**Градиентный бустинг в «лоб»**

1. Пропуски среди значений имеют следующие признаки:

* first\_blood\_time
* first\_blood\_team
* first\_blood\_player1
* first\_blood\_player2
* radiant\_bottle\_time
* radiant\_courier\_time
* radiant\_flying\_courier\_time
* radiant\_first\_ward\_time
* dire\_bottle\_time
* dire\_courier\_time
* dire\_flying\_courier\_time
* dire\_first\_ward\_time

Пропуски в признаках, связанных с «первой кровью» (first\_blood\_time, first\_blood\_player1), образуются в случае, если событие «первая кровь» не произошло в первые 5 минут матча. Пропуски в признаках, связанных с приобретением предметов (radiant\_courier\_time, radiant\_bottle\_time), образуются в случае, если эти предметы не были приобретены в первые 5 минут матча.

1. Т.к. в результате необходимо предсказать победителя матча, то целевой переменной является radiant\_win.
2. Время кросс-валидации для 30 деревьев составляет 1 минуту 17 секунд. При этом среднее качество работы алгоритма составляет 68,96%.
3. На рис. 1 представлен график зависимости точности работы алгоритма (синий график, %) и времени кросс-валидации (оранжевый график, с) от числа деревьев в классификаторе.

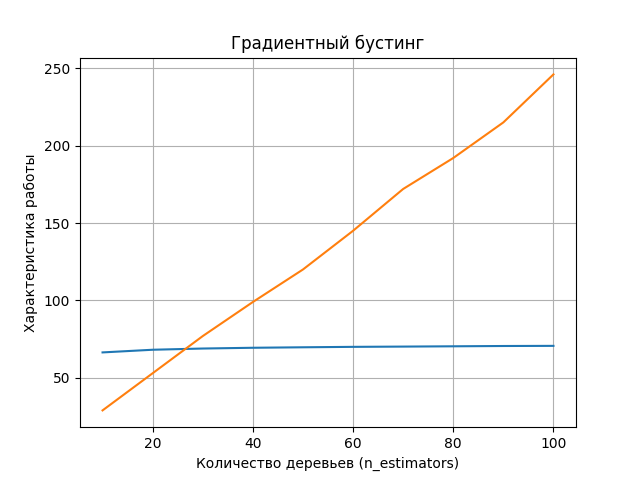


Рисунок 1 – Зависимость характеристик работы алгоритма от числа деревьев

Из рис. 1 видно, что время работы кросс-валидации с увеличением числа деревьев растет гораздо быстрее, чем качество работы. Следовательно, можно сделать вывод, что увеличение числа деревьев хоть и приведет к увеличению качества классификации, но, ввиду вычислительных затрат, увеличение числа деревьев нецелесообразно. Поэтому нет смысла использовать больше 30 деревьев в градиентном бустинге.

Максимальное качество работы алгоритма на рассмотренном диапазоне параметров было достигнуто при n\_estimators = 100, max\_depth = 5, и составляет 70,97%.

Процесс обучения классификатора можно ускорить, выявив ключевые признаки, влияющие на результат работы алгоритма классификации, и использовать только первые n информативных признаков. За счет сокращения количества признаков сократится количество вычислений, что приведет к ускорению работы алгоритма кросс-валидации. Наиболее значимые признаки данной модели представлены ниже и отсортированы по убыванию значимости:

featureImportances = pd.DataFrame(data=clf.feature\_importances\_)

d1\_gold=4.25

d2\_gold=4.21

d4\_gold=4.09

r1\_gold=4.07

r3\_gold=3.84

r4\_gold=3.75

r2\_gold=3.74

d5\_gold=3.68

r5\_gold=3.35

d3\_gold=3.14

r1\_lh=2.23

d3\_lh=2.2

d4\_lh=2.16

r5\_lh=2.07

Кроме сокращения числа признаков для ускорения работы алгоритма можно прибегнуть к уменьшению глубины решающих деревьев.

**Логистическая регрессия**

1. Наилучшее качество работы алгоритма было достигнуто при параметре регуляризации С, равном 1e-06. Наилучшее качество при этом составляет 51,35%. Полученный результат значительно ниже, чем при градиентном бустинге, ввиду того, что в обучающей выборке присутствуют категориальные признаки, которые обрабатываются как числовые. Кроме того, значения признаков не масштабированы. Логистическая регрессия работает быстрее бустинга (кросс-валидация по 5 блокам с 15 возможными значениями параметра C занимает порядка 1 минуты).
2. При удалении категориальных признаков параметр C, при котором достигается наилучшее качество, не изменился, как не изменилось и наилучшее качество модели. Объясняется это тем, что, во-первых, как мы выяснили в предыдущем задании, удаленные признаки не являются наиболее информативными, и во-вторых, линейные модели не очень хорошо работают с категориальными признаками.

При масштабировании числовых признаков оптимальный параметр C стал равен ~0.01, качество модели при этом улучшилось до 71,65%. Связано это с тем, что при масштабировании значения признаков выравниваются, исчезают резкие отклонения.

1. В игре 108 различных идентификаторов героев.