Решение задачи предсказания победителя в соревновательной онлайн-игре

Вадим А. Порватов

Декабрь 2020

Содержание

1 Постановка задачи

Dota 2 — многопользовательская компьютерная игра жанра MOBA. Игроки играют между собой матчи. В каждом матче, как правило, участвует 10 человек. Матчи формируются из живой очереди, с учётом уровня игры всех игроков. Перед началом игры игроки автоматически разделяются на две команды по пять человек. Одна команда играет за светлую сторону (The Radiant), другая — за тёмную (The Dire). Цель каждой команды — уничтожить главное здание базы противника, трон.

Задачей является построение модели, которая по данным о первых пяти минутах матча будет предсказывать его исход — то есть определять команду-победителя.

2 Исходные данные

К заданию приложены следующие файлы:

1. final-statement.ipynb и final-statement.html — постановка задачи, описание данных, инструкции по выполнению
2. features.zip — архив с обучающей выборкой
3. featuresTest.zip — архив с тестовой выборкой
4. data.zip — полный архив с сырыми данными и скриптом для извлечения признаков (этот архив понадобится вам только для участия в kaggle; для выполнения данного задания он не нужен)
5. extractFeatures.py — скрипт, извлекающий признаки из сырых данных

3 Review criteria

Необходимо провести описанные в документе final-statement.html (или final-statement.ipynb) два этапа исследования (для двух подходов к решению задачи), написать по результатам каждого этапа небольшой отчет (ниже указаны вопросы, ответы на которые должны содержаться в отчете), и предоставить для ревью данный отчет и код, с помощью которого вы выполнили задание.

Отмечается, что в выборке есть признаки, которые "заглядывают в будущее"— они помечены в описании данных как отсутствующие в тестовой выборке. Их прямое использование в модели приведет к переобучению, поэтому не забудьте исключить их из выборки.

4 Градиентный бустинг

Градиентный бустинг является универсальным алгоритмом - он не очень требователен к данным, восстанавливает нелинейные зависимости, и хорошо работает на многих наборах данных, что и обуславливает его популярность.

Результаты касаемо производительности отмечены в соответствующем разделе приложенного .ipynb файла.

5 Логистическая регрессия

Линейные методы работают гораздо быстрее композиций деревьев, поэтому кажется разумным воспользоваться именно ими для ускорение анализа данных. Одним из наиболее распространенных методов для классификации является логистическая регрессия. В данном разделе предлагается применить ее к данным, а также попробовать различные манипуляции с признаками.

Результаты касаемо производительности отмечены в соответствующем разделе приложенного .ipynb файла.

6 Выводы

Логистическая регрессия с мешком слов и конвертацией категориальных признаков в числовые позволила на порядок ускорить скорость обучения модели в сравнении с другими конфигурациями этой модели и градиентным бустингом. Кроме того была достигнута существенно более высокая точность классификации на представленном датасете.