Подход 1: градиентный бустинг «в лоб»

1. *Какие признаки имеют пропуски среди своих значений (приведите полный список имен этих признаков)? Что могут означать пропуски в этих признаках (ответьте на этот вопрос для двух любых признаков)?*

   Столбцы, имеющие пропуски:

   first_blood_player2 - 43987

   radiant_flying_courier_time - 27479

   dire_flying_courier_time - 26098

   first_blood_time - 19553

   first_blood_team - 19553

   first_blood_player1 - 19553

   dire_bottle_time - 16143

   radiant_bottle_time - 15691

   radiant_first_ward_time - 1836

   dire_first_ward_time - 1826

   radiant_courier_time - 692

   dire_courier_time – 676

   Каждое событие либо произошло после 5 минут матча, либо гипотетически могло не произойти вовсе (например, покупка ботлов, вардов, или даже курьера). По данным можно сделать вывод, что часто первая кровь - заслуга лишь одного игрока, и довольно часто апгрейд курьера происходит после 5 минуты (либо не происходит вовсе). Однако все пропуски довольно разумны и их частота обоснована (личное мнение)

2. *Как называется столбец, содержащий целевую переменную?*

   Целевую переменную содержит столбец `radiant_win`

3. *Как долго проводилась кросс-валидация для градиентного бустинга с 30 деревьями? Инструкцию по измерению времени можно найти выше по тексту. Какое качество при этом получилось?*

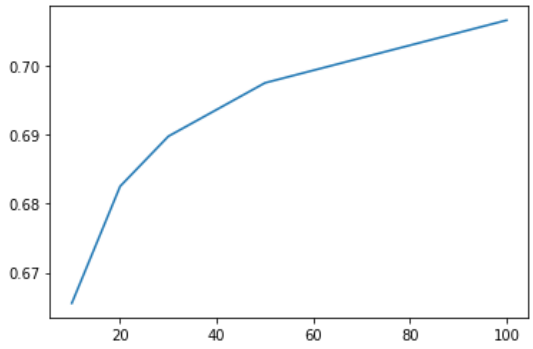
   Время для 10 деревьев: 0:00:31.954167

   Время для 20 деревьев: 0:00:51.895380

   Время для 30 деревьев: 0:01:18.672152

   Время для 50 деревьев: 0:02:10.720546

   Время для 100 деревьев: 0:04:21.051333



4. *Имеет ли смысл использовать больше 30 деревьев в градиентном бустинге? Что можно сделать, чтобы ускорить его обучение при увеличении количества деревьев*?

   График выше имеет отрицательную выпуклость, это означает, что чем больше деревьев используется, тем меньше пользы приносят новые деревья, при этом время выполнения линейно зависит от количества деревьев. Однако, график всё-таки возрастает. Поэтому, сли позволяют ресурсы, то имеет смысл использовать больше 30 деревьев. Для ускорения обучения можно использовать только часть выборки или уменьшить глубину деревьев.

Подход 2: логистическая регрессия

1. *Какое качество получилось у логистической регрессии над всеми исходными признаками? Как оно соотносится с качеством градиентного бустинга? Чем можно объяснить эту разницу? Быстрее ли работает логистическая регрессия по сравнению с градиентным бустингом?*

   Значение 0.7164022547773058

   Показатель выше самого лучшего показателя градиентного бустинга. При этом время выполнения намного меньше, это позволяет не идти на компромиссы при обучении модели. Тот факт, что логистическая регрессия работает хорошо, означает, что зависимость целевой переменной от признаков довольно линейна, поэтому линейные методы классификации хорошо будут работать для описания данной связи.

2. *Как влияет на качество логистической регрессии удаление категориальных признаков (укажите новое значение метрики качества)? Чем можно объяснить это изменение?*

   Значение 0.716510509876658

   Качество практически не изменилось. Это вовсе не означает что удалённые признаки были не важны, так как они и до этого являлись шумом. Это означает, что модель оказалась довольна устойчива к целым 11 "шумовым" признакам. По сути, это логично: результат, скорее всего, никак не коррелирует с числовым значением удалённых признаков, поэтому в обученной модели их вес, скорее всего, близок к нулю

3. *Сколько различных идентификаторов героев существует в данной игре?*

   Героев в выборке - 108. Потенциально героев - 112+

4. *Какое получилось качество при добавлении "мешка слов" по героям? Улучшилось ли оно по сравнению с предыдущим вариантом? Чем можно это объяснить?*

   Значение 0.7518733702338259

   Качество значительно улучшилось, что очевидно. До этого "мусорные" признаки не предоставляли никакой информации, а сейчас были добавлены информативные признаки в виде используемых героев. Действительно, различные герои имеют различный винрейт, поэтому более сильные герои имеют больше шансов на победу.

5. *Какое минимальное и максимальное значение прогноза на тестовой выборке получилось у лучшего из алгоритмов?*

   0.008543874997718807 0.9963968730626978 соответственно