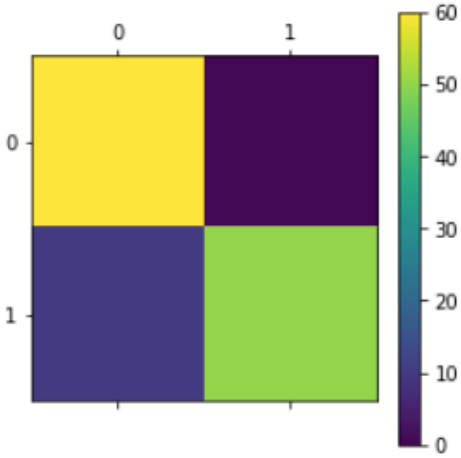
Пастинг

```
Accuracy: 0.9083333333333333
Recall: 0.9083333333333333
Precision: 0.9225352112676056
```



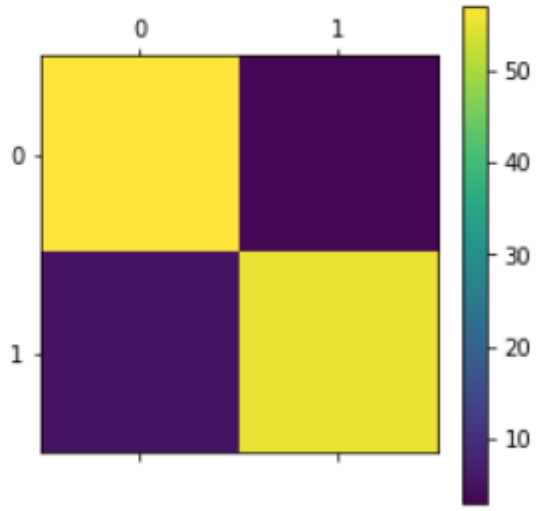
Бэггинг

```
Accuracy: 0.9166666666666666
Recall: 0.9166666666666667
Precision: 0.9285714285714286
```



За базовый оценщик я брал логистическую регрессию, количество оценщиков – 11, а максимальное число данных в подвыборке – 10. В среднем результаты этих ансамблей были чуть хуже, чем в стэкинге, однако при удачном подборе параметров можно довести accuracy до 92,5%.

```
Accuracy: 0.9333333333333333
Recall: 0.9333333333333333
Precision: 0.9338153503893214
```

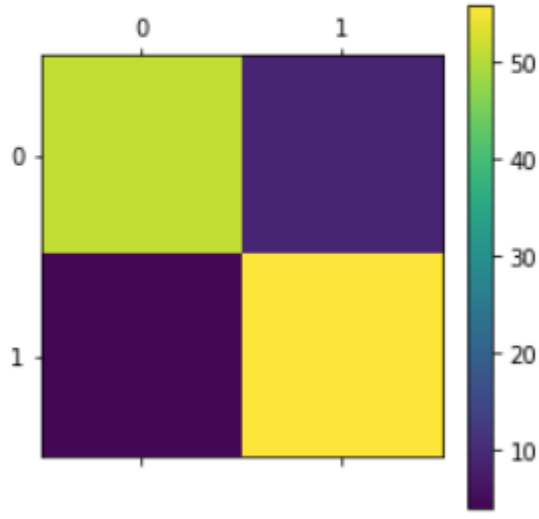


Следующим ансамблем на очереди должен был быть бустинг, однако в связи со сложностью понимания реализации, я не смог его сделать. Я взял алгоритм градиентного бустинга из библиотеки sklearn. Слева его метрики при 11 оценщиках, однако при 100 оценщиках accuracy доходит до 93,3%.

Следующей частью стала реализация жесткого и мягкого голосования.

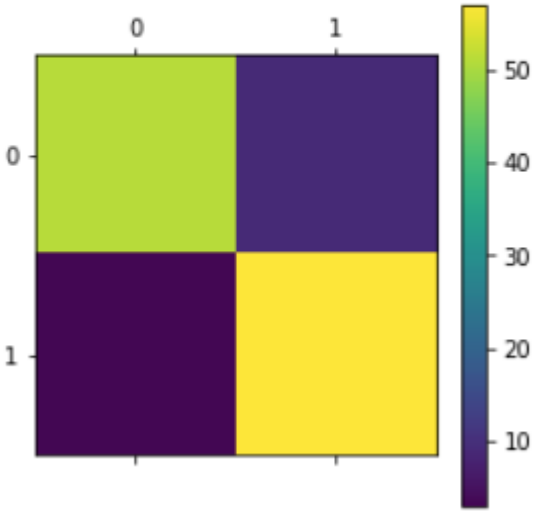
Жесткое

```
Accuracy: 0.8916666666666667
Recall: 0.8916666666666666
Precision: 0.8944055944055944
```



Мягкое

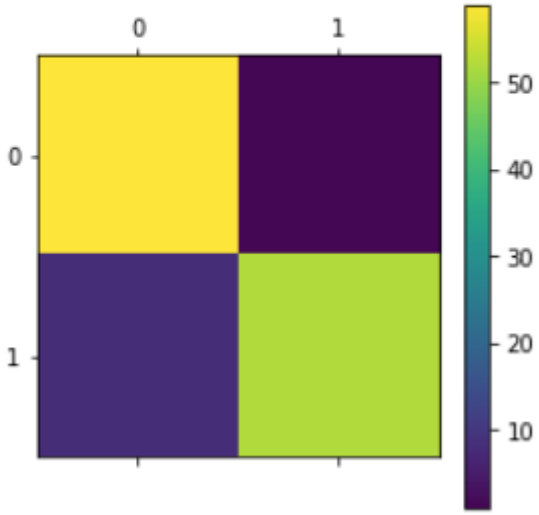
```
Accuracy: 0.9
Recall: 0.8999999999999999
Precision: 0.904040404040404
```



Классификаторами выступали все представители простых классификаторов, среди которых было два KNN с 3 и 4 соседями. В среднем мягкое голосование показывало результаты несколько лучше, чем жесткое.

Далее требовалось реализовать решающее дерево. Это был самый сложный алгоритм классификации из всех. В качестве критерия разделения я использовал критерий Джини.

```
Accuracy: 0.925
Recall: 0.925
Precision: 0.9308645451985356
```

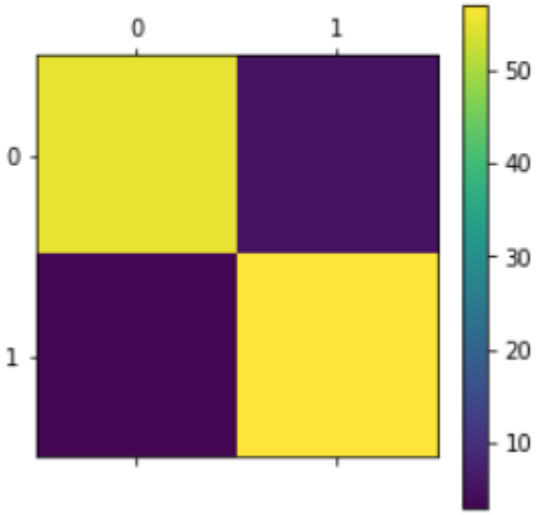


Слева представлены его метрики. Это оказался лучший классификатор из раннее реализованных. Есть очень важное отличие от встроенной реализации из sklearn. Вместо параметра глубины дерева, у меня задается уровень значимости, который определяет, что листом будет то разбиение, в котором доля первого класса либо меньше уровня значимости, либо больше единицы минус уровень значимости, т. е. в разбиении количество представителей определенного класса близко либо к нулю, либо к единице.

С помощью ансамбля бэггинга с базовым оценщиком – решающим деревом, получаем случайный лес.

После удачного подбора параметров получилось довести accuracy до результатов градиентного бустинга.

```
Accuracy: 0.9333333333333333
Recall: 0.9333333333333333
Precision: 0.9338153503893214
```



Выводы

В ходе лабораторной работы я реализовал различные типы ансамблей, решающее дерево и случайный лес. В целом все алгоритмы неплохо классифицируют выборку, однако на моем датасете не дают принципиального выигрыша. В основном это связано с малым размером датасета и хорошей разделимости данных. Однако все равно можно было заметить, что решающее дерево и в особенности случайный лес давали очень хорошие результаты. В будущем я попытаюсь адаптировать решающее дерево для решения многоклассовой классификации и протестировать его на данном датасете. GitHub.