Optimal prediction of NBA MVP award.

Vladimir Belonin
Optimization Class Project. MIPT

Ввдение

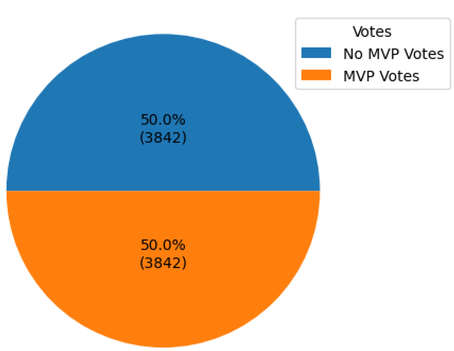
Если вы хотя бы когда-то играли в баскетбол, то наверняка слышали о таких игроках, как Майк Джордан, Леброн Джеймс. Главная лига всего мира - НБА (Национальная Баскетбольная Ассоциация). Самой желанной наградой любого баскетболиста является победа в этом чемпионате. Но это не единственная награда, которая присуждается игрокам раз в сезон. Также есть титул лучшего новичка года, лучшего тренега года, лучший оборонительный игрок и так далее. Но я считаю самой значимой по статусу наградой после победы в чемпионате - приз самого ценного игрока сезона (далее MVP). Эта награда вручается сразу после окончания регулярного сезона, победитель выбирается голосованием, в котором участвует группа спортивных обозревателей и телеведущих из США и Канады. Каждый член комиссии для голосования голосует за выборы с первого по пятое. Каждый голос за первое место приносит 10 очков; каждый голос за второе место стоит семь; каждый голос за третье место стоит пять, за четвертое место - три, а за пятое - одно. Начиная с 2010 года, один бюллетень отдан фанатам посредством онлайн-голосования. Игрок, набравший наибольшее количество очков, получается награду MVP.

Задача

Наша задача состоит в том, чтобы предсказать, какую часть общего количества очков набрал игрок. С этой точки зрения, я буду использовать регрессионную модель с учителем. Буду использовать RandomForestModel. О ней будет рассказано позже.

Данные

Подробная статистика НБА за все сезоны присутствует на сайте [1] Basketball Reference. Я использую датасет, собранный с этого сайта, который содержит 4 типа статистики - статистика регулярного сезона, расширенная статистика сезона, статистика команды и статистика голосования за MVP.
Эти данные несбалансированные, так как количество игроков, получивших голоса MVP, гораздо меньше, чем количество игроков, эти голоса не получивших.
Для начала я убрал параметры статистики, которые являются производными других параметров. Например, количество очков за матч это произведение количество попыток на процент реализации. Еще я отсек игроков, которые мало выходили в стартовой патерке, которые забивали мало очков, которые имели мало игрового времени. Добавил синтетические данные. После этих манипуляций, данные стали сбалансированными.

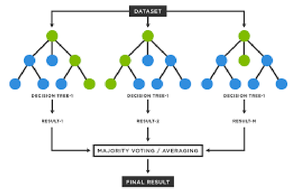


Признаки

Не вся статистика имеет одинаковую ценность в борьбе за эту награду. Например, среднее количество результативных передач является гораздо менее ценным признаком, чем среднее число очков за игру. Чтобы выбрать 'полезную' статистику использовался метод Боруты [2]. Это алгоритм, который может автоматически сделать выбор признаков в датасете. В его основании лежат две идеи - теневые признаки и биномиальное распределение.

Модель и ее гиперпараметры

В проекте использовалась модель RandomForest. Это расширение алгоритма решающих деревьев, который использует ансамбль деревьев для улучшения качества классификации и регрессии. Суть алгоритма заключается в том, что он создает множество решающих деревьев и использует их для предсказания классов объектов. Каждое дерево строится на случайном подмножестве обучающих данных и случайном подмножестве признаков. В результате, каждое дерево в ансамбле получается немного разным, что позволяет уменьшить эффект переобучения и повысить качество предсказаний.



Гиперпараметры, которые я исследую, следующие - n_estimators, max_depth, min_split, min_leaf.

Оптимизаторы

Для оптимизации гиперпараметров модели я использовал такой инструмент, как Optuna. Для выборки параметров я использовал следующие сэмплеры - TPESampler, GridSampler и RandomSampler [3]. Множества возможных значений гиперпараметров были выбраны эмпирически.

```
n_est = trial.suggest_int('n_estimators', 2, 20)
max_depth = trial.suggest_int('max_depth', 1, 32, log=True)
min_split = trial.suggest_int('min_split', 2, 10)
min_leaf = trial.suggest_int('min_leaf', 1, 10)
```

Метрики качества

Для оценки качества модели была выбрана метрика Mean Absolute Error. Приведу код, который показывает, как считается эта метрика.

```
mae = np.mean(np.absolute(valPred - valTarFold.to_numpy()[:,0]))
```

где varPred - предсказанные данные, varTarFold - тестовые данные.
Для оценки качества работы оптимизатора использовался метод .mean(), который возвращает среднее значение.
Каждая модель запускалась на каждом сезоне и % - отношение удачный прогнозов MVP ко общему числу попыток (количество сезонов, когда эта награда существует).

Результаты

	Без Optuna	TSEsampler	Gridsampler	Randomsampler
n_est	-	17	12	13
max_depth	7	31	21	19
min_split	-	2	4	2
min_leaf	-	1	1	1
mae	0.011	0.008	0.008	0.009
%	72.5	75	80	82.5

В этой таблице представленны результаты вычислений. Первые 4 строчки - значения оптимальных гиперпараметров. mae - качество модели. Каждая модель запускалась на каждом сезоне и % - отношение удачный прогнозов MVP ко общему числу попыток (количество сезонов, когда эта награда существует).

Conclusion

Как мы можем увидеть из таблицы, наименьшая MAE у TSEsampler и Gridsampler. Но общий процент правильный предсказаний значительно меньше, чем у Randomsampler. Разница MAE у этих моделей несущественная, поэтому оптимальная модель для предсказания MVP является

```
model = Random Forest Regression
Hyperparametrs = {
'n_estimators' = 13,
'max_depth' = 19,
'min_split' = 2,
'min_leaf' = 1
}
```

Список литературы

[1] Nba statistics: https://www.basketball-reference.com/.
[2] The boruta method: https://towardsdatascience.com/boruta-explained-the-way-i-wish-someone-explained-it-to-me-4489d70e154a.
[3] Optuna.samplers documentation: https://optuna.readthedocs.io/en/stable/reference/sam