Предсказательные модели для игроков и команд EPL

Годовой проект - промежуточная защита

Команда проекта

Студенты

Чуприн Александр – @Chew13

Лапшин Никита – @GodSiemens

Дубов Владислав – @dubov_vv

Куратор

Горяной Егор - @nogaromo

Описание задачи

В рамках проекта по разработке предсказательных моделей для игроков и команд EPL(Английская Премьер-лига) мы решаем следующие задачи:

- написание парсера для сбора статистики по игрокам и матчам с информационных сайтов
- проведение EDA
- построение предсказательных ML-моделей
- построение инфраструктуры и сервисов вокруг ML модели

Сбор данных

Разработаны парсеры для сбора данных из следующих источников:

- Официальный сайт Английской Премьер-лиги
- Альтернативный источник, сайт со статистикой(Fbref.com)

Были собраны признаки в количестве ~ 800 штук

Собраны данные за 10 последних сезонов

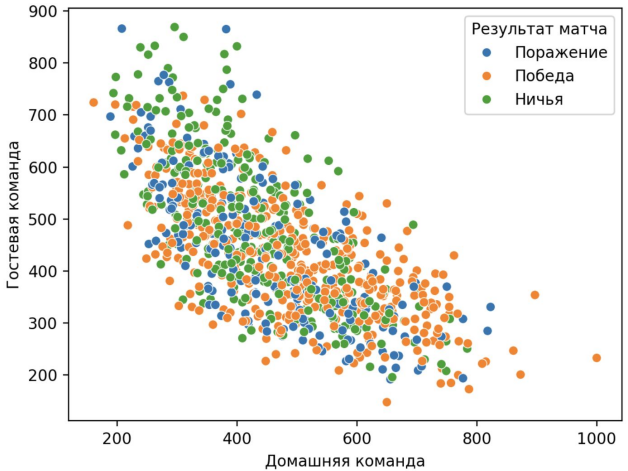
Исследовательский анализ данных

Был проведен исследовательский анализ данных в jupyter ноутбуках и развернут streamlit с динамическим EDA

Диаграмма рассеяния

Выберите признак для построения диаграммы

Передач



Топ 10 игроков

Выберите признак, по которому подбирать топ

- количество побед
- процент побед
- победные голы

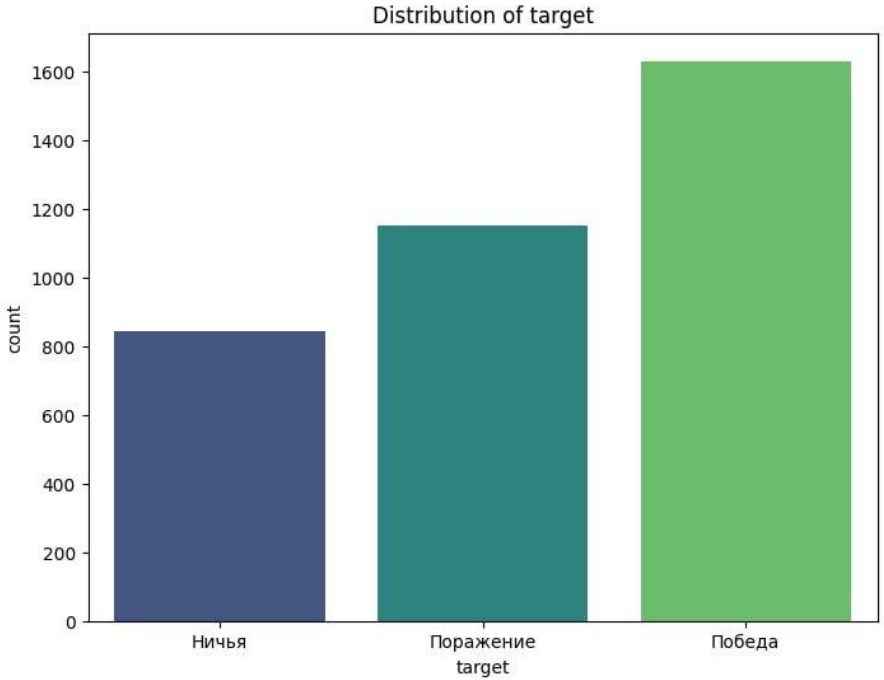
Топ 10 игроков за сезоны (2021, 2023) по признаку: количество побед

Выберите интересующие вас признаки

Имя игрока × Побед × Поражений × Ничьих × Игр × Процент побед ×

	Имя игрока	Побед	Поражений	Ничьих	Игр	Процент побед
157	Ederson	61	10	10	81	2.2774
500	Rodri	61	6	10	77	2.5379
65	Bernardo Silva	58	7	11	76	2.3639
77	Bukayo Saka	54	19	12	85	1.9298
191	Gabriel Magalhães	54	16	11	81	2.0628
394	Martin Ødegaard	53	17	12	82	1.9535
21	Alisson Becker	53	12	18	83	1.9635
476	Phil Foden	52	8	10	70	2.2688
62	Ben White	52	16	12	80	1.978
514	Rúben Dias	52	6	6	64	2.417

Распределение целевой переменной



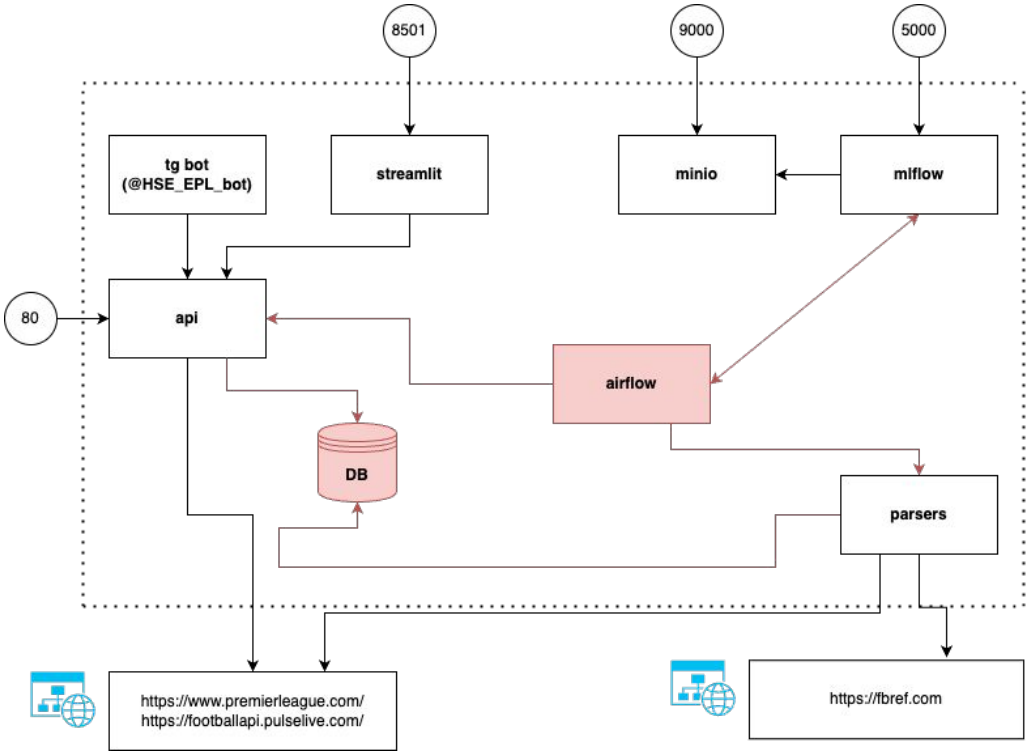
Разработка сервисов

Разработаны следующие сервисы:

- приложение fastapi
- телеграм бот
- streamlit приложение

Сервисы развернуты на VPS с помощью docker-compose.

Схема микросервисного взаимодействия



Телеграм бот @HSE_EPL_bot

Владислав Дубов
/ten_predict

E

EPL Fit Predict

Home	Away	Predict	Proba
Bournemouth	Luton Town	1	0.38
Arsenal	Crystal Palace	1	0.47
Brentford	Nottingham Forest	1	0.41
Sheffield United	West Ham United	0	0.44
Bournemouth	Liverpool	0	0.41
Brighton & Hove Albion	Wolverhampton Wanderers	0	0.38
Nottingham Forest	Arsenal	0	0.40
Fulham	Everton	1	0.38
Luton Town	Brighton & Hove Albion	0	0.36
Crystal Palace	Sheffield United	1	0.46

Разработка ML моделей

Была разработана линейная модель классификации и модель градиентного бустинга для многоклассовой классификации результатов матчей.
Была выбрана метрика F1: она учитывает recall и precision

1. Baseline модель (base_line.ipynb):
   Модель градиентного бустинга (catboost) с признаками :номер недели, сезон, домашняя команда, гостевая команда, стадион. Качество модели по метрике F1 составило 0,44

Разработка ML моделей

2. Разработана модель линейной классификации и модель градиентного бустинга с подбором гиперпараметров с помощью GridSearch на признаках: Домашняя и гостевая команда, год, месяц, час, стадион, менеджер.
F1 повысился до 0.468

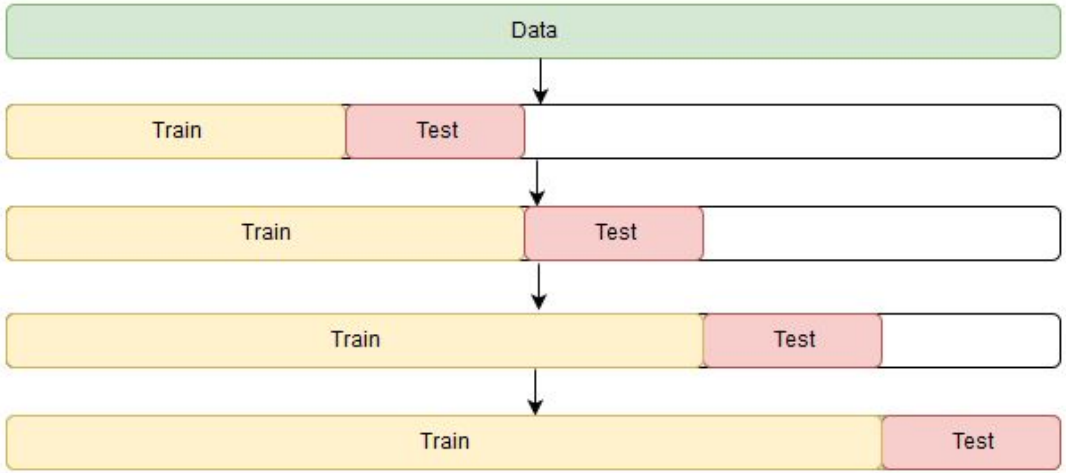
На следующем этапе добавили агрегированных признаков по каждой команде, рассчитанные на тренировочной выборке, также добавили лаговый признаки - статистика команды за предыдущий сезон, и обучили модель градиентного бустинга с подбором гиперпараметров
значение метрики F1: 0.486

Разработка ML моделей

3. На этом этапе (model_update.ipynb) для обучение модели мы использовали датасет из прошлого пункта, с добавлением лаговых признаков результат последних игр каждой команды и результат последних игр между двумя текущими командами. Была использована статистика по игрокам, каждая команда включает в себя статистику игроков (отранжировано от количества игр в сезоне и позиции игрока), т.е каждая команда представлена как совокупность статистик игроков, агрегированная статистика по клубу и статистика клуба, сдвинутая на год. Количество признаков получилось 1195.

Разработка ML моделей

Чтобы избежать утечку данных при обучении использовалось разбиение на трейн и валидацию как во временных рядах :



Разработка ML моделей

1. При обучении на этом датасете мы получили значение F1 = 0.494.
2. Чтобы увеличить метрику F1 мы ввели веса классов, чтобы модель уверенно предсказывала редкий класс (ничья), метрика поднялась до F1=0.532.
3. Установили порог предсказания (0.42), для того, чтобы не предсказывать результат там где модель не уверена и где она больше всего ошибается, F1=0.628. Больше всего модель ошибается на классе “ничья”, модель все еще редко его предсказывает, но и при его предсказании почти не ошибается (высокий precision)

Разработка ML моделей

4. По результатам анализа предсказаний по установленному нами порогу меняем его на 0.4 для ничьи и 0.48 для других классов. Количество предсказаний упало до 124 из 400, метрика F1 выросла до 0.692.
Далее будем везде применять этот порог для предсказания.

Разработка ML моделей

Применение метода снижения размерности и отбора признаков:

1. Применили метод снижение размерности PCA до 50 компонент, метрика F1 упала до 0.526, после использования порога предсказания F1=0.664.
2. Подобрали в цикле количество компонент от 50 до 650 с шагом 50, лучший результат получили на 500 компонент с метрикой F1=0.75
3. Отобрали признаки по важности исходного датафрейма применив SelectFromModel, в качестве estimator использовали решающее дерево. Лучшее качество по метрике F1 получили на 600 признаках F1=0.739

Планы на будущее

- Турнирная таблица(вывод в tg боте)
- Подписка на команду и матчи(вывод в tg боте)
- Уведомление о событиях в букмекерской компании с положительным математическим ожиданием(вывод в tg боте)
- улучшить модели предсказания используя больше данных, feature engineering
- применение методов анализа временных рядов
- доработка сервисов, внедрение инструментов промышленной разработки, внедрение DevOps и MLOps инструментов, инструментов автоматического тестирования
- Интеграция AirFlow в проект

Выводы

- научились собирать данные с помощью парсинга
- научились проводить исследование данных
- построили линейную модель и модель градиентного бустинга для задачи классификации
- разработали fastapi сервис, tg бота
- развернули сервисы с помощью docker-compose