

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

Кваліфікаційна робота
на здобуття ступеня бакалавра
за освітньо-науковою програмою «Інформатика»
спеціальності 122 «Комп'ютерні науки»
на тему:

**ПРОГНОЗ РЕЗУЛЬТАТІВ ФУТБОЛЬНИХ ПОДІЙ НА
ОСНОВІ МАШИННОГО НАВЧАННЯ**

Виконав студент 4-го курсу
Роман БОГДАН



(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Тарас ПАНЧЕНКО

(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент



(підпис)

Роботу розглянуто і допущено до
захисту на засіданні кафедри теорії та
технології програмування «08» травня
2023 р., протокол № 16

Завідувач кафедри

Микола НІКІТЧЕНКО

(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи: основний текст 42 сторінки, 6 ілюстрацій, 2 таблиці та 16 джерел посилань.

ПРОГНОЗУВАННЯ ФУТБОЛЬНИХ РЕЗУЛЬТАТІВ, DJANGO, DJANGO TEMPLATES, МАШИННЕ НАВЧАННЯ, ЗЧИТУВАННЯ, АНАЛІЗ ТА ОБРОБКА ДАНИХ.

Об'єктом даної роботи є моделі прогнозування результатів футбольних матчів. Предметом розробки є побудова моделі прогнозу кількості забитих м'ячів та розробка системи для аналізу футбольних подій.

Метою кваліфікаційної роботи є розробка системи прогнозу результатів футбольних матчів, спираючись на статистичні та історичні дані, у вигляді вебзастосунку.

Під час розробки було використано відкритий ресурс, який надає доступ до статистичних даних та розкладу з футбольних застосунків. Основним середовищем розробки було JetBrains PyCharm. Основною мовою розробки була Python, але для вебзастосунку також були використані: HTML, CSS, Javascript. Також, було використано Django Framework.

Результатом роботи є повноцінний вебзастосунок, який може прогнозувати результати футбольних матчів.

ЗМІСТ

| | |
|---|-----------|
| ВСТУП..... | 5 |
| РОЗДІЛ 1. ОГЛЯД НАЯВНИХ СИСТЕМ НА РИНКУ..... | 9 |
| 1.1 Вебзастосунок “Football Predictions.com”..... | 9 |
| 1.2 Вебзастосунок “Forebet”..... | 10 |
| 1.3 Вебзастосунок “Vitibet”..... | 12 |
| РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ..... | 14 |
| 2.1 Open source scraper Soccer Data..... | 14 |
| 2.2 Machine Learning..... | 15 |
| 2.3 Django Framework..... | 16 |
| РОЗДІЛ 3. ПРИЗНАЧЕННЯ ТА ЦІЛІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ..... | 18 |
| 3.1 Призначення системи..... | 18 |
| 3.2 Вимоги до системи..... | 19 |
| 3.3 Технічні вимоги до системи..... | 20 |
| РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ..... | 22 |
| 4.1 Реалізація зчитування та аналізу даних..... | 22 |
| 4.2 Реалізація інтерфейсу користувача..... | 25 |
| 4.3 Реалізація моделі прогнозування..... | 27 |
| РОЗДІЛ 5. ПОКРАЩЕННЯ ТА РОЗВИТОК ВЕБЗАСТОСУНКУ..... | 35 |
| 5.1 Покращення користувацького інтерфейсу..... | 36 |
| 5.2 Покращення моделі прогнозів..... | 38 |
| ВИСНОВКИ..... | 40 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 41 |

| | |
|---|-----------|
| | 4 |
| ДОДАТКИ..... | 43 |
| ДОДАТОК А. Прогнози футбольних матчів..... | 43 |
| ДОДАТОК В. Прогнози футбольних матчів та результати..... | 44 |
| ДОДАТОК С. Використання календарю..... | 45 |
| ДОДАТОК D. Таблиця футбольної ліги..... | 46 |
| ДОДАТОК Е. Фрагмент коду, відповідальний за пока списку матчів.... | 47 |
| ДОДАТОК F. Фрагмент коду, відповідальний за показ таблиці ліги..... | 49 |
| ДОДАТОК H. Фрагмент коду, відповідальний за прогноз..... | 51 |

ВСТУП

Оцінка сучасного стану розробки. Футбол є одним із найпопулярніших та найзахоплюючих видів спорту, який збирає мільйони шанувальників по всьому світу. Цей спорт має величезне значення як соціокультурне явище, впливаючи на спосіб життя, емоції та суспільну динаміку. У зв'язку з цим дослідження у галузі футболу має величезний науковий і практичний інтерес. Цей вид спорту вважають першим у світі. Деякі матчі стають гордістю та культурним спадком багатьох країн.

Нещодавно відбувся чемпіонат світу, який зібрав найкращі збірні та різні, навіть протилежні, культури. Протягом проведення турніру було багато спекуляцій та прогнозів щодо переможця. Деякі гадали за допомогою тварин, а дехто використав штучний інтелект для визначення тріумфатора. Більшість штучних інтелектів, спираючись на велику вибірку даних спрогнозували переможця правильно.

Футбольні матчі не є рідкісним явищем, вони проводяться кожного дня по всьому світу. Здебільшого штучні інтелекти використовуються в комерційних цілях, прикладом можуть слугувати букмекерські контори, які щодня виводять коефіцієнти для всіх матчів. Якщо проаналізувати цей спорт детально, можна прийти до висновку, що штучні моделі можуть приносити набагато більше користі даній сфері.

Якщо подивитися на штучний інтелект зі сторони тренерського штабу, то можна швидко зрозуміти, що автоматичні моделі можуть самостійно підібрати правильний склад на гру, спираючись на статистичний аналіз або навіть будувати тактику, яка допоможе слабкішій команді здобути перемогу. Це лише один приклад використання, проте користі може бути набагато більше.

Як зазначено вище, штучний інтелект може сильно змінити цей спорт, але на полі не буде штучного інтелекту – лише гравці, команда арбітрів та тренерські штаби, які і мають відіграти ключову роль у матчі. Можна навести таке філософічне речення: “Тренерські плани на гру руйнуються зі стартовим свистком гри”. Можна спрогнозувати поведінку за оптимальних умов, але ніхто не застрахований від несподіваної ситуації – наприклад нівелювання різниці в три м’ячі за десять хвилин.

Актуальність роботи та підстави для її виконання. Як зазначено вище, штучний інтелект уже бере участь у прогнозах на футбольні матчі для букмекерів, але на ринку не має великої кількості ресурсів, які прогнозують саме результат матчу для простих фанатів. Якщо потрібно оцінити шанси на перемогу певної команди, спираючись на статистичний аналіз чи спрогнозувати переможця турніру. Дана робота може бути використана в комерційних і практичних цілях.

Прикладом може бути комерційний вебзастосунок, який надає статистичні прогнози на матч. Такий додаток може запропонувати переможця матчу зі статистичними даними.

Практичним застосуванням даної роботи є допомога тренерським штабам оцінити шанси своєї команди та опрацювати деякі елементи гри. Оскільки модель спирається на статистику, то можна оцінити вірогідність пропущеного чи забитого голу, що допоможе відкоригувати чи взагалі змінити тактику на гру.

Футбол прогресує і деякі клуби використовують штучний інтелект, але далеко не всі клуби можуть собі дозволити власний застосунок. Розроблена програма є відкритою, та може бути використана ким завгодно.

Щодо підстав для виконання, багато людей є фанатами футболу та постійно слідкують за матчами, клубами та гравцями. Крім того, штучний інтелект є досить популярним та використовується у багатьох сферах людського життя. Вивчаючи штучний інтелект можна отримати нові

знання та цінний досвід. За цими причинами було обрано дану тему дипломної роботи.

Мета й завдання роботи. Метою кваліфікаційної роботи є розробка системи прогнозу футбольних матчів спираючись на статистичні та історичні дані, створити вебзастосунок, який зробить мої старання доступними користувачам. Для досягнення мети було поставлено та виконано наступні завдання:

- Проаналізувати існуючі застосунки для прогнозу футбольних результатів;
- Визначити спосіб аналізу та отримання даних.
- Розробити вимоги до програми;
- Реалізувати модель, яка може прогнозувати переможця, очікувані голи та ймовірність успішного результату команди;
- Реалізувати користувацький інтерфейс.

Об'єкт, методи та засоби розробки. Об'єктом даної роботи є моделі прогнозування результатів футбольних матчів. **Предметом розробки** є побудова моделі прогнозу кількості забитих м'ячів та розробка системи для аналізу футбольних подій. Розробка системи складається з наступних етапів:

- Отримання та аналіз даних;
- Створення моделі на отриманих даних;
- Розробка вебзастосунку.

Для розробки даного проекту було застосовано ітеративний цикл, що передбачає розробку, тестування та вдосконалення кожного етапу програми. Після тестування, конкретна частина програми оцінювалась та

вдосконалюватися, за необхідності тестування повторювалось. У результаті, отримано стабільний продукт.

Під час розробки було використано відкритий ресурс, який надає доступ до статистичних даних та розкладу з футбольних застосунків. Основним середовищем розробки було JetBrains PyCharm. Основною мовою розробки була Python, але для вебзастосунку також були використані: HTML, CSS, Javascript. Також, було використано Django Framework.

РОЗДІЛ 1. ОГЛЯД НАЯВНИХ СИСТЕМ НА РИНКУ

Ідея вебзастосунку з футбольними прогнозами не є новою. Як було зазначено вище, штучний інтелект часто використовується в футболі. На ринку вже є купа ресурсів, які пропонують користувачам отримати прогноз за деяку плату або безкоштовно. Кожен може відкрити такий застосунок та ознайомитися з деталями, обрати будь-який футбольний матч та подивитися на шанси тієї чи іншої команди. Здебільшого такі ресурси є безкоштовними.

1.1 Вебзастосунок “Football Predictions.com”

На рисунку 1 зображено користувацький інтерфейс застосунку “Football Predictions.com” [1]. Цей застосунок пропонує опис матчу, який має заохочити користувача до перегляду, опис команд та деяких найкращих гравців та можливість користувачам зробити власний прогноз проголосувавши за той чи інший результат.

Також, крім титульної сторінки є опція вибору конкретного матчу на ту чи іншу дату, перегляд турнірної сітки лише чемпіонату Англії та можливість прочитати новини, але вони давно не оновлювались. Іншими частинами цього застосунку є можливості поставити гроші на футбольні матчі. Після прогнозу, це є місцем найбільшого прикладання зусиль команди розробників.

Загалом, даний застосунок не пропонує нічого крім прогнозу результату для пересічного користувача. Великий акцент зроблено на

комерційну частину, можливо навіть більший чим на прогнозування результатів, прото цей вебзастосунок користується певною популярністю.

The screenshot displays the website's interface with a navigation bar at the top containing links for HOME, FOOTBALL PREDICTIONS, BET OF THE DAY, BETSLIP, FREE BETS, BOOKMAKERS, and NEWS. The main content area features match predictions for Bayern Munich vs Leipzig and Nottingham Forest vs Arsenal. Each prediction includes team logos, a brief analysis, a predicted score, and buttons for 'PREDICTION' and 'BET NOW'. A right-hand sidebar contains social media posts from @FBPredictions and a newsletter sign-up form.

Рисунок 1 - Зовнішній вигляд застосунку “Football Predictions.com”

1.2 Вебзастосунок “Forebet”

З назви можна зробити висновок, що даний застосунок також суто комерційний, але порівнюючи з попереднім він має набагато більше функціоналу для користувача та лише прогнозує результат без посилань на букмекерів [2]. На рисунку 2 зображено титульну сторінку даного ресурсу. На ній видно, що це програма пропонує відсоткове співвідношення на результат матчу, прогнозованого переможця та приблизний результат.

Крім, того на титульній сторінці є результат матчу, погодні умови та прогнозована кількість забитих м'ячів.

Застосунок має детальний аналіз матчу, який включає в себе останні результати команди, результати очних зустрічей і відстань до стадіону для обох команд.

Крім прогнозу та статистика для аналізу, вони показують статистичні факти про команди, наприклад те що команда не програвала п'ять матчів поспіль чи не пропускала на виїзді протягом місяця. На цьому ресурсі присутні статті щодо футбольних матчів, порівняння статистики двох любих команд, навіть якщо вони ніколи не грали між собою, але не дають прогнозу на це, лише статистичний аналіз. Також, можна знайти список травмованих гравців та посилання на їх інший застосунок, але присвячений тенісу.

The screenshot displays the 'Forebet' app interface. On the left is a dark sidebar menu with categories like 'MY LEAGUES', 'QUICK FOREBETS', and 'POPULAR LEAGUES'. The main content area is titled 'Mathematical football predictions /forebets/ and football statistics' and features a table of 'Featured matches'. Each row represents a match with columns for home/away teams, probability percentages, forebet numbers, correct scores, average goals, weather conditions, odds, and final scores.

| Home team | Away team | Probability % | | | Forebet | Correct score | Avg. goals | Weather conditions | Odds | Score | Live odds |
|---------------------|---------------|---------------|----|----|---------|---------------|------------|--------------------|------|-----------------------|-----------|
| | | 1 | X | 2 | | | | | | | |
| Valur Reykjavik | UMF Grindavik | 39 | 37 | 24 | 1 | 2-1 | 3.25 | 10° | 1.06 | FT 1-3 (0-2) | - |
| FC Twente W | PSV W | 35 | 25 | 40 | 2 | 1-2 | 3.45 | 12° | 7.50 | FT 4-0 (1-0) | - |
| Wolfsburg W | Freiburg W | 58 | 23 | 19 | 1 | 4-1 | 4.03 | 14° | 1.10 | FT 4-1 (1-1) | - |
| Rivers United | Doma United | 30 | 32 | 38 | 2 | 1-2 | 2.50 | 32° | 8.00 | FT 2-0 (2-0) | - |
| Fenerbahçe SK | Trabzonspor | 48 | 32 | 20 | 1 | 3-2 | 3.62 | 15° | 1.55 | FT 3-1 (1-0) | - |
| Zamalek SC | Aswan FC | 48 | 34 | 18 | 1 | 1-0 | 1.96 | 26° | 1.91 | FT 1-0 (1-0) | - |
| Vikingur Reykjavik | Grotta | 65 | 18 | 17 | 1 | 3-0 | 3.58 | 10° | 1.03 | FT 2-1 (1-1) | - |
| Newcastle United | Brighton | 29 | 35 | 36 | 2 | 0-2 | 2.04 | 16° | 3.60 | 76 FT 2-1 (2-0) | 49 |
| Sheffield Wednesday | Peterborough | 38 | 30 | 33 | 1 | 2-0 | 2.17 | 17° | 1.83 | 48 FT 2-0 (2-0) | 1.00 |

Рисунок 2 - Зовнішній вигляд застосунку "Forebet"

Даний застосунок має купу переваг для пересічного користувача та не є комерційним, лише прогнози на різні матчі та хороша статистика футбольних зустрічей.

1.3 Вебзастосунок “Vitibet”

Тут можна провести аналогію з минулим конкурентом. Додаток справді схожий на попередній, але має набагато менше функціоналу [3]. На рисунку 3 зображено титульну сторінку та користувацький інтерфейс даного застосунку. Першим чином в очі кидаються емблеми футбольних клубів - вони застарілі, та і сам інтерфейс виглядає застаріло. Проте цей застосунок все ще корисний для аналізу майбутніх матчів.

of salt. Our betting tips are based on a mathematical algorithm that predicts the results of future matches based on past sports results. This is tricky, of course, because the mathematical algorithm does not take into account other factors, from special motivation, to injuries of key players, and most importantly, it is still a statistic. However, Vitibet betting tips can serve as a good guide on which matches from the endless offer of bookmakers to further focus on and analyze them.

Join our tipsters competitions for the German Bundesliga, 2nd Bundesliga, Spanish LaLiga and Swiss Superliga. Each contest is sponsored with \$100 for the winner. If you have already joined our World Cup tipsters competition, no new registration is required. Just confirm your participation in the new leagues.

[ENTER VITISPORT TIPSTERS COMPETITIONS](#)

| FAQ INDEX | TOP TIPS IN 7 DAYS | VITISPORT TIP | | | | | INDEX | | |
|-------------------------------------|--------------------|----------------------|-------|------|------|------|-------|------|--|
| | | score | 1 | 0 | 2 | tip | | | |
| Turkey Süper Lig | | | | | | | | | |
| 18.05 | Fenerbahçe SK | Trabzonspor | 2 : 0 | 57 % | 24 % | 19 % | 1 | 2.9 | |
| United States MLS | | | | | | | | | |
| 18.05 | Minnesota United | Dynamo Houston | 1 : 2 | 30 % | 32 % | 38 % | 02 | -0.6 | |
| 18.05 | FC Cincinnati | CF Montréal | 1 : 0 | 41 % | 31 % | 28 % | 10 | 1.0 | |
| 18.05 | Columbus Crew | Los Angeles Galaxy | 2 : 0 | 55 % | 25 % | 20 % | 1 | 2.7 | |
| 18.05 | Orlando City | New York City FC | 1 : 0 | 39 % | 32 % | 30 % | 10 | 0.7 | |
| 18.05 | Philadelphia Union | Washington DC United | 2 : 1 | 43 % | 30 % | 27 % | 10 | 1.2 | |
| 18.05 | Charlotte FC | Chicago Fire | 2 : 1 | 42 % | 30 % | 28 % | 10 | 1.1 | |
| 18.05 | FC Dallas | Vancouver Whitecaps | 1 : 0 | 42 % | 30 % | 28 % | 10 | 1.1 | |
| 18.05 | Toronto FC | Red Bull New York | 1 : 1 | 34 % | 34 % | 33 % | 10 | 0.1 | |
| 18.05 | Los Angeles FC | Kansas City Wizards | 2 : 0 | 62 % | 22 % | 17 % | 1 | 3.5 | |
| 18.05 | Seattle Sounders | Austin FC | 2 : 0 | 57 % | 24 % | 19 % | 1 | 2.9 | |
| 18.05 | Nashville SC | Inter Miami CF | 1 : 0 | 48 % | 28 % | 25 % | 1 | 1.8 | |
| 18.05 | Atlanta United | Colorado Rapids | 1 : 1 | 32 % | 33 % | 35 % | 02 | -0.2 | |
| 18.05 | Real Salt Lake | Portland Timbers | 1 : 2 | 27 % | 30 % | 43 % | 02 | -1.2 | |
| Belorussia Vysheysshaya Liga | | | | | | | | | |
| 18.05 | FC Smorgon | Shakhter Soligorsk | 1 : 3 | 19 % | 24 % | 58 % | 2 | -3.0 | |
| 18.05 | Dinamo Minsk | Energetik-BGU Minsk | 3 : 0 | 82 % | 13 % | 5 % | 1 | 6.3 | |
| 18.05 | Dinamo Brest | FC Gomel | 1 : 0 | 37 % | 32 % | 31 % | 10 | 0.5 | |
| 18.05 | Torpedo Zhodino | Naftan Novopolotsk | 2 : 0 | 75 % | 16 % | 9 % | 1 | 5.2 | |
| 18.05 | Isloch Minsk | BATE Borisov | 1 : 0 | 37 % | 32 % | 31 % | 10 | 0.5 | |

Рисунок 3 - Зовнішній вигляд застосунку “Vitibet”

Даний застосунок містить дуже детальну статистику команд та футбольних ліг. Це є великою перевагою для аналітики, майже всі корисні дані зібрані в одному місці. На рахунок прогнозів, вони тут також присутні. Застосунок дає прогнозований рахунок та будує індекси команд у даному матчі, але це не основна його частина. Як було зазначено вище, статистика - головна частина. Ресурс має схожий функціонал для інших видів спорту, таких як баскетбол чи хокей.

Загалом, даний ресурс зосереджений на аналізі та статистики, та пропонує прогнозований рахунок для більшої повноти картини, але очікується що його будуть використовувати для ставок на футбольні події.

РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

Для розробки та реалізації системи були використано наступні технології:

- Open source scraper Soccer Data;
- Machine Learning;
- Django Framework.

2.1 Open source scraper Soccer Data

Soccer Data - це відкритий ресурс для зчитування даних з футбольних платформ, написаний на мові програмування Python [4]. Він зчитує інформацію з “Klub Elo”, “ESPN”, “DBref”, “FiveThirtyEight”, “SoFifa” та “Who Scored”. У результаті він повертає Pandas DataFrame з ідентифікаторами для всіх датасетів.

Перевагою даного ресурсу є можливість зберігати дані в кеші, що допомагає оптимізацію програми та швидко відповідати на запити користувача. Дана технологія парсить дуже детальну статистику та дає купу даних для аналізу й побудови моделі. Саме цей ресурс є основою моєї програми, він дає змогу продемонструвати розклад матчів, таблицю ліги та дає дані для тренування моделі машинного навчання.

Набір різних футбольних платформ для зчитування даних, дозволяє створити купу функцій для користувача. Це може бути детальний аналіз

результату матчу, чи, навіть рейтинг з характеристиками гравців, базований на базі даних Fifa.

2.2 Machine Learning

Машинне навчання (Machine Learning) - це галузь штучного інтелекту, що вивчає алгоритми та моделі, які дають комп'ютерам здатність самостійно навчатися та приймати рішення на основі даних без явного програмування [5-11].

У машинному навчанні комп'ютерна програма будує математичну модель, яка може аналізувати та виходити на залежності у великих обсягах даних, шукати закономірності та зробити прогнози або зробити рішення на основі цих даних. Це може бути застосовано в різних галузях, таких як медицина, фінанси, транспорт, реклама та багато інших.

Одним з ключових понять у машинному навчанні є "навчання на основі прикладів". Комп'ютерна програма отримує велику кількість прикладів даних, які містять вхідні параметри та відповідні результати. Вона використовує ці дані для побудови моделі, яка може узагальнити залежності та зробити прогнози для нових невідомих даних.

У даній роботі машинне навчання використовується для прогнозу футбольних матчів. Результатом прогнозу є можливі пропущенні та можливі забиті голи першою та другою командами, після цього виводиться фінальне значення можливого рахунку. Модель тренується на даних за останні декілька сезонів, як і на результатах, так і на статистиці, це допомагає розширити спектр аналізу та побудувати більш точні результати. Важливим компонентом даної моделі є очікуємі голи команди в завершеному матчі. Вони вагому роль у аналізі, тому що є агрегацією

статистики всього матчу та допомагають побудувати саме прогноз, а не очікуємий результат на основі завершених матчів.

Машинне навчання чудово підходить для теми та мети моєї роботи та є основною частиною проекту.

2.3 Django Framework

Django - це відкритий веб-фреймворк для швидкої розробки веб-додатків на мові програмування Python [12]. Django надає зручні інструменти для створення потужних і надійних веб-додатків з мінімальними. Основні можливості Django включають:

Архітектура Model-View-Controller (MVC). Django використовує архітектурний підхід MVC, який дозволяє розділити бізнес-логіку, представлення та обробку даних. Це спрощує розробку і підтримку коду.

Об'єктно-реляційне відображення (ORM). Django надає потужний ORM, який дозволяє вам працювати з базою даних, використовуючи об'єктно-орієнтований підхід замість написання SQL-запитів безпосередньо. Це спрощує роботу з базами даних і забезпечує переносимість між різними Системами Керування Базами Даних (СКБД).

Адміністративний інтерфейс: Django надає вбудований адміністративний інтерфейс, який дозволяє легко керувати даними в вашому додатку. Ви можете створювати, редагувати і видаляти записи в базі даних без написання додаткового коду.

Роутинг та URL-шаблони. Django має потужний механізм роутингу, який дозволяє визначати URL-шаблони та направляти запити до відповідних представлень. Це дозволяє структурувати ваші URL-адреси і забезпечує чистий та логічний маршрутизаційний шлях.

Шаблони. Шаблони дозволяють створювати динамічні веб сторінки, використовуючи мову DTL для визначення зовнішнього вигляду та структури веб сторінок [13-15].

Django Framework є базовою частиною мого проекту, він поєднує модель для прогнозів та користувацький інтерфейс. Було розроблено веб сервіс який надає інформацію для користувача в залежності від запиту. Застосунок отримує запит, обробляє його та повертає дані, отримані зі зчитування футбольних платформ.

За допомогою Django, а саме його шаблонів, було написано алгоритм показу всіх можливих матчів із прогнозами на результат. Шаблони дуже сильно спростили мені написання коду на мові html.

РОЗДІЛ 3.

ПРИЗНАЧЕННЯ ТА ЦІЛІ СТВОРЕННЯ ВЕБЗАСТОСУНКУ

3.1 Призначення системи

Призначенням системи є використання проекту для покращення футбольного аналізу у вільному доступі. Кожен фанат може відкрити та подивитися прогноз на гру його улюбленої команди, можливо заключити парі з кимось. А також, система може бути використана для аналізу нинішньої форми команди. Тренерський штаб може відкрити - оцінити форму команди, оцінити приблизний результат та внести певні корективи - наприклад зробити неочікувану заміну перед стартом гри. Приклад футбольного розкладу з прогнозами наведено в додатку А.

Кваліфікаційна робота передбачає:

- Вивчення та дослідження методів, технологій для розробки вебзастосунків;
- Дослідження та аналіз футбольної статистики;
- Пошук методів для реалізації системи;
- Програмну реалізацію системи;
- Тестування даних на основі реальних результатів.

Систему створено з метою:

- Аналізу футбольних матчів;
- Прогнозу футбольних результатів;
- Полегшення роботи тренерського штабу, шляхо оцінки шансів команди до старту матчу;

- Збільшення цікавості матчу для пересічних фанатів.

3.2 Вимоги до системи

Система по плану є у вільному доступі та доступна кожному без реєстрації. Кожен користувач інтернету може відкрити та отримати аналіз чи прогноз на обраний футбольний матч.

Для приваблювання користувачів, вебзастосунок має мати простий та привітливий дизайн. Якщо проаналізувати поведінку користувачів, то можна помітити, що здебільшого всі користуються лише титульними сторінками та рідко відвідують додаткові, тому має сенс розмістити основний функціонал програми на титульній сторінці, а побічний, такий як перегляд таблиці ліги, зробити окремими сторінками.

Дуже важливо врахувати, щоб інтерфейс був простим, це вже було згадано вище. Чим простіше сайт, тим краще буде користувачу адаптуватися і почати постійно ним користуватися. Також, не треба забувати про можливість історичних результатів та майбутніх матчів, застосунок має показувати аналіз по датам: минулим, теперішнім та майбутнім. Крім того, варто розділяти футбольні матчі на категорії: завершені, онлайн та майбутні для зручності користувача.

Вебзастосунок передбачає наступні реалізації:

- Можливість переглядати список минулих та майбутніх футбольних матчів, фільтруючи по даті;
- Прогнозування можливого результату матчу, кількості голів забитих командами;
- Перегляд таблиці футбольної ліги;
- Прогноз відсотку на той чи інший результат.

Щодо швидкості запитів, програма має швидко показувати список матчів, не дивлячись на те що вони читаються з різних футбольних ресурсів. Кешування має бути правильно реалізовано, щоб надавати максимально свіжу та валідну дату.

Прогнозування футбольного матчу є важкою справою, тому система яка має правильність приблизно п'ятдесят - шістдесят відсотків буде великим успіхом. Це футбол, тут усе залежить від гравців та ніхто не застрахований від неочікуваної ситуації на полі. Правильність відсоткового розрахунку має бути приблизно в цьому діапазоні.

Отже, головною частиною системи є прогноз і головною вимогою від користувача буде правильність прогнози більша за п'ятдесят відсотків - це дозволить користувачам стабільно нас використовувати в своїх цілях. Також дуже важливо надати всю потрібну інформацію, від розкладу матчів до положення команд у таблиці ліги.

3.3 Технічні вимоги до системи

Застосунок повинен бути розроблений, використовуючи технологію Django Framework. Цей фреймворк має поєднати оформлення застосунку з реальними даними за допомогою шаблонів.

Результати та календар футбольних матчів має бути прочитано за допомогою відкритого ресурсу "soccerdata" та приведено до приємного вигляду. Головною ідеєю є те, що замість бази даних ми використовуємо відкритий ресурс який зчитує останні дані, тому при перенесенні чи відміні будь якого матчу ми будемо мати правильний результат. Дуже важливо правильно розрахувати кешування даних, тому що зчитування нових даних кожного разу це дуже довго та не оптимально, а зберігання

усіх даних у файлі не дає можливості їх оновлення та, наприклад, перегляду поточних результатів матчів у прямому ефірі.

Крім того, потрібно подбати про користувацький інтерфейс. Як було уже зазначено вище, застосунок має бути простим та максимально корисним для користувача. На це треба звернути увагу та виділити час.

Дуже важливо розробити правильну модель, яка зможе прогнозувати максимально точні результати. Головною вимогою до такої моделі, є звичайно її точність, вища за п'ятдесят чи шістдесят відсотків. Щоб досягти такої високої планки, потрібно правильно натренувати модель, тобто підібрати максимально зручний набір даних з правильними результатами. Для цього, бажано знову таки використовувати відкритий ресурс “soccerdata” для зчитування статистики, а після того як усі дані будуть у нас на руках - лише надати їх натренованій моделі. Дуже важливо, максимально розширити дані та надати як можна більше правильної інформації.

РОЗДІЛ 4. РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ

4.1 Реалізація зчитування та аналізу даних

Для зчитування даних, було використано відкритий ресурс “soccerdata”. Це парсинг даних на мові програмування Python. Даний ресурс зчитує дані з футбольних платформ та передає їх як датафрейм. Здебільшого використано дані з платформи “FiveThirtyEight”. Дана платформа містить розклад майбутніх футбольних подій та результати минулих матчів, саме які й показано користувачу. А також, звідси береться положення клубів у таблиці футбольної ліги. Дана платформа може видати дату, указавши на вхід назву ліги та рік сезону, тобто можна отримати дату з будь якої доступної ліги за будь який рік - отже це можна буде використати для моделі.

Дані для обробки та аналізу програма отримує в форматі датафрейму. Дата фрейм містить наступні поля:

Таблиця 1 - структура дата фрейму на запит футбольних матчів

| Номер | Поле | Опис |
|-------|---------------|--|
| 1 | league | назва та ідентифікатор футбольної ліги |
| 2 | season | рік початку футбольного сезону |
| 3 | game | ідентифікатор гри, який включає в себе дату, домашню та виїзні команди |
| 4 | date | дата та час проведення футбольного матчу |
| 5 | status | статус проведення матчу(pre, live, post) |

| | | |
|----|--------------|---|
| 6 | home_team | ім'я домашньої команди |
| 7 | away_team | ім'я виїзної команди |
| 8 | home_id | ідентифікатор домашньої команди, здебільшого це число |
| 9 | away_id | ідентифікатор виїзної команди, здебільшого це число |
| 10 | home_code | скорочене ім'я домашньої команди |
| 11 | away_code | скорочене ім'я виїзної команди |
| 12 | score_home | кількість м'ячів, забитих командою вдома, число змінюється протягом матчу і до початку матчу є пустим |
| 13 | score_away | кількість м'ячів, забитих командою в гостях |
| 14 | chances_home | очікувана кількість забитих м'ячів домашньої команди, тобто агрегація статистики, яка показує на яку кількість голів команда зіграла і скільки могла забити |
| 15 | chances_away | очікувана кількість забитих м'ячів виїзної команди |

Отже, на основі цих даних можна вивести все футбольні події за датою. Для цього читаються всі футбольні матчи топових ліг, фільтрую за датою та після ще одного етапу обробки відправляю користувачу. Приклад коду, який відповідає за зчитування:

```
def read_matches(date_param, status_param):
    matches = read_games().filter(like=date_param.__str__(), axis=0).sort_values(by=['date'])
    if status_param is not None:
        matches = matches[matches['status'] == status_param]
    return matches.fillna('nan')
```

```
def read_games() -> pd.DataFrame:
    return pd.concat(list(map(lambda league: sd.FiveThirtyEight(league, '2022', no_cache =
False).read_games(), leagues)))
```

Зчитування матчів не є одним функціоналом роботи. За допомогою цього ресурсу можна зчитати нинішнє положення команди в таблиці футбольної ліги. Дата фрейм містить наступні поля:

Таблиця 2 - структура дата фрейму на запит футбольних ліг

| Номер | Поле | Опис |
|-------|----------------|--|
| 1 | league | назва та ідентифікатор футбольної ліги |
| 2 | season | рік початку футбольного сезону |
| 3 | game | ідентифікатор гри, який включає в себе дату, домашню та виїзні команди |
| 4 | team | назва та ідентифікатор футбольної команди |
| 5 | code | скорочення назва команди |
| 6 | current_losses | кількість поразок команди за поточний сезон |
| 7 | current_ties | кількість матчів зіграних внічию даною командою за поточний сезон |
| 8 | current_wins | кількість перемог команди за поточний сезон |
| 9 | current_points | кількість балів набраних командою на даний момент сезону |

За допомогою цих полів можна збудувати список команд, додавши такі параметри як кількість набраних балів, перемоги, нічії та поразки в

сезоні, звичайно відсортувавши за кількістю балів. Також, додавши значення полей з кількостями перемог, поразок та нічьїх - можна отримати загальну кількість ігор у сезоні, що є дуже важливим полем. Ось фрагмент коду, який відповідає за показ таблиці футбольної ліги:

```
def read_league(league_id):
    size = leagues.get(league_id)[1]
    forecasts = sd.FiveThirtyEight(leagues.get(league_id)[0], '2022',
no_cache=True).read_forecasts().filter(regex=("current_*"))
    teams = forecasts.sort_values(by=['last_updated', 'current_points'], ascending=[False,
False]).head(size)
    teams['team_name'] = teams.index.get_level_values(3)
    teams['position'] = list(range(1, size + 1))
    teams['match_played'] = teams[['current_wins', 'current_ties', 'current_losses']].sum(axis=1)
    return teams
```

4.2 Реалізація інтерфейсу користувача

Отже, як було зазначено вище, для побудови користувацького інтерфейсу, використано Django шаблони. Воно дозволяють швидко та зручно виводити отримані дані. Для візуалізації, звичайно, були використані Html та Css. При реалізації дуже корисним був застосунок CodePen із деякими шаблонами, які було використано [16]. Даний застосунок, обов'язково має бути доступним для пересічних користувачів, тому покращення моїх навичок у цій сфері стало обов'язковим протягом написання кваліфікаційної роботи.

Список головних ендпоінтів користувача не є великим, загалом їх лише два - основою роботи є модель та правильність прогнозування.

Отже, першою та головною частиною користувацького інтерфейсу є головна сторінка. Вона містить список усіх матчів, доступних сьогодні. Цей список - це результат отриманого дата фрейму, який було виведено за допомогою цикла `for` у шаблоні Django. Приклад титульної сторінки наведено в додатку А та В. Фрагмент коду, відповідальний за показ списку футбольних матчів наведений у додатку Е.

Звичайно, користувач може відфільтрувати дані за датою, для цього було розроблено інтерактивний календар, який саме і допоможе обрати потрібну дату та переглянути відповідні матчі. Значення дати передається як параметер реквесту та фільтрує дані на рівні датасету. Приклад використання календарю наведено в додатку С.

Усі футбольні матчі можна відфільтрувати за трьома категоріями: майбутні матчі, завершені та ті що проходять прямо зараз. Мій застосунок дає змогу відфільтрувати всі матчі за цим показником, що покращує користувачу аналізувати дані. Фільтрація виконується на рівні датасету.

На правій панелі від імен команд та логотипів, показано відсотки на три можливі кінцівки матчу. Ці дані було отримано за допомогою штучного інтелекту. Також, знайти та правильно поставити всі логотипи було досить не простою задачею, вони були приєднані по іменам, але в результаті вийшло досить гарно.

Між назвами команд та логотипами можна помітити поточний або кінцевий рахунок зустрічі. Також трохи далі було додано результати прогнозу, тобто кількість очікуваних голів для кожної з команд. Приклад прогнозування наведено в додатку А та В.

Ну що ж, крім перегляду всіх матчів, на панелі зліва є посилання на всі доступні футбольні ліги, тобто на їх таблицю. За це відповідає інший ендпоінт, який повертає дані лише за одним параметром - назвою футбольної ліги. Для реалізації цього, також було використано Django

шаблон, який вивів кожен рядок з відповідною командою. Фрагмент коду, відповідальний за виведення даних футбольної ліги наведено в додатку F.

Варто зазначити, що Django Framework може опрацювати реквест, та зробити рендеринг використовуючи як і шаблони так і мову html. Це дуже сильно полегшує імплементацію, якщо маєш дуже обмежений набір знань з front end розробки. Фрагмент коду, відповідальний за обробку реквесту титульної сторінки зі сторони Django:

```
def home(request):
    date_param = request.GET.get('date')
    if date_param is None:
        date_param = date.today().__str__()
    status_param = request.GET.get('status')
    matches = read_matches(date_param, status_param)
    matches_with_predict = predict(matches).to_dict('records')
    context = {
        'matches': matches_with_predict,
        'status': status_param,
        'date': date_param
    }

    return render(request, 'football/home.html', context)
```

4.3 Реалізація моделі прогнозування

Найцікавішою частиною проекту є прогнозування. У даному пункті, буде детальний опис моделі прогнозування футбольних результатів. Для

тренування моделі, було використано футбольні результати минулих матчів, протягом останніх двох сезонів. Це зроблено тому що дані старше від двох років будуть відображати неточну картину, бо за два роки склади та тактики команд можуть кардинально змінитися. Якщо ж узяти меншу кількість ігор - модель може бути недотренованою та видавати доволі неточний результат. Приклади прогнозу наведено в додатку А та В. Додаток А містить прогнози до футбольних матчів, а додаток В включає значення прогнозу та результат для порівняння.

Модель здебільшого працює на спрощених даних, тобто використано лише реальний результат та можливі забиті м'ячі, тому що саме ці дані є основою прогнозування футбольних результатів та мають мати найвищу вагу в існуючій моделі, інші параметри, такі як статистика ударів, можливий склад чи прогноз погоди будуть мати значну меншу вагу. Протягом аналізу результату минулих матчів, очікувані голи беруться як агрегація статистики футбольної зустрічі та є найбільш корисним параметром. Плани щодо додавання інших параметрів наведені в розділі 5.

Побудова моделі. Для створення самої моделі, було використано Random Forest Regressor з декількома полями в результируючому датасету. Як було сказано, модель спочатку рахує забиті та пропущені голи командою, тобто на виході потрібно отримувати два поля. Random Forest Regressor здатна до обробки багатовимірних даних, що дозволяє працювати з даними, які включають багато ознак. А також, дана модель може визначити важливість кожної ознаки в моделі, тобто знайти ті, які мають найбільший вплив на результат прогнозу. У моєму випадку, дана модель може проаналізувати, що важливіше для команди: забити чи не пропустити в матчі та провести оцінку як команда грає з подібними футбольними колективами щоб правильно побудувати прогноз.

Для побудови моделі, зокрема ансамблю випадкових лісів задається N кількість дерев. Для побудови кожного дерева обираються випадковий

піднабір даних з тренувального набору даних та випадковий набір ознак для кожного розділення дерева. Для побудови кожного розділення дерева, треба обрати найкращі ознаки для конкретного спліта, визначити значення порогу та значення обраної ознаки. Після побудови дерев, можна прогнозувати результат, для цього потрібно:

1. Подати новий вхідний приклад;
2. Пропустити приклад через кожне дерево в ансамблі;
3. Отримати прогнозований результат для кожного дерев;
4. Обчислити фінальний прогноз, який буде середнім значення або медіаною прогнозів усіх дерев.

У Random Forest Regressor формули використовуються для побудови та оцінки кожного дерева в ансамблі. Основні формули, що використовуються включають:

- Вибір найкращої ознаки для розділення, тобто використано різні метрики, такі як ентропія для визначення найкращої ознаки та порогу розділення, який максимізує інформаційний приріст або зменшує нечистоту вузлів дерева. Формула:

$$H = \sum(p * \log_2(p)), \quad (4.1)$$

де p - ймовірність кожного можливого результату;

- Розділення дерева на дві гілки, що включає в себе застосування порогу розділень до значень обраної ознаки для розподілу прикладів між лівою та правою гілками дерева. Формула:

$$G = H(\text{parent}) - [\text{Weighted Average} * H(\text{left_child}) + (1 - \text{Weighted Average}) * H(\text{right_child})], \quad (4.2)$$

де H - ентропія;

Weighted Average - зважений середній відсоток прикладів, які потрапляють у лівий та правий вузли;

- Проходження прикладу через кожне дерево під час прогнозування результатів, перевірка значень обраної ознаки та порогу для кожного розділення в дереві;
- Вирахування середнього значення або медіани прогнозів для всіх дерев ансамблю:

$$\begin{aligned} \text{Predicted Value} = & (\text{Prediction_1} + \text{Prediction_2} + \dots \\ & + \text{Prediction_N})/N, \end{aligned} \quad (4.3)$$

$$\begin{aligned} \text{Predicted Value} = & \text{Median}(\text{Prediction_1} + \text{Prediction_2} + \dots \\ & + \text{Prediction_N}), \end{aligned} \quad (4.4)$$

де N - кількість дерев у ансамблі;

Prediction_1, Prediction_2 - прогнози результатів, отриманих у ансамблі.

Загалом модель включає в себе дві характеристики, це кількість забитих та пропущених м'ячів першої команди в матчі проти другої. Порахувавши вагу даних характеристик, отримано значення 0.508 та 0.492, де перший показник це забиті, а другий - пропущені. Можна зробити висновок, що вага забитих м'ячів є трохи вищою та вони відіграють більшу роль під час аналізу. До цих значень входять як і реальні результати, так і очікувані, що базуються на агрегації статистики.

Тренування моделі. Для тренування моделі, потрібно зчитати всі ігри, та відфільтрувати за статусом. Логічно, що для тренування треба лише завершені матчі. Після фільтрації, можна видалити непотрібні дані та розділити датафрейм. Варто зазначити, що тренування та прогноз результатів зроблено дуже не звичним шляхом. Для тренування використовуються такі дані: ідентифікатор команди, ідентифікатор противника та кількість забитих, пропущених голів командою. Тобто з одного матчу, отримуються два рядки датасету: результат першої команди А у грі проти команди В та результат команди В у грі проти команди А. Це

дозволяє розширити діапазон аналізу та прибрати лишні залежності між командами в датасеті.

Як було зазначено вище, використовуються результати команд як дані, але не все так просто. Тут датасет збільшується у два рази, бо як результати взято реальні та очікувані голи забиті командами. Очікувані голи в матчі - агрегація футбольної статистики, що показує приблизну кількість голів, котрі мали забити команди. Ці значення базуються на кількості ударів, відстані, точності ударів і так далі. Отже, з однієї зустрічі беруться два значення: очікуваних та реальних голів, що дозволяє розширити мій датасет та дати дані базуючись не лише на реальних голах, а й статистичних виключеннях минулого матчу. У додатку G наведено фрагмент коду, який відповідає за тренування моделі.

Прогнозування результатів. Для прогнозування результатів, ми маємо натреновану модель, описану в попередньому пункті. Здебільшого, прогнозування використовується для титульної сторінки, де показано список усіх матчів, тому ми прогнозуємо відразу для декількох подій. Для прогнозування, потрібно лише отримати ідентифікатори обох команд та відправити їх моделі. Раніше було зазначено, що модель рахує дані лише для однієї команди, тобто вказавши клуб та його супротивника - можна отримати очікуваний результат забитих і пропущених для даного клубу. Повторюємо цю процедуру для кожної команди зустрічі, тобто беру два значення. Практика показує, що значення досить схожі та можуть відрізнитися максимум на 0.1, тому після цього обчислює середнє значення - і ми маємо прогноз для футбольного матчу.

Крім прогнозу можливих голів, даний застосунок рахує процентне відношення на той чи інший результат. Воно рахується за допомогою формули, що базується на очікуваних голах. Моя програма спочатку

розраховує процентне відношення на нічию, а потім базуючись на ньому - рахує шанси першої та другої команди. Приклад процентного відношення є в додатку А та В. У додатку Н наведено фрагмент коду, який містить прогноз футбольних подій.

Тестування моделі. Протягом імплементації, було проведено декілька тестів, наскільки добре дана моель прогнозує результати. На рисунках 4 та 5 зображено порівняння реальних результатів та прогнозованих, базуючись на тестових даних. Для побудови графіку, було взято десять останніх зіграних матчів, даного сезону. Після проведення тренування, було побудовано прогноз на кожен із десяти матчів. На рисунку 4 позначено порівняння результатів команд удома, а на рисунку 5 на виїзді.

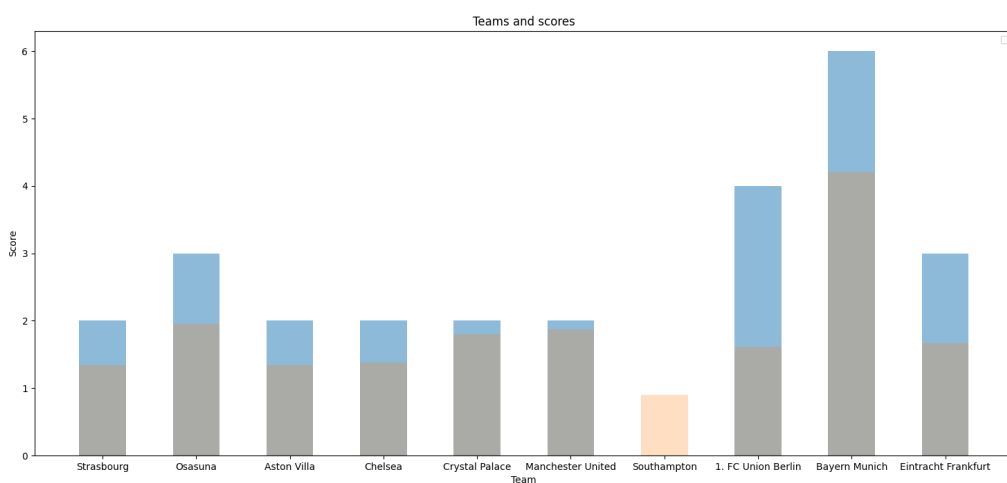


Рисунок 4 - Порівняння реальних голів та прогнозованих результатів команд вдома

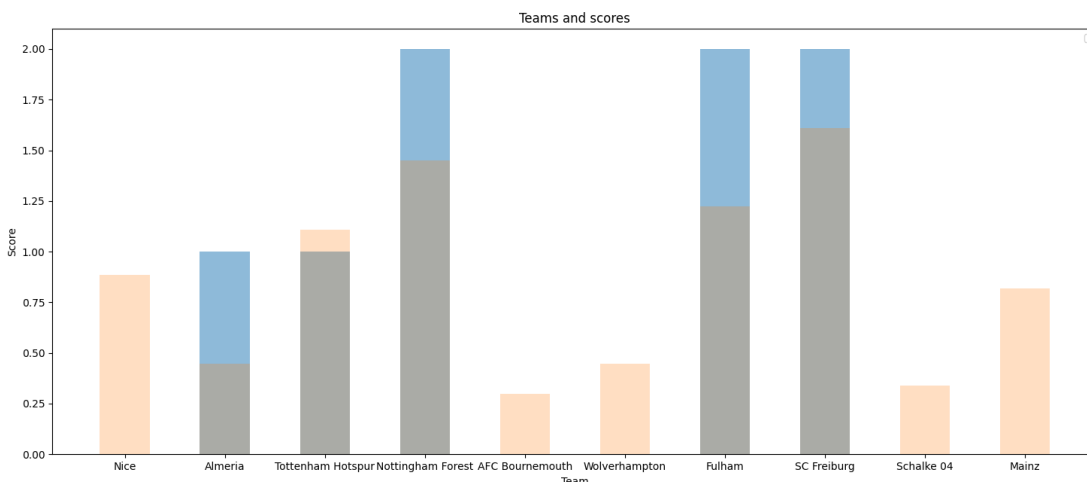


Рисунок 5 - Порівняння реальних голів та прогнозованих результатів команд у гостях

На рисунку, можна побачити, що модель може приблизно прогнозувати кількість забитих м'ячів командами. Так, звичайно, можна помітити великі різниці в прогнозі та реальному результаті, але це футбол і тут досить великий вплив мають гравці та інші нестандартні ситуації на футбольному полі, тому іноді дуже важко побудувати точну модель. Загалом, середня похибка не перевищує одного забитого м'яча, що є досить хорошим результатом.

Дуже важливою частиною тестування є розрахунок точності моделі. Спираючись, на тестові дані було проведено розрахунок точності моделі на прогнозування точності забитих м'ячів. Модель показала точність близько 40 відсотків прогнозу забитих м'ячів у матчі, що є досить хорошим результатом. Для точності прогнозування переможця, було вручну порівняно результати прогнозування до матчу та після, модель показала точність вище 50 відсотків.

Було обчислено матрицю помилок, для її побудови прогнозовані дані було заокруглено, що вплинуло на точність матриці. На рисунку 6 зображено матрицю помилок, на ньому можна побачити що похибка є

досить невеликою. Якщо ж модель і помиляється то здебільшого не більше чим на один забитий м'яч, що можна рахувати успішним результатом.

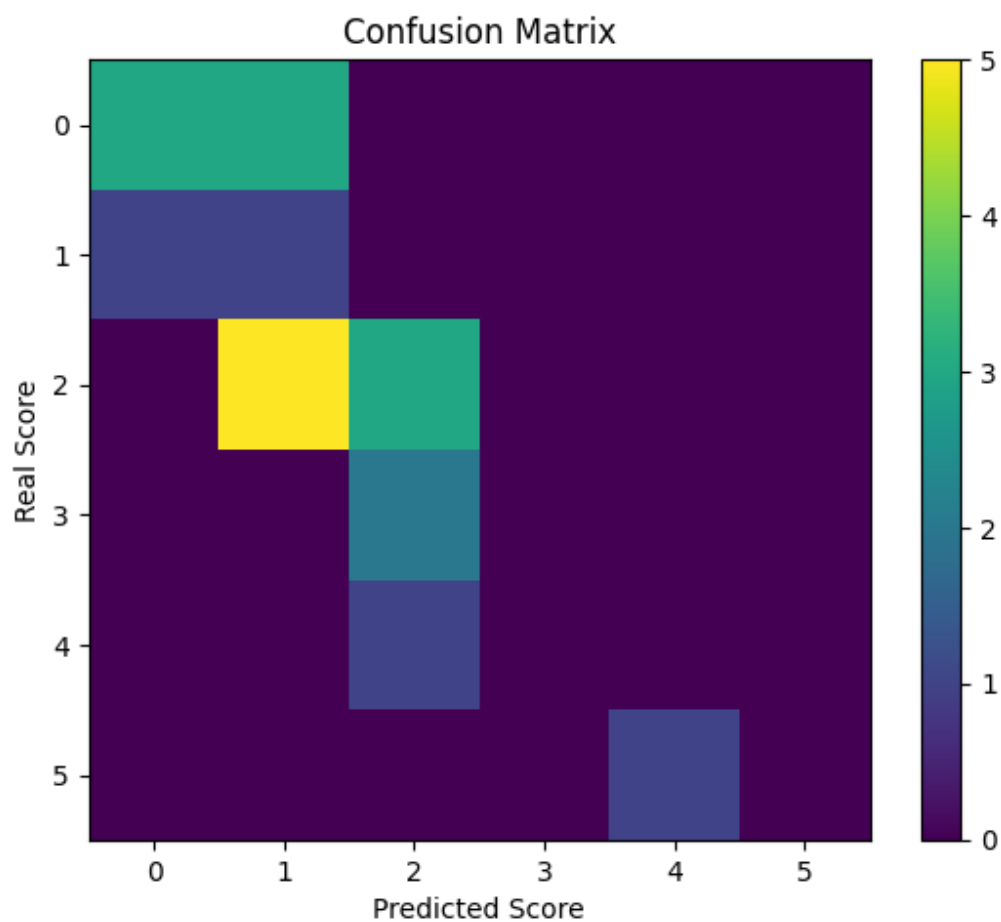


Рисунок 6 - Матриця помилок моделі

РОЗДІЛ 5.

ПОКРАЩЕННЯ ТА РОЗВИТОК ВЕБЗАСТОСУНКУ

У даному розділі, буде описано шляхи покращення та розвитку даного застосунку. Найголовнішим кроком для розвитку моєї системи є відхід від зчитування даних за допомогою відкритого ресурсу до використання футбольних ресурсів, які надають усі футбольні дані шляхом HTTP запитів. Звичайно, такі ресурси надають дані за відповідну плату та не є безкоштовними, тому не використовувались протягом написання кваліфікаційної роботи, але з розвитком застосунку обов'язковим кроком буде зміна способу отримання даних. В інтернеті досить багато схожих ресурсів, котрі надають всю потрібну інформацію, тому обрати буде не складно. Це дасть змогу, збільшити набір даних для аналізу та прогнозу, полегшить імплементацію нових сторінок та покращення користувацького інтерфейсу.

Також, ще одним загальним фактором розвитку є збільшення кількості футбольних ліг. Це не є можливим при нинішній імплементації з використанням зчитування даних, але буде можливо при переході на використання платних футбольних API. У світі існують сотні професіональних футбольних чемпіонатів та турнірів, ще одним кроком розвитку - є збільшення кількості футбольних турнірів, на які застосунок дає прогноз.

Варто зазначити, що додавання аналізу та прогнозу на матчі збірних також є досить важким та вагомим кроком для розвитку, це дозволить використовувати ресурс під час кубку світу чи кубку Європи з футболу - де проводяться найпрестижніші футбольні матчі.

5.1 Покращення користувацького інтерфейсу

Голосування за результат матчу. Так би мовити, прогноз користувачів мого вебзастосунку. Під час аналізу одного з конкурентів, було помічено можливість голосувати за результат футбольної події. Ця функція дає змогу оцінити не лише прогноз моделі, а й думку користувачів стосовно футбольної події. Як варіант, це можна врахувати в розрахуванні результату, але звичайно вага даної статистики буде досить малою. Головною ціллю, буде заохочити користувачів відчувати себе прогнозистами та пропонувати якісь бонуси за правильно вгадані результати. У останньому пункту ви дізнаєтесь про розвиток застосунку з комерційної точки зору, але тут варто зазначити, що бонусом за вгадані п'ять чи десять подій підряд може бути наприклад продвинута статистика на одну футбольну подію - це заохоче користувачів робити додаткову активність на сторінці мого застосунку.

Прогноз на результат футбольного турніру. Дана робота вже містить показ результатів футбольної ліги, але було б круто додати туди прогноз на місце по завершенні сезону. Це допоможе користувачам краще зрозуміти, на що претендує дана команда та які в неї шанси на чемпіонство. У цілому, це дасть більше даних для аналізу, а при натиску на любую позицію ліги - можна показати результати завершених матчів та прогнозовані результати майбутніх зустрічей, які і формують фінальне прогнозоване місце. Також, як було зазначено вище, у планах є додавання кубкових турнірів, де немає позиції команд, а лише турнір на вибування. Для такого виду змагань, можна побудувати комбінацію реальних завершених матчів із прогнозованими результатами та показати прогнозованих переможця, фіналістів чи півфіналістів. Про реалізацію

машинного навчання даної функції можна знайти більше інформації в пункті 5.2.

Продвинута статистика на завершені та майбутні футбольні події. Досить важливою новою функцією є продвинута статистика на майбутні футбольні зустрічі. Наприклад, на завершений матч, крім нашого прогнозу та реального результату, слід додати повну статистику футбольних клубів та гравців, для повноти аналізу. На майбутні матчі, крім нашого прогнозу потрібно додати трохи більше даних, таких як: минулі зустрічі команд, положення в футбольній лізі, прогнозовані склади, гравці котрі не можуть прийняти участь та продвинутий прогноз на футбольну подію. Детальний, або продвинутий, прогноз має включати в себе не лише процентне відношення та очікуєма кількість забитих голів, а приблизні склади команд, очікуємі оцінки гравців, очікуємий результат першого тайму, можлива кількість кутових, ударів та порушень правил і тд.

Футбольні статті та факти. Ще однією цікавою функцією для користувача може стати сторінка зі статтями та фактами. Головною їхньою ціллю є надихнути користувача на перегляд футбольного матчу. Ознайомившись з багатою історією протистояння, фактами та прогнозами - більшість людей захочу ввімкнути цю подію на своєму телевізорі чи комп'ютері та поринути в атмосферу футбольного матчу.

Live - перегляд матчу з реальними коментарями. Досить важливою функцією є перегляд текстовою трансляції з реальними коментарями. Далеко не всі матимуть змогу дивитися футбольний матч у відео - трансляції, тому їм стане в нагоді наш онлайн текстовий перегляд з реальними коментарями. Реалізація даної функції є неможливою зараз, лише за імплементації платних футбольних платформ, можна буде отримувати такі коментарі щосекунди.

Стрічка футбольних новин. Якщо чесно, тут ситуація схожа з попереднім пунктом. Для цього потрібно десь читати ці коментарі і буде

краще використати уже готовий футбольний ресурс з API, котрі можуть надати всю потрібну інформацію.

5.2 Покращення моделі прогнозів

Використання більшої кількості даних для аналізу. Нинішня модель досить ефективно використовує дані з реальною кількістю голів та очікуваними результатами, але з розвиток системи - потрібно збільшувати прогнозовані дані. На результат можуть впливати, загальна кількість ударів, кількість кутових, число фолів та коефіцієнт пресингу. Усі ці фактори потрібно врахувати та додати до тренування моделі, щоб збільшити відсоток точності прогнозу.

Додати вплив футболістів на фінальний прогноз. Крім статистики футбольних клубів нічого не впливає на поточний прогноз, але в планах потрібно додати можливість гравців впливати на прогнозування. Наприклад, ми розраховали прогноз команди на матч, але тут з'являється новина що ключовий нападник не зможе прийняти участь і нікому буде завершувати атаки, отже потрібно перерахувати та змінити прогноз, така ж сама ситуація і при оголошенні стартового складу команди. Додавання даної функції дуже сильно збільшить точність моделі прогнозування.

Додати розрахунок результату футбольного турніру. На основі розрахунку результату футбольних матчів можна розрахувати очікуємиий результат турніру, наприклад спрогнозувати переможця чи фіналістів. Для прогнозування футбольної ліги, потрібно взяти нинішню кількість балів команд та додати до неї прогнозовану кількість, базовану на прогнозованих результатах майбутніх матчів. Тобто, пропустивши всі майбутні футбольні події цієї ліги через модель машинного навчання,

можна порахувати прогнозовані бали, додати до реальних та отримати прогноз. Стосовно, кубкових турнірів ситуація там виглядає трохи легше. Потрібно лише спрогнозувати переможців у запланованих матчах і на їх основі побудувати прогнозовану сітку, де продовжувати прогнозувати результати, аж поки ми не отримуємо переможця. Якщо ж сітку побудувати не вдасться, наприклад жеребкування після кожного раунду, то після прогнозу запланованого раунда - просто обрати найсильнішу команду, яка в теорії має одержати перемогу. Ну прогнозувати випадкове жеребкування - це трохи занадто, тому лише можливий переможець.

ВИСНОВКИ

За підсумком кваліфікаційної роботи розроблено систему прогнозування футбольних матчів спираючись на статистичні та історичні дані, створено вебзастосунок, який є доступним для користувачів. Для досягнення мети виконано наступні завдання:

- Проаналізовано способи отримання футбольної статистики, проаналізовано існуючі додатки, котрі можуть прогнозувати результати футбольних подій, визначено переваги та недоліки кожного з них, поставлено відповідні задачі до розробки системи;
- Визначено спосіб аналізу та отримання даних, котрі підходять для розробки застосунку та побудови моделі;
- Спираючись на поставлені задачі, було сформовано вимоги щодо функціоналу системи та обрано інструменти для реалізації поставлених завдань. Розроблена система, здатна швидко відповідати на запити користувача та показувати прогнозування на майбутні події;
- Реалізовано модель, яка здатна прогнозувати переможця футбольного матчу, очікуємі забиті голи та ймовірність успішного результату команди;
- Реалізовано користувацький інтерфейс, який включає в себе розклад футбольних подій, таблицю для перегляду поточного місця команди та прогнози на футбольні матчі.

За підсумком роботи, можна зробити висновок, що результатом роботи є повноцінний вебзастосунок, який може прогнозувати результати футбольних матчів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Football predictions com [Електронний ресурс] - Режим доступу до ресурсу: <https://footballpredictions.com/>
2. Mathematical football predictions/forebets/ and football statistics [Електронний ресурс] - Режим доступу до ресурсу: <https://www.forebet.com/en/>
3. Vitibet free betting tips and livescore [Електронний ресурс] - Режим доступу до ресурсу: <https://www.vitibet.com/>
4. SoccerData, efficient scrapper soccer data from various source [Електронний ресурс] - Режим доступу до ресурсу: <https://github.com/probberechts/soccerdata>
5. Sebastian Raschka, Vahid Mirjalili, “Machine Learning in Python: Essential Techniques for Predictive Analyses”, 3rd Edition – 2019
6. Martin Görner, Ryan Gillard, Valliappa Lakshmanan, “Practical Machine Learning for Computer Vision”, O'Reilly Media – 2021
7. Machine learning algorithms for football predictions [Електронний ресурс] - Режим доступу до ресурсу: <https://towardsdatascience.com/machine-learning-algorithms-for-football-prediction-using-statistics-from-brazilian-championship-51b7d4ea0bc8>
8. Building a simple football prediction model using machine learning [Електронний ресурс] - Режим доступу до ресурсу: <https://medium.com/geekculture/building-a-simple-football-prediction-model-using-machine-learning-f061e607bec5>
9. Predictive modeling techniques - a comprehensive guide [Електронний ресурс] - Режим доступу до ресурсу: <https://www.projectpro.io/article/predictive-modelling-techniques/598>





































10. Eight machine learning models explained in twenty minutes [Электронный ресурс] - Режим доступа до ресурсу: <https://www.datacamp.com/blog/machine-learning-models-explained>
11. Defining predictive modeling in machine learning [Электронный ресурс] - режим доступа до ресурсу: <https://medium.com/analytics-steps/defining-predictive-modeling-in-machine-learning-887c23b7a278>
12. Aidas Bendoraitis, “Django 3 Web Development Cookbook”, Packt Publishing – 2020.
13. Django templates documentation with examples [Электронный ресурс] - Режим доступа до ресурсу: <https://docs.djangoproject.com/en/4.2/topics/templates/>
14. W3School [Электронный ресурс] - Режим доступа до ресурсу: https://www.w3schools.com/html/html_examples.asp
15. Django Templates [Электронный ресурс] - Режим доступа до ресурсу: <https://www.geeksforgeeks.org/django-templates/>
16. Codepen, resource to build, test and discover front-end code [Электронный ресурс] - Режим доступа до ресурсу: <https://codepen.io/>

ДОДАТКИ

ДОДАТОК А.
Прогнози футбольних матчів

| Football leagues | | All matches | Live | Completed | Scheduled | | | |
|------------------|--|-------------|--------|-----------|-----------|--|--|--|
| Time | | 1st | D | 2nd | | | | |
| 13:00 | Spezia (1.54) -- : -- (0.74) Torino | 48.0 % | 29.0 % | 23.0 % | | | | |
| 13:00 | Salernitana (0.9) -- : -- (0.7) Udinese | 26.0 % | 54.0 % | 20.0 % | | | | |
| 13:30 | 1. FC Union Berlin (1.81) -- : -- (0.72) Werder Bremen | 55.0 % | 23.0 % | 22.0 % | | | | |
| 13:30 | Borussia Dortmund (1.9) -- : -- (0.89) Mainz | 53.0 % | 23.0 % | 25.0 % | | | | |
| 13:30 | Borussia Monchengladbach (1.14) -- : -- (1.32) FC Augsburg | 29.0 % | 38.0 % | 34.0 % | | | | |
| 13:30 | Eintracht Frankfurt (1.64) -- : -- (1.24) SC Freiburg | 40.0 % | 30.0 % | 30.0 % | | | | |
| 13:30 | FC Cologne (0.91) -- : -- (2.21) Bayern Munich | 24.0 % | 19.0 % | 58.0 % | | | | |
| 13:30 | RB Leipzig (3.83) -- : -- (0.39) Schalke 04 | 89.0 % | 4.0 % | 8.0 % | | | | |
| 13:30 | VfB Stuttgart (1.73) -- : -- (1.67) TSG Hoffenheim | 36.0 % | 29.0 % | 35.0 % | | | | |
| 13:30 | VfL Bochum (0.49) -- : -- (1.95) Bayer Leverkusen | 17.0 % | 16.0 % | 67.0 % | | | | |

ДОДАТОК В. Прогнози футбольних матчів та результати

| Time | | 1st | D | 2nd | |
|-------|--|--------|--------|--------|---|
| 11:30 | Tottenham Hotspur  (1.57) 1 : 3 (2.11)  Brentford | 33.0 % | 23.0 % | 44.0 % |  |
| 12:00 | Girona FC  (0.77) 1 : 2 (1.99)  Villarreal | 22.0 % | 20.0 % | 58.0 % |  |
| 13:00 | Cremonese  (1.2) 1 : 5 (2.4)  Bologna | 27.0 % | 19.0 % | 54.0 % |  |
| 13:30 | Werder Bremen  (1.1) 1 : 1 (2.07)  FC Cologne | 27.0 % | 22.0 % | 51.0 % |  |
| 13:30 | TSG Hoffenheim  (1.66) 4 : 2 (2.23)  1. FC Union Berlin | 33.0 % | 22.0 % | 45.0 % |  |
| 13:30 | Schalke 04  (1.76) 2 : 2 (2.51)  Eintracht Frankfurt | 33.0 % | 19.0 % | 47.0 % |  |
| 13:30 | Hertha Berlin  (1.05) 1 : 1 (1.75)  VfL Bochum | 27.0 % | 27.0 % | 46.0 % |  |
| 14:00 | AFC Bournemouth  (0.35) 0 : 1 (2.04)  Manchester United | 13.0 % | 12.0 % | 75.0 % |  |
| 14:00 | Wolverhampton  (0.99) 1 : 1 (1.27)  Everton | 27.0 % | 39.0 % | 34.0 % |  |
| 14:00 | Liverpool  (2.12) 1 : 1 (2.19)  Aston Villa | 38.0 % | 23.0 % | 39.0 % |  |
| 14:00 | Fulham  (1.58) 2 : 2 (1.14)  Crystal Palace | 40.0 % | 31.0 % | 29.0 % |  |
| 14:15 | Athletic Bilbao  (0.54) 2 : 1 (1.0)  Celta Vigo | 19.0 % | 46.0 % | 35.0 % |  |

ДОДАТОК С. Використання календарю





















| All matches | | Live | Completed | Scheduled | |
|-------------|---------------------------------|-----------------------|-----------|---------------|----------------------|
| Time | | | | | Show |
| 10:30 | Lecce | (0.33) 0 : 0 (0.38) | | Spezia | |
| 11:00 | AC Ajaccio | (0.66) 0 : 5 (2.46) | | Stade Rennes | |
| 12:00 | Rayo Vallecano | (1.61) 1 : 2 (1.04) | | Espanyol | |
| 12:30 | West Ham United | (2.2) 3 : 1 (1.17) | | Leeds United | |
| 13:00 | Nice | (1.19) 0 : 0 (0.87) | | Toulouse | |
| 13:00 | Brest | (2.31) 2 : 1 (0.79) | | Clermont Foot | |
| 13:00 | Torino | (0.96) 1 : 1 (1.06) | | Fiorentina | 25.0 % 47.0 % 28.0 % |
| 13:00 | Reims | (1.47) 2 : 2 (1.26) | | Angers | 36.0 % 34.0 % 31.0 % |
| 13:00 | • Live Brighton and Hove Albion | (1.58) -- : -- (1.38) | | Southampton | 37.0 % 31.0 % 32.0 % |

MAY 2023

| Su | Mo | Tu | We | Th | Fr | Sa |
|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | | | |

ДОДАТОК D.

Таблиця футбольної ліги

| | CLUB | MP | W | D | L | PTS |
|----|--|----|----|----|----|-----|
| 1 |  Manchester City | 35 | 27 | 4 | 4 | 85 |
| 2 |  Arsenal | 37 | 25 | 6 | 6 | 81 |
| 3 |  Manchester United | 36 | 21 | 6 | 9 | 69 |
| 4 |  Newcastle | 36 | 19 | 12 | 5 | 69 |
| 5 |  Liverpool | 37 | 19 | 9 | 9 | 66 |
| 6 |  Brighton and Hove Albion | 36 | 18 | 7 | 11 | 61 |
| 7 |  Aston Villa | 37 | 17 | 7 | 13 | 58 |
| 8 |  Tottenham Hotspur | 37 | 17 | 6 | 14 | 57 |
| 9 |  Brentford | 37 | 14 | 14 | 9 | 56 |
| 10 |  Fulham | 37 | 15 | 7 | 15 | 52 |
| 11 |  Crystal Palace | 37 | 11 | 11 | 15 | 44 |
| 12 |  Chelsea | 35 | 11 | 10 | 14 | 43 |
| 13 |  Wolverhampton | 37 | 11 | 8 | 18 | 41 |
| 14 |  West Ham United | 37 | 11 | 7 | 19 | 40 |
| 15 |  AFC Bournemouth | 37 | 11 | 6 | 20 | 39 |
| 16 |  Nottingham Forest | 37 | 9 | 10 | 18 | 37 |
| 17 |  Everton | 37 | 7 | 12 | 18 | 33 |
| 18 |  Leeds United | 37 | 7 | 10 | 20 | 31 |
| 19 |  Leicester City | 36 | 8 | 6 | 22 | 30 |
| 20 |  Southampton | 37 | 6 | 6 | 25 | 24 |

ДОДАТОК Е.

Фрагмент коду, відповідальний за пока списку матчів

```

{% for match in matches %}
<tr>
  <td>
    <span class="matches__time matches__time--live">
      {{match.date|date:"H:i"}} </span>
      {% if match.status == 'live' %}
        <span class="tag tag--red tag--icon"><svg width="6" height="6"
viewBox="0 0 8 8">
          <circle fill="currentColor" cx="4" cy="4" r="4" />
        </svg>Live</span>
      {% endif %}
    </td>
  <td>
    <div class="score score--vertical">
      <div class="score__team score__team--vertical">
        <span>{{match.home_team}}</span>
      </div>
      <p class="score__result score__result--vertical">
        <span class="score__goals">({{match.home_predict}}) {% if
match.score_home != 'nan' %} {{match.score_home|floatformat:"0"}} {% else
%} -- {% endif %}</span><span class="score__separator">:</span><span
class="score__goals">{% if match.score_home != 'nan' %}
{{match.score_away|floatformat:"0"}} {% else %} -- {% endif %}
({{match.away_predict}})</span>
      </p>
      <div class="score__team score__team--vertical">
        <span>{{match.away_team}}</span>
      </div>
    </div>
  </td>
</tr>
{% endfor %}

```

```
</td>
<td>
  <span class="tag rating">{{match.home_win}} %</span>
</td>
<td>
  <span class="tag rating">{{match.draw}} %</span>
</td>
<td>
  <span class="tag rating">{{match.away_win}} %</span>
</td>
</tr>
{% endfor %}
```

ДОДАТОК F.

Фрагмент коду, відповідальний за показ таблиці ліги

```
{% for team in teams %}
<tr class="bg-black bg-opacity-20">
  <td class="pl-4">
    {{team.position}}
  </td>
  <td class="flex px-6 py-4 whitespace-nowrap">
    
    <span class="ml-2 font-medium">{{team.team_name}}</span>
  </td>
  <td class="px-6 py-4 whitespace-nowrap">
    {{team.match_played|floatformat:"0"}}
  </td>
  <td class="px-6 py-4 whitespace-nowrap">
    {{team.current_wins|floatformat:"0"}}
  </td>
  <td class="px-6 py-4 whitespace-nowrap">
    {{team.current_ties|floatformat:"0"}}
  </td>
  <td class="px-6 py-4 whitespace-nowrap">
    {{team.current_losses|floatformat:"0"}}
  </td>
  <td class="px-6 py-4 whitespace-nowrap">
    {{team.current_points|floatformat:"0"}}
  </td>
</tr>
{% endfor %}
```

ДОДАТОК G.

Фрагмент коду, відповідальний за тренування моделі

```

games_2022 = read_games_2022()
games_2021 = read_games_2021()
games = pd.concat([games_2022, games_2021])
train_all = games[games['status'] == 'post']
train = train_all[['home_id', 'away_id', 'score_home', 'score_away',
'chances_home', 'chances_away']].dropna()
train_stat_base = pd.concat([train[['home_id',
'away_id']].rename(columns={'home_id': 'Team', 'away_id': 'Enemy'}),
train[['away_id', 'home_id']].rename(columns={'away_id':
'Team', 'home_id': 'Enemy'})])
train_stat = pd.concat([train_stat_base, train_stat_base])
train_res = pd.concat([train[['score_home',
'score_away']].rename(columns={'score_home': 'GF', 'score_away': 'GA'}),
train[['score_away',
'score_home']].rename(columns={'score_away': 'GF', 'score_home': 'GA'}),
train[['chances_home', 'chances_away']].rename(
columns={'chances_home': 'GF', 'chances_away': 'GA'}),
train[['chances_away', 'chances_home']].rename(
columns={'chances_away': 'GF', 'chances_home': 'GA'})])

model.fit(train_stat, train_res)

```

ДОДАТОК Н.
Фрагмент коду, відповідальний за прогноз

```
games_2022 = read_games_2022()
games_2021 = read_games_2021()
games = pd.concat([games_2022, games_2021])
train_all = games[games['status'] == 'post']
train = train_all[['home_id', 'away_id', 'score_home', 'score_away',
'chances_home', 'chances_away']].dropna()
train_stat_base = pd.concat([train[['home_id',
'away_id']].rename(columns={'home_id': 'Team', 'away_id': 'Enemy'}),
train[['away_id', 'home_id']].rename(columns={'away_id':
'Team', 'home_id': 'Enemy'})])
train_stat = pd.concat([train_stat_base, train_stat_base])
train_res = pd.concat([train[['score_home',
'score_away']].rename(columns={'score_home': 'GF', 'score_away': 'GA'}),
train[['score_away',
'score_home']].rename(columns={'score_away': 'GF', 'score_home': 'GA'}),
train[['chances_home', 'chances_away']].rename(
columns={'chances_home': 'GF', 'chances_away': 'GA'}),
train[['chances_away', 'chances_home']].rename(
columns={'chances_away': 'GF', 'chances_home': 'GA'})])

model.fit(train_stat, train_res)
```