Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Факультет інформаційних технологій
(факультет)

Кафедра системного аналізу та управління
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня магістра

Студента Швидкого Романа Олександровича

академічної групи 124м – 22 – 1

спеціальності 124 Системний аналіз

на тему: «Інформаційно-аналітична система для прогнозування показників успішності спортивних команд»

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**Інституційною**	
кваліфікаційної роботи	*к.т.н., доц. Коряшкіна Л.С.*			
розділів:				
Інформаційно-аналітичний	*к.т.н., доц. Коряшкіна Л.С.*			
Спеціальний розділ	*к.т.н., доц. Коряшкіна Л.С.*			
Рецензент				
Нормоконтролер	*к.ф.-м.н., доц. Хом'як Т.В.*			

Дніпро
2023

ЗАТВЕРДЖЕНО:
завідувач кафедри
Системного аналізу та управління

(повна назва)

_____ *к.т.н., доц. Желдак Т.А.*
(підпис) (прізвище, ініціали)

«_____»_____20____року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня *магістра*

студенту *Швидкому Р.О.* **академічної групи** *124м-22-1*

спеціальності: *124 Системний аналіз*

на тему «*Інформаційно-аналітична система для прогнозування показників успішності спортивних команд»*
затверджену наказом ректора НТУ «Дніпровська політехніка»
від *09.10.2023 р. №1227-с*

Розділ	Зміст	Терміни виконання
1. Інформаційно-аналітичний розділ	*Проаналізувати структуру об'єкта дослідження. Визначити предметну область дослідження та проблему, що розв'язується. Обґрунтувати методи виконання поставлених завдань*	04.09.2023 – 18.10.2023
2.Спеціальнийр озділ	*Розв'язати поставлені задачі: проаналізувати результати виступів команд та їх відповідність до очікуваних, спрогнозувати зміни в результатах команд*	18.10.2023 – 30.11.2023

Завдання видано _____ *доц. Коряшкіна Л.С*
(підпис) (прізвище, ініціали)

Дата видачі: 04.09.2023 *р.*
Дата подання до екзаменаційної комісії:_____

Прийнято до виконання _____ Швидкий Р.О
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 55 с., 25 рис., 3 додатки, 8 джерел.

Об'єктом дослідження в роботі є процес аналіз виступів спортивних команд протягом тривалого відрізку часу з прикладу клубів Англійської Прем'єр Ліги.

Предметом дослідження є сукупність теоретико-методичних та науково-практичних аспектів аналізу спортивних команд

Мета даної кваліфікаційної роботи – це вивчення теоретичних підходів до статистичного аналізу даних про футбольні команди та розробка ефективних моделей для прогнозування їхніх результатів на основі використання аналітичних методів та візуалізації ключових показників гри.

Методи дослідження: метод головних компонент для виявлення ключових факторів, метод візуалізації для аналізу статистичних даних та побудова дерева рішень для стратегічного прогнозування результатів гри та визначення впливових чинників в контексті футбольних команд з використанням мови програмування Python.

В *інформаційно–аналітичному розділі* наведено теоритичні дані щодо методів, що будуть використовуватися при аналізі об'єкту дослідження. Поставлені задачі дослідження та обрано концепції їх розв'язання.

У *спеціальному розділі* сформовано алгоритм програми для аналізу поставленої задачі, написано програмний код для розв'язання існуючої проблеми.

Практична цінність отриманих результатів полягає в тому, що запропонована розроблена система дає можливість своєчасно виявити невідповідність фактичних результатів клубу його фактичному рівню гри.

Ключові слова: АНАЛІЗ , МОДЕЛЬ, ОЧІКУВАНІ ГОЛИ, ГОЛИ, ФУТБОЛ, ШТУЧНИЙ ІНТЕЛЕКТ.

ABSTRACT

Explanatory note: 55 pages, 25 figures, 3 appendices, 8 sources.

The object of the research in this work is the process of analyzing the performance of sports teams over an extended period of time, using examples from clubs in the English Premier League.

The subject of the study encompasses a set of theoretical, methodological, and scientific-practical aspects of analyzing sports teams.

The aim of this qualification work is to study theoretical approaches to the statistical analysis of data about football teams and develop effective models for predicting their results based on the use of analytical methods and visualization of key game indicators.

Research methods include the principal component method for identifying key factors, visualization method for analyzing statistical data, and decision tree construction for strategic forecasting of game results and identifying influential factors in the context of football teams, using the Python programming language.

The informational-analytical section provides theoretical information about the methods to be used in the analysis of the research object. Research tasks are formulated, and concepts for their solutions are chosen.

In the special section, an algorithm for the program analysis of the stated problem is formulated, and programming code is written to solve the existing problem.

The practical value of the obtained results lies in the fact that the proposed developed system allows timely identification of discrepancies between a club's actual results and its actual level of play.

Keywords: ANALYSIS, MODEL, EXPECTED GOALS, GOALS, FOOTBALL, ARTIFICIAL INTELLIGENCE.

ЗМІСТ

ВСТУП

Футбол є найпопулярнішим спортом в світі і з кожним роком він привертає до себе більше уваги не тільки з боку простих фанатів, а й з боку різноманітних компаній, які стають спонсорами футбольних команд або ж займаються діяльністю, що пов’язана напряму з футболом.

Тільки у 2022 році клуби три найбільших ліг світу (англійська, іспанська німецька) зареєстрували прибутки на суму 12.9 мільярдів євро (6.4 мрд., 3.3 мрд., 3.2 мрд. відповідно). Англійська Прем’єр Ліга є найбагатшою і найпопулярнішою футбольною лігою світу, про що свідчить той факт, що 11 з 20 найбагатших клубів планети базуються саме в Англії. Саме через подібну її актуальність для аналізу в даній роботі було обрано саме її.

Аналіз виступів команд та прогнозування їх подальших результатів можуть бути корисними як для самих команд та підготовки до конкретного суперника, так і для інших аспектів футболу, як от наприклад спонсорство команд для розширення свого бренду, інвестування в футбольні клуби, робота букмекерських контор, тощо. Актуальність даної теми пов’язана з все більшим і більшим інтересом до даного ринку з боку нових країн (ОАЕ, Саудівська Аравія, Катар, США), які приносять в спорт велику кількість грошей. Тому виникла потреба у глибокому вивченні методів аналізу виступів команд та їх результатів.

РОЗДІЛ 1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ

1.1 Статистичний аналіз даних в спорті

Статистичний аналіз даних в спорті є надзвичайно важливим інструментом для розуміння та покращення результатів спортивних команд, індивідуальних атлетів та загальних стратегій тренувань. Якщо ж статистичний аналіз даних проводиться на рівні всіх команд однієї спортивної ліги, то він може бути корисним для різних цілей і використовуватися в різних аспектах спортивного управління та стратегій.

Подібними цілями можуть бути:

- **Оцінка конкурентоспроможності команд:**
  - *Визначення лідера ліги:* Аналіз результатів та статистичних показників команд дозволяє визначити та оцінити лідера чемпіонату, що може бути важливим для спостереження за загальною конкурентоспроможністю.
- **Вивчення тактичних та стратегічних тенденцій:**
  - *Аналіз тактик гри:* Спостереження за тим, які тактики використовують різні команди, може допомогти розкрити успішні стратегії та найефективніші методи гри.
- **Оптимізація календаря матчів:**
  - *Розподіл сильних та слабких суперників:* Аналіз даних може використовуватися для оптимізації календаря матчів, роблячи його більш справедливим та конкурентоспроможним для всіх команд.
- **Аналіз травм та здоров'я гравців:**
  - *Моніторинг травм:* Спостереження за статистикою травм гравців може допомогти лікарям та тренерам приймати рішення щодо фізичного навантаження та управління ризиками.
- **Ефективність арбітражу:**
  - *Оцінка суддівства:* Статистичний аналіз може допомогти в оцінці роботи арбітрів, виявленні можливих проблем та вдосконаленні

системи арбітражу. В приклад можна привести нещодавнє рішення щодо введення в використання Англійською Прем’єр Лігою м’яча з функцією фіксації положення на полі та того, якою кінцівкою гравець доторкнувся до неї. Дане рішення приведе до більш чесних і чітких рішень арбітрів в таких важливих рішеннях як: чи вийшов м’яч за межі поля, чи була гра рукою з боку гравця, чи повністю м’яч пересік лінію воріт

- **Ефективність ліги:**
    - *Привабливість для глядачів та спонсорів:* Вивчення статистичних даних може допомогти зрозуміти, наскільки цікавою є ліга для глядачів та спонсорів, що може впливати на їхню зацікавленість.
- **Розробка нових правил та поліпшення лігового управління:**
    - *Оптимізація правил:* Аналіз даних може слугувати основою для внесення змін у правила гри або системи лігового управління з метою поліпшення конкурентоспроможності та привабливості ліги.
- **Реклама та маркетинг:**
    - *Сприяння рекламі та спонсорству:* Аналіз даних може надати важливу інформацію для рекламних компаній та спонсорів щодо цінності спортивної ліги як рекламної платформи. Ефективність арбітражу:

1.2 Візуалізація даних

Візуалізація статистичних даних грає важливу роль у розумінні та інтерпретації даних. Вона дозволяє перетворити числові дані на графічні зображення, що полегшує сприйняття та виявлення залежностей, закономірностей та аномалій.

Найпопулярнішими методами візуалізації статистичних даних є:

1. *Гістограми:* Вони використовуються для візуалізації розподілу одновимірних даних та вказують, як часто значення входять в певний діапазон. Гістограма дозволяє визначити форму розподілу, наявність аномалій та центральні тенденції.
2. *Діаграми:* розсіювання: Вони використовуються для вивчення взаємозв'язку між двома змінними. Кожна точка на діаграмі представляє пару значень, що дозволяє виявляти кореляції та аномалії.
3. *Лінійні графіки:* Використовуються для відображення змін змінних у часі. Це може бути корисно для виявлення тенденцій, сезонності та інших часових закономірностей.
4. *Кругові діаграми:* Вони представляють частки цілого та використовуються для відображення відносної частки кожної категорії у відсотках.
5. *Коробкові діаграми (Box Plots):* Цей тип графіку надає інформацію про мінімум, максимум, медіану, нижній та верхній квартилі та виявляє наявність викидів.
6. *Теплові карти:* Вони використовують кольорове кодування для відображення інтенсивності значень у двовимірному просторі. Це може бути корисно для виявлення патернів у великих наборах даних.
7. *Радарні графіки:* Використовуються для візуалізації багатовимірних даних, які представлені у вигляді точок у просторі.
8. *Контурні графіки:* Вони використовуються для відображення тривимірних даних на плоскій поверхні.

Важливо враховувати, що вибір конкретного методу візуалізації повинен відповідати меті дослідження чи аналізу, а також характеристикам даних, які вивчаються. Комбінування різних видів графіків та діаграм може надати комплексний погляд на статистичні дані.

1.3 Методи статистичного аналізу даних

Методи статистичного аналізу даних — це засоби та процедури, які використовуються для вивчення, аналізу та висновків з даних. Вони дозволяють вам робити висновки про популяцію на основі вибірки даних, проводячи різні види аналізу та використовуючи різні статистичні методи.

Основні методи статистичного аналізу включають у себе:

- *Описова статистика*: Це включає в себе методи, спрямовані на опис і узагальнення властивостей вибірки даних. До таких методів відносяться середні значення, медіани, моди, дисперсії та інші.
    - *Мета*: Надає підсумкову інформацію про основні характеристики даних, такі як середні значення, медіани, моди, дисперсії.
    - *Важливість*: Допомагає отримати загальне уявлення про розподіл та характеристики даних, спрощує їх інтерпретацію.
- *Інференційна статистика*: Цей тип статистичного аналізу використовується для формування висновків про популяцію на основі обмеженої вибірки даних. Включає методи, такі як довірчі інтервали, тестування гіпотез, аналіз варіації тощо.
    - *Мета*: Дозволяє робити висновки про популяцію на основі вибірки даних, використовуючи довірчі інтервали, тестування гіпотез, аналіз варіації та інші методи.
    - *Важливість*: Дозволяє здійснювати загальні висновки про популяцію на основі обмеженої вибірки даних та оцінювати невизначеність результатів.
- *Кореляційний аналіз*: Дозволяє визначити ступінь зв'язку між двома чи більше змінними. Коефіцієнт кореляції вказує на силу та напрямок цього зв'язку.
    - *Мета*: Визначає ступінь зв'язку між двома змінними.

- *Важливість*: Дозволяє виявляти та вимірювати силу та напрямок взаємозв'язків між різними змінними, що може бути важливо для прогнозування та розуміння даних.

- *Регресійний аналіз*: Використовується для розуміння відносин між залежною та незалежними змінними. Метою є прогнозування значень залежної змінної на основі значень незалежних.
  - *Мета*: Встановлює відносини між залежною та незалежними змінними.
  - *Важливість*: Дозволяє прогнозувати значення залежної змінної на основі значень незалежних, що важливо для моделювання та прогнозування.
- *Тестування гіпотез*: Статистичні тести використовуються для визначення, чи можна відкинути або не відкидати гіпотези про популяцію на основі даних вибірки.
  - *Мета*: Визначає, чи можна відкинути або не відкидати гіпотези про популяцію на основі даних вибірки.
  - *Важливість*: Допомагає приймати рішення на основі статистичних даних та перевіряти припущення про популяцію.

Ці методи статистичного аналізу є важливим інструментом для науковців, дослідників, бізнес-аналітиків та інших фахівців для отримання інсайтів з даних та прийняття обґрунтованих рішень.

1.4 Дерева рішень, методи їх побудови та їх переваги в аналізі

Дерево рішень в статистичному аналізі є моделлю машинного навчання, яка використовується для прийняття рішень на основі вхідних даних. Це графічна модель, що представляє собою деревоподібну структуру з вузлами та гілками. Кожен вузол у дереві рішень представляє рішення або тест для одного з атрибутів, а кожна гілка представляє можливий вихід з тесту.

Мета побудови дерева рішень:

- *Класифікація*: Дерева рішень можуть використовуватися для класифікації об'єктів у конкретні категорії або класи.
- *Регресія*: Ці моделі можуть також використовуватися для прогнозування значень числової змінної.

Процес побудови дерева рішень включає:

- *Вибір атрибута*: Обирається атрибут, який найкраще розділяє дані на підгрупи. Вибір зазвичай базується на метриці, такій як коефіцієнт Джині, ентропія чи середньоквадратична помилка, залежно від типу задачі (класифікація або регресія).
- *Розбиття даних*: Дані розбиваються на підгрупи відповідно до значень обраного атрибута.
- *Рекурсивний процес*: Процес повторюється для кожної нової підгрупи, поки не виконуються певні критерії зупинки, такі як максимальна глибина дерева, мінімальна кількість об'єктів у вузлі чи інші.
- *Листя дерева*: Коли досягається критерій зупинки, вузли стають листями, які представляють прогноз або клас для нового об'єкта.

Переваги дерева рішень в аналізі даних:

- *Легка інтерпретація*: Дерева рішень легко інтерпретувати та візуалізувати. Вони підходять для неспеціалістів, оскільки логіка прийняття рішень легко зрозуміла.
- *Врахування важливості атрибутів*: Дерева рішень дозволяють визначити важливість різних атрибутів у прийнятті рішень.

- *Можливість врахування нелінійних залежностей*: Дерева рішень можуть ефективно моделювати складні нелинійні залежності в даних.
- *Обробка якісних та кількісних даних*: Дерева рішень можуть обробляти різні типи даних без необхідності попередньої обробки.
- *Робастність до відсутності деяких даних*: Вони можуть ефективно працювати з неповними або відсутніми даними.

1.5 Кластеризація даних в статистичному аналізі

Кластеризація даних в статистичному аналізі — це процес групування схожих об'єктів або спостережень в кластери чи групи. Метою кластеризації є визначення природних структур або подібностей в даних, а також виділення груп, які можуть бути корисні для подальшого аналізу чи розуміння даних. Кластеризація може бути використана в різних галузях для полегшення роботи з даними

Основні етапи кластеризації

- Визначення метрики відстані: Вибір метрики відстані або схожості між об'єктами. Це може бути евклідова відстань, косинусна схожість, кореляція та інші.
- Вибір алгоритму кластеризації: Існує безліч алгоритмів кластеризації, таких як K-середніх, ієрархічна кластеризація, агломеративна кластеризація, DBSCAN та інші.
- Вибір кількості кластерів: У деяких алгоритмах необхідно заздалегідь вказати кількість кластерів (наприклад, у K-середніх), в інших алгоритмах кластери формуються автоматично.
- Виконання алгоритму кластеризації: Проведення самого процесу кластеризації згідно обраних параметрів та алгоритму.
- Оцінка та інтерпретація результатів: Оцінка якості кластеризації інструментами, такими як внутрішні метрики (наприклад, індекс Сілуету) та зовнішні метрики (наприклад, скоригований індекс Ренді). Після цього можлива інтерпретація та використання отриманих кластерів у подальшому аналізі.

Переваги кластеризації в аналізі даних:

- Виявлення структури в даних: Кластеризація дозволяє виділити природні структури або групи в наборі даних, що допомагає зрозуміти природу даних та їх взаємозв'язки.
- Спрощення аналізу даних: Кластеризація може спростити аналіз великих об'ємів даних, роблячи його більш зрозумілим та легко інтерпретованим.

- Пошук аномалій: Виділення кластерів може допомогти виявити аномалії або викиди в даних, оскільки вони можуть утворювати окремі групи.
- Маркетингові дослідження: Кластеризація використовується для сегментації аудиторії та визначення груп споживачів з подібними характеристиками.
- Управління даними: Допомагає впорядковувати та категоризувати дані, полегшуючи їх управління та аналіз.

Кластеризація є важливим інструментом у сфері аналізу даних, дозволяючи виявляти приховані патерни та структури, що можуть бути важливими для прийняття рішень у різних областях.

1.6 Інтелектуальний аналіз даних

Інтелектуальний аналіз даних (ІАД) — це область досліджень та практики, яка об'єднує техніки аналізу даних та методи штучного інтелекту (ШІ) з метою здійснення більш складного, глибшого та автоматизованого аналізу інформації. Цей підхід став надзвичайно важливим у зв'язку з зростанням обсягу та складності даних, які генеруються та зберігаються у різних галузях.

Основні аспекти інтелектуального аналізу даних:

- Машинне навчання (Machine Learning): Це галузь ШІ, яка вивчає алгоритми та моделі, які дозволяють комп'ютерам самостійно навчатися на основі даних та робити прогнози чи приймати рішення без явного програмування.
- Глибинне навчання (Deep Learning): Це підгалузь машинного навчання, яка використовує нейронні мережі з багатьма шарами (глибокими архітектурами) для вирішення завдань навчання та передбачення.
- Аналіз текстів та обробка природної мови (Natural Language Processing - NLP): Включає в себе розуміння та генерацію людської мови, що дозволяє аналізувати текстові дані та взаємодіяти з користувачами у природній мові.

- Інтелектуальна обробка зображень та відео (Computer Vision): Займається виявленням, розпізнаванням та інтерпретацією зображень та відео за допомогою комп'ютерних алгоритмів.
- Автоматизована обробка мови (Automated Speech Processing): Досліджує технології, які дозволяють комп'ютерам розуміти та обробляти людську мову у формі звукових сигналів.
- Інтеграція даних та великі дані (Big Data Integration): Включає в себе методи та технології для ефективного оброблення, аналізу та інтерпретації великих обсягів даних.

Переваги інтелектуального аналізу даних:

- Покращення точності та ефективності: ІАД дозволяє вдосконалити процес аналізу даних, роблячи його більш точним та ефективним.
- Автоматизація процесу прийняття рішень: Машинне навчання дозволяє автоматизувати прийняття рішень на основі аналізу даних та побудови прогнозів.
- Виявлення складних патернів та залежностей: Алгоритми глибинного навчання та інші методи ІАД дозволяють виявляти складні зв'язки та патерни в даних, які можуть бути важко виявити іншими методами.
- Розширення можливостей аналізу: Дозволяє аналізувати та використовувати дані в нових та більш продуктивних способах.
- Автоматизована обробка неструктурованих даних: Дозволяє аналізувати неструктуровані дані, такі як текст, зображення та відео.
- Інтелектуальний аналіз даних є важливою галуззю в сучасній аналітиці даних, оскільки він дозволяє використовувати весь потенціал сучасних алгоритмів та технік для зрозуміння та використання даних у багатьох сферах життя.

1.7 Метод головних компонент

Метод головних компонент (Principal Component Analysis, PCA) є статистичним методом зменшення розмірності даних, який використовується для перетворення оригінальних змінних в новий набір, який називається головними компонентами. Головні компоненти є лінійними комбінаціями оригінальних змінних, і вони вибираються таким чином, щоб зберегти якнайбільше дисперсії в даних.

Основні цілі методу головних компонент:

- Зменшення розмірності даних:
    - Проблема великої кількості ознак: В сучасних даних може бути велика кількість ознак, і багатовимірність може стати проблемою. PCA допомагає зменшити кількість ознак, зберігаючи при цьому більшість інформації.

- Виявлення структури та взаємозв'язків в даних:
    - Пошук головних напрямків: PCA визначає головні напрямки, або компоненти, які представляють максимальну дисперсію в даних. Це дозволяє виявити основні структури та зв'язки.

- Зниження кореляції між ознаками:
    - Ортогональність головних компонент: Головні компоненти вибираються так, щоб вони були ортогональні один одному. Це означає, що вони не корелюють між собою, що може поліпшити стабільність та ефективність моделей.

- Застосування у візуалізації:

- Візуалізація даних в двовимірному просторі: Зменшення розмірності даних до двох чи трьох головних компонент дозволяє візуалізувати дані у зручний спосіб.

- Застосування у компресії зображень:
  - Зменшення обсягу даних: PCA може використовуватися для стиснення зображень чи відео, зберігаючи при цьому важливу інформацію.
- Очищення даних від шуму:
  - Виділення суттєвої інформації: PCA дозволяє виділити суттєві компоненти та відокремити їх від шумових компонент.

- Застосування у машинному навчанні:
  - Підготовка даних для моделювання: Зменшення розмірності може поліпшити ефективність моделей машинного навчання та сприяти уникненню перенавчання.

Процес PCA:

- *Стандартизація даних*: Всі змінні стандартизуються, щоб мати одиничні середні та дисперсії.
- *Побудова коваріаційної матриці*: Визначається коваріаційна матриця для оцінки взаємозв'язків між змінними.
- *Розрахунок власних векторів та власних значень*: Визначаються головні компоненти.
- *Вибір головних компонент*: Вибираються перші k головних компонент, які відповідають найбільшим власним значенням.
- *Побудова нового простору ознак*: Дані проекціються на простір, що визначений обраними головними компонентами.

PCA є потужним інструментом для аналізу та зменшення розмірності даних у різних галузях, включаючи статистику, машинне навчання та обробку сигналів.

1.8 Висновки до розділу 1

Завдання аналізу показників успішності футбольних клубів є важливою задачею не тільки для самих команд і їх власників, а й для бізнесів, що ніяк не пов’язані з футболом. Можливість прорекламувати свою компанію за допомогою найпопулярнішої гри світу це дуже перспективна інвестиція

Але сліпе інвестування коштів в першу команду, що показує обрі результати в даний момент часу може принести більше збитків ніж прибутку та присутності в медіа. Тому є важливим ідентифікувати команди, що будуть продовжувати виступати на високому рівні для розширення бренду на якомога більшу кількість людей.

Також перспективною інвестицією будуть команди, що поки не показують позитивних результатів, але вже довгий час прогресують і за всіма показниками мали б бути успішнішими ніж в даний момент. Реклама за допомогою подібних клубів, як і інвестиції в них, можуть бути менш витратними, але при цьому показувати кращі результати.

Для того, щоб ідентифікувати подібні команди ми використаємо такі методи як візуалізація даних, метод головних компонент та дерево рішень, бо вони будуть найбільш придатними до статистичних даних команд

РОЗДІЛ 2 СПЕЦІАЛЬНИЙ

2.1 Постановка задачі

Проведемо аналіз виступів клубів, що беруть участь в Англійській Прем'єр Лізі, протягом останніх 5 сезонів і скористаємося отриманими даними для прогнозу змін результатів їх виступів в поточному сезоні

Для цієї роботи використаємо десктоп середовище розробки PyCharm та програмне забезпечення Deductor Studio Academic

Дані отримані в результаті парсингу в розділі 2.2 будуть вказані в Додатку В

2.2 Парсинг даних для аналізу

Достатньо детальна статистика щодо футбольних клубів в інтернеті, на жаль, недоступна для завантаження, тому дані для аналізу було вирішено отримати з сайту https://fbref.com/ методом парсингом.

Проводимо парсинг статистики команд за сезон 2019/2020:

```
import requests
from bs4 import BeautifulSoup
import pandas as pd


url = "https://fbref.com/en/comps/9/2019-2020/2021-2022-Premier-League-Stats"
response = requests.get(url)
response = response.content
eplstats = BeautifulSoup(response, 'html.parser')

stats_list1=[]
stats_list2=[]
stats_list3=[]
```

Необхідні дані містяться в різних таблицях, тому знаходимо кожну з них в коді сайту:

```
ol1 = eplstats.find('div')
ol1 = ol1.find('div', id='content')
ol = ol1.find('div', class_='table_wrapper tabbed')
ol = ol.find('div', id='switcher_results2019-202091')
ol = ol.find('div', id='div_results2019-202091_overall')
ol = ol.find('table')
ol = ol.find('tbody')
teams_stats = ol.find_all('tr')

ol2 = eplstats.find('div')
ol2 = ol2.find('div', id='content')
ol2 = ol2.find_all('div', class_='table_wrapper tabbed')[1]
```

```
ol22 = ol2.find_all('div')[7]
ol22 = ol22.find('table')
ol22 = ol22.find('tbody')
teams_stats2 = ol22.find_all('tr')

ol3 = ol2.find_all('div')[10]
ol3 = ol3.find('table')
ol3 = ol3.find('tbody')
teams_stats3 = ol3.find_all('tr')
```

Шукаємо ті стовпці даних, які необхідні нам для аналізу, в трьох різних таблицях та зберігаємо їх у списках:

```
for team_stats in teams_stats:
    team_name = team_stats.find('td', attrs={'data-stat':'team'})
    team_name = team_name.find('a').text
    team_games = team_stats.find('td', attrs={'data-stat':'games'}).text
    team_games = int(team_games)
    team_wins = team_stats.find('td', attrs={'data-stat':'wins'}).text
    team_wins = int(team_wins)
    team_wins = round((team_wins/team_games)*100, 3)
    team_ties = team_stats.find('td', attrs={'data-stat':'ties'}).text
    team_ties = int(team_ties)
    team_ties = round((team_ties/team_games)*100, 3)
    team_losses = team_stats.find('td', attrs={'data-stat':'losses'}).text
    team_losses = int(team_losses)
    team_losses = round((team_losses/team_games)*100, 3)
    team_gf = team_stats.find('td', attrs={'data-stat':'goals_for'}).text
    team_gf = int(team_gf)
    team_gf = round(team_gf/team_games, 3)
    team_ga = team_stats.find('td', attrs={'data-stat':'goals_against'}).text
    team_ga = int(team_ga)
    team_ga = round(team_ga/team_games, 3)
    team_gd = team_stats.find('td', attrs={'data-stat':'goal_diff'}).text
    team_gd = int(team_gd)
    team_gd = round(team_gd/team_games, 3)
    team_points = team_stats.find('td', attrs={'data-stat':'points'}).text
    team_points = int(team_points)
    team_points_p90 = round(team_points/team_games, 2)
    team_xgf = team_stats.find('td', attrs={'data-stat':'xg_for'}).text
    team_xgf = float(team_xgf)
    team_xgf = round(team_xgf/team_games, 3)
    team_xga = team_stats.find('td', attrs={'data-stat':'xg_against'}).text
    team_xga = float(team_xga)
    team_xga = round(team_xga/team_games, 3)
    team_xgd90 = team_stats.find('td', attrs={'data-stat':'xg_diff_per90'}).text
    team_xgd90 = float(team_xgd90)
    stats_list1.append([team_name, team_games, team_wins, team_ties,
team_losses, team_points, team_points_p90, team_gf, team_ga, team_gd, team_xgf,
team_xga, team_xgd90])

for team_stat in teams_stats2:
  team_name2 = team_stat.find('th', attrs={'data-stat':'team'})
  team_name2 = team_name2.find('a').text
  team_players_used = team_stat.find('td', attrs={'data-
stat':'players_used'}).text
  team_players_used = int(team_players_used)
  team_avg_age = team_stat.find('td', attrs={'data-stat':'avg_age'}).text
  team_avg_age = float(team_avg_age)
  team_possession = team_stat.find('td', attrs={'data-stat':'possession'}).text
  team_possession = float(team_possession)
  team_npG = team_stat.find('td', attrs={'data-stat':'goals_pens'}).text
  team_npG = float(team_npG)
  team_npG = round(team_npG/team_games, 3)
  team_pen_goals = team_stat.find('td', attrs={'data-stat':'pens_made'}).text
```

```
  team_pen_goals = int(team_pen_goals)
  team_pen_goals = round(team_pen_goals/team_games, 3)
  team_pen_taken = team_stat.find('td', attrs={'data-stat':'pens_att'}).text
  team_pen_taken = int(team_pen_taken)
  team_pen_taken = round(team_pen_taken/team_games, 3)
  team_npxg = team_stat.find('td', attrs={'data-stat':'npxg'}).text
  team_npxg = float(team_npxg)
  print(team_npxg)
  team_npxg = round(team_npxg/team_games, 3)
  team_prog_carr_for = team_stat.find('td', attrs={'data-
stat':'progressive_carries'}).text
  team_prog_carr_for = int(team_prog_carr_for)
  team_prog_carr_for = round(team_prog_carr_for/team_games, 3)
  team_prog_pass_for = team_stat.find('td', attrs={'data-
stat':'progressive_passes'}).text
  team_prog_pass_for = int(team_prog_pass_for)
  team_prog_pass_for = round(team_prog_pass_for/team_games, 3)
  team_yellow_cards_for = team_stat.find('td', attrs={'data-
stat':'cards_yellow'}).text
  team_yellow_cards_for = int(team_yellow_cards_for)
  team_yellow_cards_for = round(team_yellow_cards_for/team_games, 3)
  team_red_cards_for = team_stat.find('td', attrs={'data-
stat':'cards_red'}).text
  team_red_cards_for = int(team_red_cards_for)
  team_red_cards_for = round(team_red_cards_for/team_games, 3)
  stats_list2.append([team_name2, team_players_used, team_avg_age,
team_possession, team_npG, team_npxg, team_pen_goals, team_pen_taken,
team_prog_carr_for, team_prog_pass_for, team_yellow_cards_for,
team_red_cards_for])

for team_stat in teams_stats3:
  team_name3 = team_stat.find('th', attrs={'data-stat':'team'})
  team_name3 = team_name3.find('a').text
  team_name3 = team_name3[3:]
  team_npG_ag = team_stat.find('td', attrs={'data-stat':'goals_pens'}).text
  team npG ag = float(team npG ag)
  team_npG_ag = round(team_npG_ag / team_games, 3)
  team_pen_goals_ag = team_stat.find('td', attrs={'data-stat':'pens_made'}).text
  team_pen_goals_ag = int(team_pen_goals_ag)
  team_pen_goals_ag = round(team_pen_goals_ag / team_games, 3)
  team pen taken ag = team stat.find('td', attrs={'data-stat':'pens att'}).text
  team_pen_taken_ag = int(team_pen_taken_ag)
  team_pen_taken_ag = round(team_pen_taken_ag / team_games, 3)
  team_npxg_ag = team_stat.find_all('td')[17].text
  team_npxg_ag = float(team_npxg_ag)
  team_npxg_ag = round(team_npxg_ag / team_games, 3)
  team_prog_carr_ag = team_stat.find('td', attrs={'data-stat':
'progressive_carries'}).text
  team_prog_carr_ag = int(team_prog_carr_ag)
  team_prog_carr_ag = round(team_prog_carr_ag / team_games, 3)
  team_prog_pass_ag = team_stat.find('td', attrs={'data-stat':
'progressive_passes'}).text
  team_prog_pass_ag = int(team_prog_pass_ag)
  team_prog_pass_ag = round(team_prog_pass_ag / team_games, 3)
  team_yellow_cards_ag = team_stat.find('td', attrs={'data-
stat':'cards_yellow'}).text
  team_yellow_cards_ag = int(team_yellow_cards_ag)
  team_yellow_cards_ag = round(team_yellow_cards_ag / team_games, 3)
  team_red_cards_ag = team_stat.find('td', attrs={'data-stat':'cards_red'}).text
  team_red_cards_ag = int(team_red_cards_ag)
  team_red_cards_ag = round(team_red_cards_ag / team_games, 3)
  stats_list3.append([team_name3, team_npG_ag, team_pen_goals_ag,
team_pen_taken_ag ,team_npxg_ag, team_prog_carr_ag, team_prog_pass_ag,
team_yellow_cards_ag, team_red_cards_ag])
```

Конвертуємо списки в дата фрейми, робимо з них один дата фрейм та зберігаємо дані файлом в форматі csv:

```
stats_data1 = pd.DataFrame(stats_list1, columns = ['team', 'games', 'w', 't',
'l', 'p', 'p p90', 'gf', 'ga', 'gd', 'xg for', 'xg ag', 'xgd'])
stats_data2 = pd.DataFrame(stats_list2, columns = ['team', 'pl used', 'avg age',
'poss', 'nPg for', 'pS for', 'pA for', 'nPxg for', 'pr carr for', 'pr pass for',
'y cards for', 'r cards for'])
stats_data3 = pd.DataFrame(stats_list3, columns = ['team', 'nPg ag', 'pS ag',
'pA ag', 'nPxg ag', 'pr carr ag', 'pr pass ag', 'y cards ag', 'r cards ag'])

stats_data_merged = pd.merge(stats_data1, stats_data2, on='team', how='right')
stats_data_merged_again = pd.merge(stats_data_merged, stats_data3, on='team',
how='right')
stats_data_merged_again.to_csv('stats_data_19_20.csv')
```

Повторюємо ті самі дії для сторінок сайту, що містять інформацію про сезони 2020/21, 2021/22, 2022/23 та інформацію про останні на момент парсингу 13 матчів поточного сезону 2023/24

2.3 Створення таблиць з даними за 5 сезонів

Для подальшого аналізу нам необхідно буде утворити з отриманих даних дві нові таблиці: одну, яка буде містити дані за всі 5 сезонів, іншу з середніми значеннями кожного з показників за 5 сезонів.

Формат Англійської Прем'єр Ліги передбачає виліт трьох команд з найменшою кількістю очок за сезон, тому наші дві таблиці будуть містити тільки команди, що брали участь в усіх 5 сезонах

Формуємо таблицю з даними за всі 5 сезонів:

```
s19 = pd.read_csv ('stats_data_19_20.csv')
s20 = pd.read_csv ('stats_data_20_21.csv')
s21 = pd.read_csv ('stats_data_21_22.csv')
s22 = pd.read_csv ('stats_data_22_23.csv')
s23 = pd.read_csv ('stats_data_13_games.csv')

## Creating all-in-one pd DataFrame for 5 seasons

# Dropping teams which are not present in every table

indexTeam19 = s19[(s19['team'] != 'Arsenal')&(s19['team'] !=
'Villa')&(s19['team'] != 'Brighton')&(s19['team'] != 'Chelsea')&(s19['team'] !=
'Palace')&(s19['team'] != 'Everton')&(s19['team'] != 'Liverpool')
                &(s19['team'] != 'Man City')&(s19['team'] != 'Man
Utd')&(s19['team'] != 'Newcastle')&(s19['team'] != 'Wolves')&(s19['team'] !=
'West Ham')&(s19['team'] != 'Tottenham')].index
indexTeam20 = s20[(s20['team'] != 'Arsenal')&(s20['team'] !=
'Villa')&(s20['team'] != 'Brighton')&(s20['team'] != 'Chelsea')&(s20['team'] !=
```

```
'Palace')&(s20['team'] != 'Everton')&(s20['team'] != 'Liverpool')
                &(s20['team'] != 'Man City')&(s20['team'] != 'Man
Utd')&(s20['team'] != 'Newcastle')&(s20['team'] != 'Wolves')&(s20['team'] !=
'West Ham')&(s20['team'] != 'Tottenham')].index
indexTeam21 = s21[(s21['team'] != 'Arsenal')&(s21['team'] !=
'Villa')&(s21['team'] != 'Brighton')&(s21['team'] != 'Chelsea')&(s21['team'] !=
'Palace')&(s21['team'] != 'Everton')&(s21['team'] != 'Liverpool')
                &(s21['team'] != 'Man City')&(s21['team'] != 'Man
Utd')&(s21['team'] != 'Newcastle')&(s21['team'] != 'Wolves')&(s21['team'] !=
'West Ham')&(s21['team'] != 'Tottenham')].index
indexTeam22 = s22[(s22['team'] != 'Arsenal')&(s22['team'] !=
'Villa')&(s22['team'] != 'Brighton')&(s22['team'] != 'Chelsea')&(s22['team'] !=
'Palace')&(s22['team'] != 'Everton')&(s22['team'] != 'Liverpool')
                &(s22['team'] != 'Man City')&(s22['team'] != 'Man
Utd')&(s22['team'] != 'Newcastle')&(s22['team'] != 'Wolves')&(s22['team'] !=
'West Ham')&(s22['team'] != 'Tottenham')].index
indexTeam23 = s23[(s23['team'] != 'Arsenal')&(s23['team'] !=
'Villa')&(s23['team'] != 'Brighton')&(s23['team'] != 'Chelsea')&(s23['team'] !=
'Palace')&(s23['team'] != 'Everton')&(s23['team'] != 'Liverpool')
                &(s23['team'] != 'Man City')&(s23['team'] != 'Man
Utd')&(s23['team'] != 'Newcastle')&(s23['team'] != 'Wolves')&(s23['team'] !=
'West Ham')&(s23['team'] != 'Tottenham')].index

s19.drop(indexTeam19, inplace=True)
s19.index=range(len(s19))
s19 = s19[s19.columns[1:]]
s19i = s19.loc[:, ~s19.columns.isin(['games'])]
s19i.columns = s19i.columns.map(lambda x: str(x) + '_19')
s20.drop(indexTeam20, inplace=True)
s20.index=range(len(s20))
s20 = s20[s20.columns[1:]]
s20i = s20.loc[:, ~s20.columns.isin(['team','games'])]
s20i.columns = s20i.columns.map(lambda x: str(x) + '_20')
s21.drop(indexTeam21, inplace=True)
s21.index=range(len(s21))
s21 = s21[s21.columns[1:]]
s21i = s21.loc[:, ~s21.columns.isin(['team','games'])]
s21i.columns = s21i.columns.map(lambda x: str(x) + '_21')
s22.drop(indexTeam22, inplace=True)
s22.index=range(len(s22))
s22 = s22[s22.columns[1:]]
s22i = s22.loc[:, ~s22.columns.isin(['team','games'])]
s22i.columns = s22i.columns.map(lambda x: str(x) + '_22')
s23.drop(indexTeam23, inplace=True)
s23.index=range(len(s23))
s23 = s23[s23.columns[1:]]
s23i = s23.loc[:, ~s23.columns.isin(['team','games'])]
s23i.columns = s23i.columns.map(lambda x: str(x) + '_23')


s19_23 = s19i
s19_23 = pd.merge(s19_23,s20i, left_index=True, right_index=True)
s19_23 = pd.merge(s19_23,s21i, left_index=True, right_index=True)
s19_23 = pd.merge(s19_23,s22i, left_index=True, right_index=True)
s19_23 = pd.merge(s19_23,s23i, left_index=True, right_index=True)
```

Формуємо таблицю з даними про середні показники статистики за 5 сезонів:

```
s19_23avg=s19

s19_23avg['w_avg'] = round((s19['w']*38/100 + s20['w']*38/100 + s21['w']*38/100
+ s22['w']*38/100 + s23['w']*13/100)/(38*4+13),3)
```

```
s19_23avg['t_avg'] = round((s19['t']*38/100+ s20['t']*38/100 + s21['t']*38/100 +
s22['t']*38/100 + s23['t']*13/100)/(38*4+13),3)
s19_23avg['l_avg'] = round((s19['l']*38/100 + s20['l']*38/100 + s21['l']*38/100
+ s22['l']*38/100 + s23['l']*13/100)/(38*4+13),3)
s19_23avg['p_p90_avg'] = round(((s19['p p90'] + s20['p p90']+ s21['p p90']+
s22['p p90'])*38+ s23['p p90']*13)/(38*4+13),3)
s19_23avg['gf_avg'] = round(((s19['gf'] + s20['gf'] + s21['gf'] + s22['gf'])*38+
s23['gf']*13)/(38*4+13),3)
s19_23avg['ga_avg'] = round(((s19['ga'] + s20['ga'] + s21['ga'] + s22['ga'])*38+
s23['ga']*13)/(38*4+13),3)
s19_23avg['gd_avg'] = round(((s19['gd'] + s20['gd'] + s21['gd'] + s22['gd'])*38+
s23['gd']*13)/(38*4+13),3)
s19_23avg['xg_for_avg'] = round(((s19['xg for']+ s20['xg for']+ s21['xg for'] +
s22['xg for'])*38+ s23['xg for']*13)/(38*4+13),3)
s19_23avg['xg_ag_avg'] = round(((s19['xg ag'] + s20['xg ag']+ s21['xg ag'] +
s22['xg ag'])*38+ s23['xg ag']*13)/(38*4+13),3)
s19_23avg['xgd_avg'] = round(((s19['xgd'] + s20['xgd'] + s21['xgd']+
s22['xgd'])*38 + s23['xgd']*13)/(38*4+13),3)
s19_23avg['pl_used_avg'] = round((s19['pl used']+ s20['pl used']+ s21['pl
used']+ s22['pl used']+ s23['pl used'])/5,3)
s19_23avg['avg_age_avg'] = round((s19['avg age']+ s20['avg age']+ s21['avg
age']+ s22['avg age']+ s23['avg age'])/5,3)
s19_23avg['poss_avg'] = round((s19['poss']+ s20['poss']+ s21['poss']+
s22['poss']+ s23['poss'])/5,3)
s19_23avg['nPg_for_avg'] = round(((s19['nPg for']+ s20['nPg for']+ s21['nPg
for']+ s22['nPg for'])*38+ s23['nPg for']*13)/165,3)
s19_23avg['nPxg_for_avg'] = round((s19['nPxg for']+ s20['nPxg for']+ s21['nPxg
for']+ s22['nPxg for'])*38+ s23['nPxg for']*13)/(38*4+13),3)
s19_23avg['pA_for_avg'] = round((s19['pA for']+ s20['pA for']+ s21['pA for']+
s22['pA for'])*38+ s23['pA for']*13)/(38*4+13),3)
s19_23avg['pS_for_avg'] = round((s19['pS for']+ s20['pS for']+ s21['pS for']+
s22['pS for'])*38+ s23['pS for']*13)/(38*4+13),3)
s19_23avg['pr_carr_for_avg'] = round(((s19['pr carr for']+ s20['pr carr for']+
s21['pr carr for']+ s22['pr carr for'])*38 + s23['pr carr for']*13)/(38*4+13),3)
s19_23avg['pr_pass_for_avg'] = round(((s19['pr pass for']+ s20['pr pass for']+
s21['pr pass for']+ s22['pr pass for'])*38 + s23['pr pass for']*13)/(38*4+13),3)
s19_23avg['y_cards_for_avg'] = round(((s19['y cards for']+ s20['y cards for']+
s21['y cards for']+ s22['y cards for'])*38 + s23['y cards for']*13)/(38*4+13),3)
s19_23avg['r_cards_for_avg'] = round(((s19['r cards for']+ s20['r cards for']+
s21['r cards for']+ s22['r cards for'])*38 + s23['r cards for']*13)/(38*4+13),3)
s19 23avg['nPg ag avg'] = round(((s19['nPg ag']+ s20['nPg ag']+ s21['nPg ag']+
s22['nPg ag'])*38 + s23['nPg ag']*13)/(38*4+13),3)
s19_23avg['nPxg_ag_avg'] = round(((s19['nPxg ag']+ s20['nPxg ag']+ s21['nPxg
ag']+ s22['nPxg ag'])*38+ s23['nPxg ag']*13)/(38*4+13),3)
s19_23avg['pA_ag_avg'] = round(((s19['pA ag']+ s20['pA ag']+ s21['pA ag']+
s22['pA ag'])*38+ s23['pA ag']*13)/(38*4+13),3)
s19_23avg['pS_ag_avg'] = round(((s19['pS ag']+ s20['pS ag']+ s21['pS ag']+
s22['pS ag'])*38+ s23['pS ag']*13)/(38*4+13),3)
s19_23avg['pr_carr_ag_avg'] = round(((s19['pr carr ag']+ s20['pr carr ag']+
s21['pr carr ag']+ s22['pr carr ag'])*38 + s23['pr carr ag']*13)/(38*4+13),3)
s19_23avg['pr_pass_ag_avg'] = round(((s19['pr pass ag']+ s20['pr pass ag']+
s21['pr pass ag']+ s22['pr pass ag'])*38 + s23['pr pass ag']*13)/(38*4+13),3)
s19_23avg['y_cards_ag_avg'] = round(((s19['y cards ag'] + s20['y cards ag'] +
s21['y cards ag'] + s22['y cards ag'])*38 + s23['y cards ag']*13)/(38*4+13),3)
s19_23avg['r_cards_ag_avg'] = round(((s19['r cards ag'] + s20['r cards ag'] +
s21['r cards ag'] + s22['r cards ag'])*38 + s23['r cards ag']*13)/(38*4+13),3)

print(s19_23avg)

s19_23avg=s19_23avg.drop(s19_23avg.columns[1:32],axis=1)
print(s19_23avg)

s19_23avg.to_csv('Seasons19-23 Average stats.csv')
```

2.4 Візуалізація отриманих даних в вигляді графіків

2.4.1 Створення точкових діаграм за середніми показниками

Створюємо точкову діаграму для середніх значень статистичних показників кожної з 13 команд, що брали участь у всіх 5 сезонах, що розглядаються.

```
s19 = pd.read_csv ('stats_data_19_20.csv')
s20 = pd.read_csv ('stats_data_20_21.csv')
s21 = pd.read_csv ('stats_data_21_22.csv')
s22 = pd.read_csv ('stats_data_22_23.csv')
s23 = pd.read_csv ('stats_data_13_games.csv')
s19_23 = pd.read_csv ('Seasons19-23 Average stats.csv')

s23['g_xg for diff'] = s23['gf'] - s23['xg for']
s23['g_xg ag diff'] = s23['ga'] - s23['xg ag']
s22['g_xg for diff'] = s22['gf'] - s22['xg for']
s22['g_xg ag diff'] = s22['ga'] - s22['xg ag']
s21['g_xg for diff'] = s21['gf'] - s21['xg for']
s21['g_xg ag diff'] = s21['ga'] - s21['xg ag']
s20['g_xg for diff'] = s20['gf'] - s20['xg for']
s20['g_xg ag diff'] = s20['ga'] - s20['xg ag']
s19['g_xg for diff'] = s19['gf'] - s19['xg for']
s19['g_xg ag diff'] = s19['ga'] - s19['xg ag']
s19_23['g_xg_for_diff'] = s19_23['gf_avg'] - s19_23['xg_for_avg']
s19_23['g_xg_ag_diff'] = s19_23['ga_avg'] - s19_23['xg_ag_avg']

fig, xgfa = plt.subplots(figsize=(23, 23))
colors = {'Arsenal':'#EF0107', 'Villa':'#95BFE5', 'Bournemouth':'#B50E12',
'Brighton':'#0057B8', 'Burnley':'#6C1D45', 'Chelsea':'#034694',
          'Palace':'#1B458F', 'Everton':'#003399', 'Liverpool':'#C8102E', 'Man
City':'#6CABDD', 'Man Utd':'#EF3829', 'Newcastle':'#241F20',
          'Fulham':'#3c3c3c', 'Wolves':'#FDB913', 'West Ham':'#7A263A',
'Tottenham':'#132257', 'Sheffield':'#000000', 'Forest':'#913831',
          'Luton':'#ff9a00', 'Brentford':'#811331', 'Leicester City':'#0057B8',
'Southampton':'#913831', 'Norwich City':'#FDB913', 'Watford':'#000000', 'Leeds
United':'#0057B8', 'West Brom':'#003399'}

size1923=s19_23['p_p90_avg']*165

scatter = xgfa.scatter( s20['g_xg ag diff']*38, s20['g_xg for diff']*38, color=[
colors[i] for i in s20['team'] ], sizes=s20['p']*30)


tm = dict(prop="sizes", num=4.5, color=scatter.cmap(0.7), fmt="{x} оч.",
          func=lambda s: s/30)
legend2 = xgfa.legend(*scatter.legend_elements(**tm),
                      loc="lower right", title="Очки", markerscale=0.6,
bbox_to_anchor=(1.12, 0.06), columnspacing=20, draggable=True, fontsize=14,
shadow=True, title_fontsize=16, handleheight=2)

for idx, row in s20.iterrows():
    xgfa.annotate((row['team'],row['p']), (row['g_xg ag diff']*38, row['g_xg for
diff']*38), xytext=(-16, 23), textcoords='offset points', size=11)


plt.title('Вплив ступіню оверперформансу команд на їх кількість очок, сезон
2020/21', size=22, y=1.05)
plt.ylabel('Різниця між фактично забитими та очікуваними забитими', size=16)
plt.xlabel('Різниця між фактично пропущеними та очікуваними пропущеними',
size=16)
```

```
plt.grid()
plt.savefig('Overperformance influence on results 2020.jpg', dpi=500)

plt.show()
```

Точкова діаграма, яка показує по свої осях кількість пропущених і забитих командами м’ячів за останні 5 сезонів, де розмір точки відображує кількість очок, зароблених за цей час:

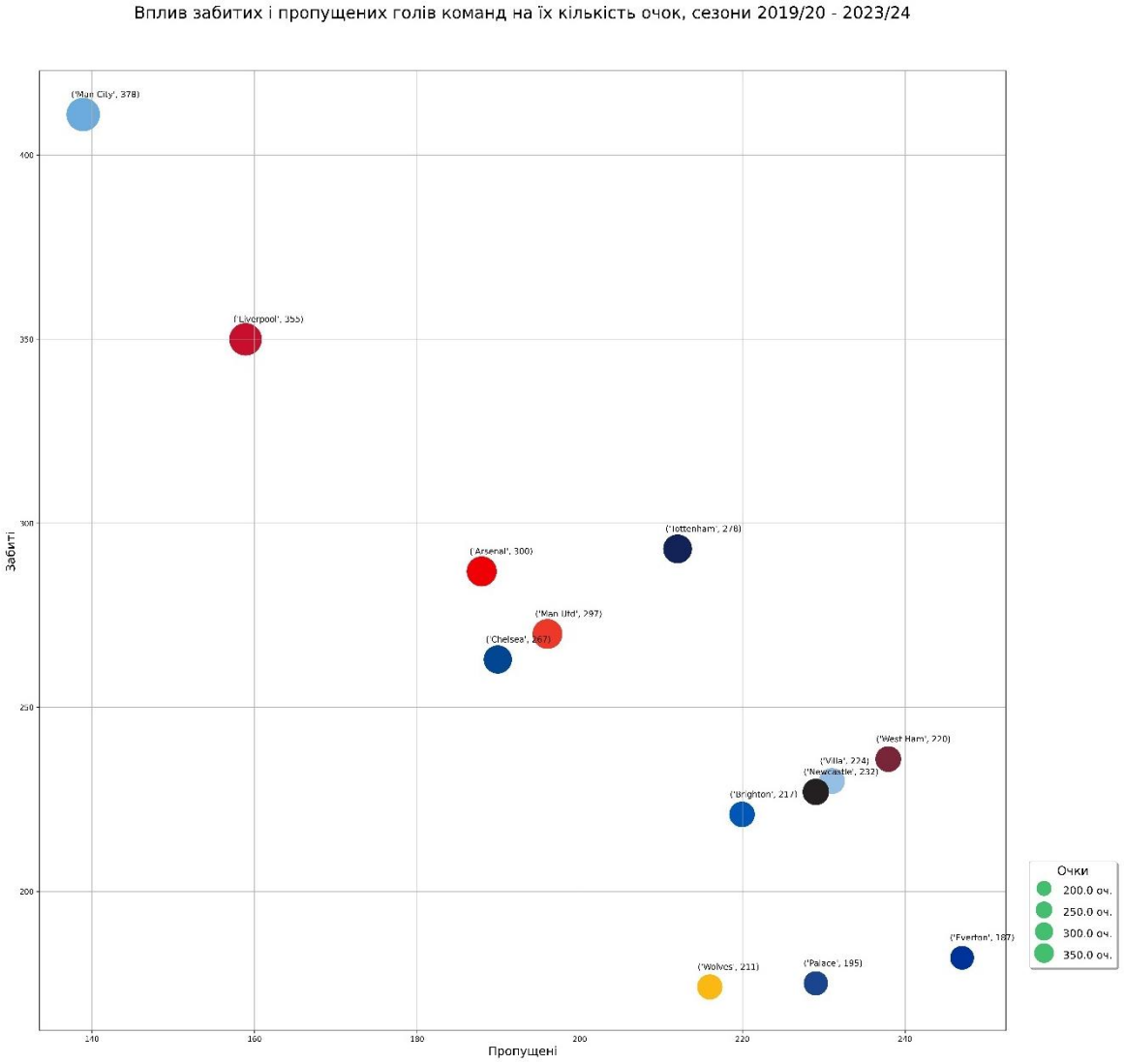


Рисунок 1 Вплив забитих і пропущених голів за 5 сезонів

Точкова діаграма, яка показує по свої осях кількість очікуваних пропущених і очікуваних забитих командами м’ячів за останні 5 сезонів, де розмір точки відображує кількість очок, зароблених за цей час:

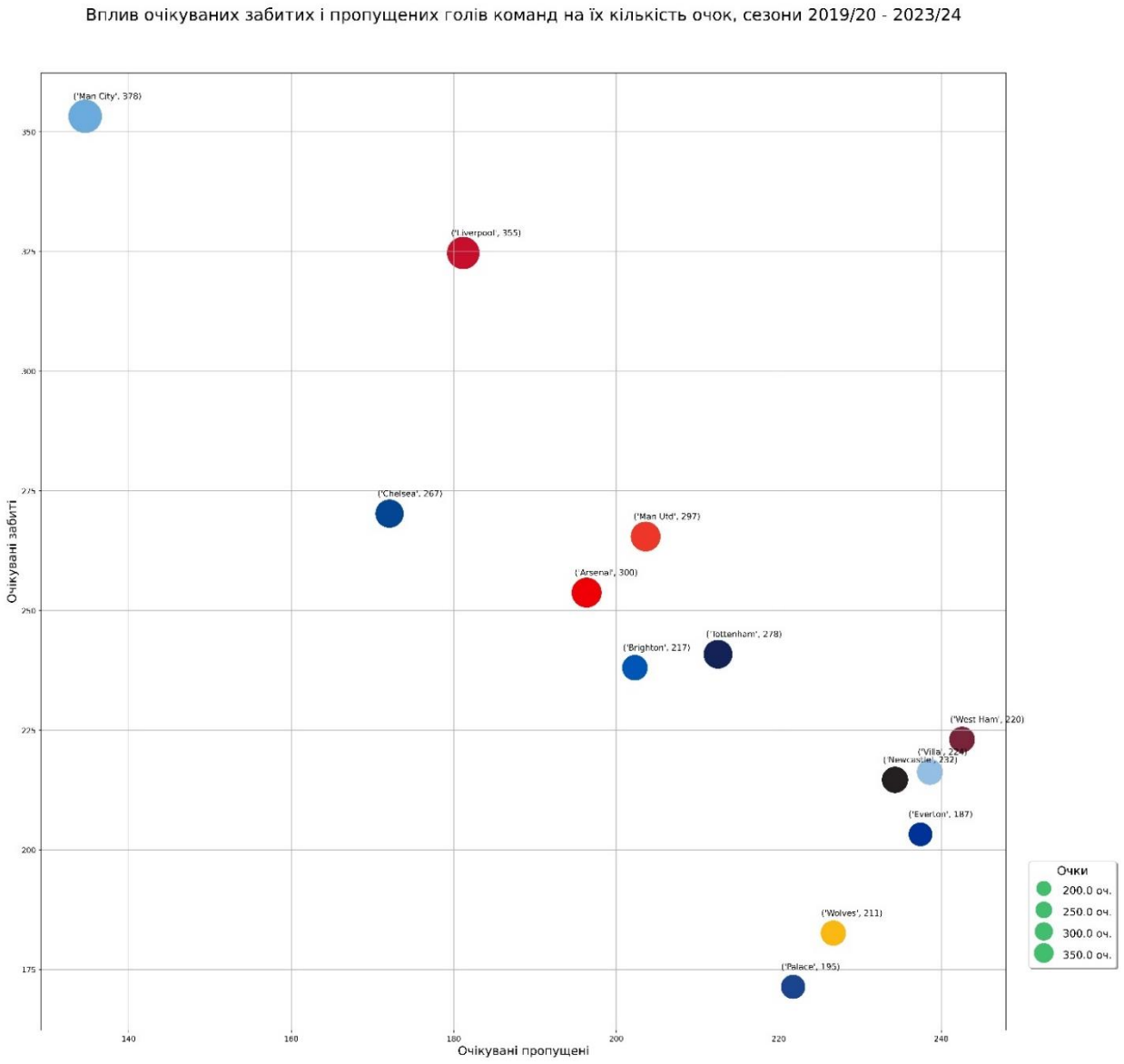


Рисунок 2 Вплив очікуваних забитих і очікуваних пропущених голів за 5 сезонів

Точкова діаграма, яка показує по свої осях різницю між фактично пропущеними голами та очікуваними пропущеними і різницю між фактично забитими голами та очікуваними забитими м’ячами за останні 5 сезонів, де розмір точки відображує кількість очок, зароблених за цей час. Вона відображає тенденцію команд до оверперформансу (більша результативність виступів, ніж очікувалася згідно зі статистичними даними) або андерперформансу (менша результативність виступів, ніж очікувалася згідно зі статистичними даними)

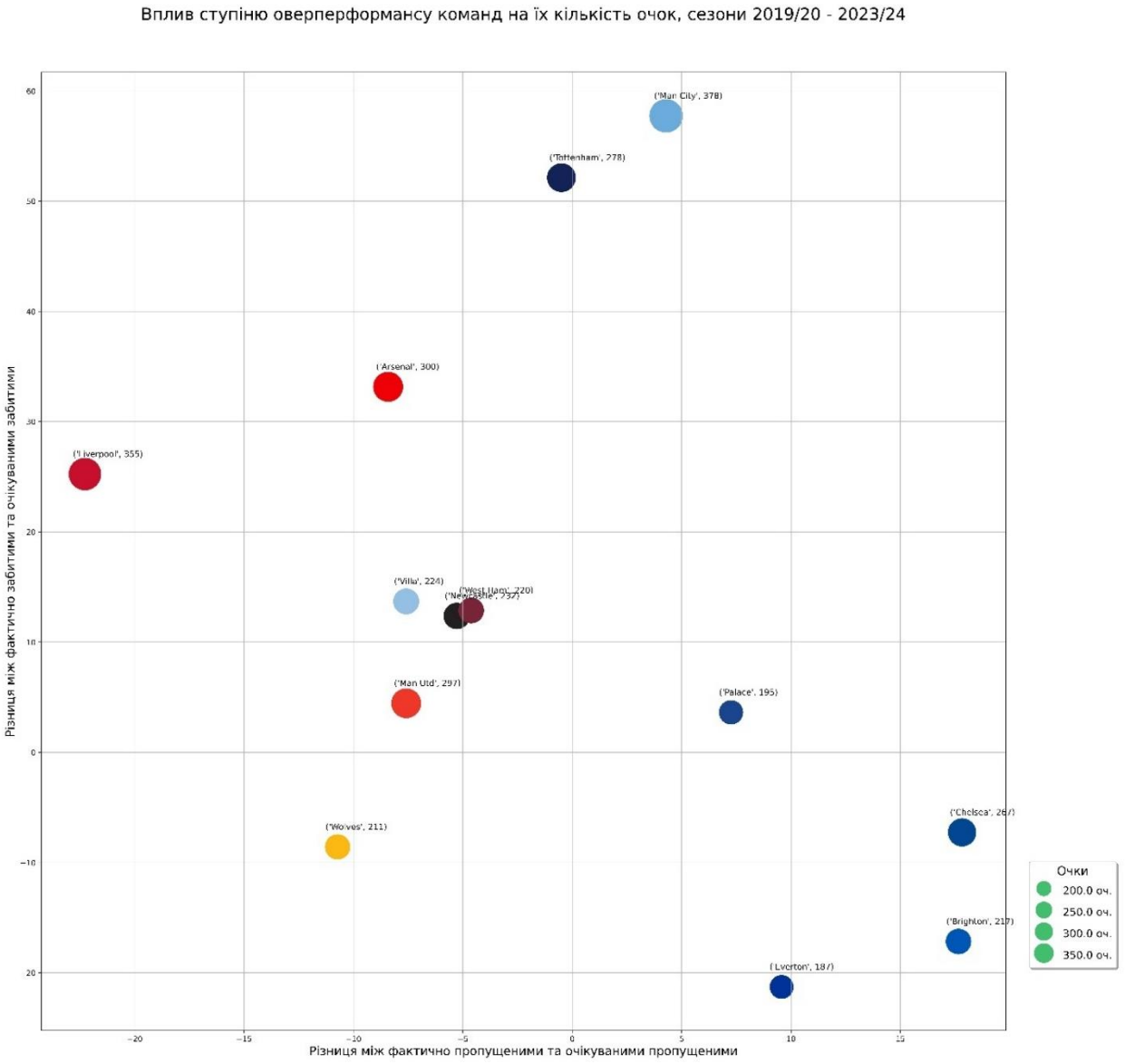


Рисунок 3 Вплив різниці між фактичними та очікуваними голами за 5 сезонів

2.4.2 Аналіз створених діаграм

За першими двома діаграмами можемо кластеризувати команди на такі групи:

- Лідери (верхній лівий кут): Man City, Liverpool
- Команди зі стабільно хорошими показниками: Arsenal, Tottenham, Chelsea, Man Utd
- Середняки та аутсайдери: Brighton, Palace, West Ham, Villa, Newcastle, Wolves, Everton

Але під час даної кластеризації ми стикаємося з дилемою – до якої саме групи віднести Brighton. За фактичними результатами ця команда має бути звичайним середняком, але ось згідно з очікуваними результатами команда має бути в числі команд, що видають стабільно хороші результати і знаходяться в числі найкращих. Більш чітко дану розбіжність можна побачити на третій діаграмі – Брайтон це команда з найбільшим ступенем андерперформансу в лізі, вони пропустили за цей час на 17 голів більше ніж мали б і забили на 18 голів менше ніж мали б.

Також за третьою діаграмою можемо кластеризувати команди на такі групи:

- Команди, що стабільно показують оверперформанс: Man City, Liverpool, Arsenal, Tottenham
- Команди, що виступають на тому рівні, на якому мали б: Palace, West Ham, Villa, Newcastle, Wolves
- Команди, що стабільно показують андерперформанс: Everton, Brighton, Chelsea

З отриманої інформації ми можемо зробити висновок, що Everton, Brighton та Chelsea це команди, які в найближчих сезонах претендують на зміну своєї групи на більш високу. Але враховуючи дуже нестабільний період і зміну

власника в Chelsea та фінансові проблеми Everton, найбільший потенціал для зміни групи має все ж Brighton. Ця команда в сезоні 2022/23 вже показала цей потенціал фінішувавши на 6 місці в чемпіонаті з 20. Для цього їй потрібно було в рамках даного сезону перейти до групи команд, що виступають на очікуваному рівні (різниця забитих за сезон: -1.3, різниця пропущених за сезон: +2.8)

2.4.3 Створення точкових діаграм за показниками поточного сезону

Створюємо точкові діаграми за тим самим принципом але вже використовуючи дані тільки за поточний сезон

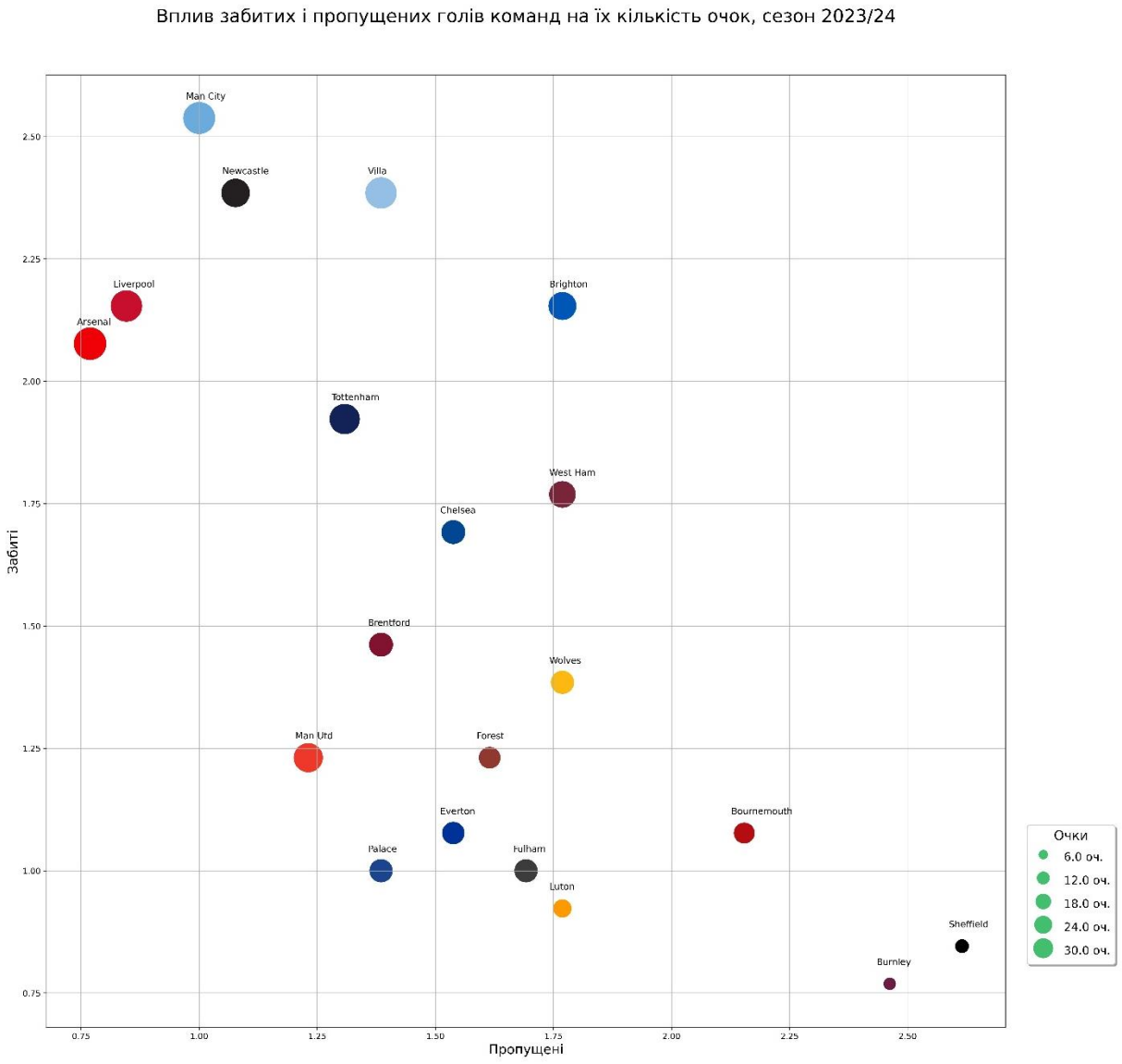


Рисунок 4 Вплив забитих і пропущених голів за сезон 2023/24

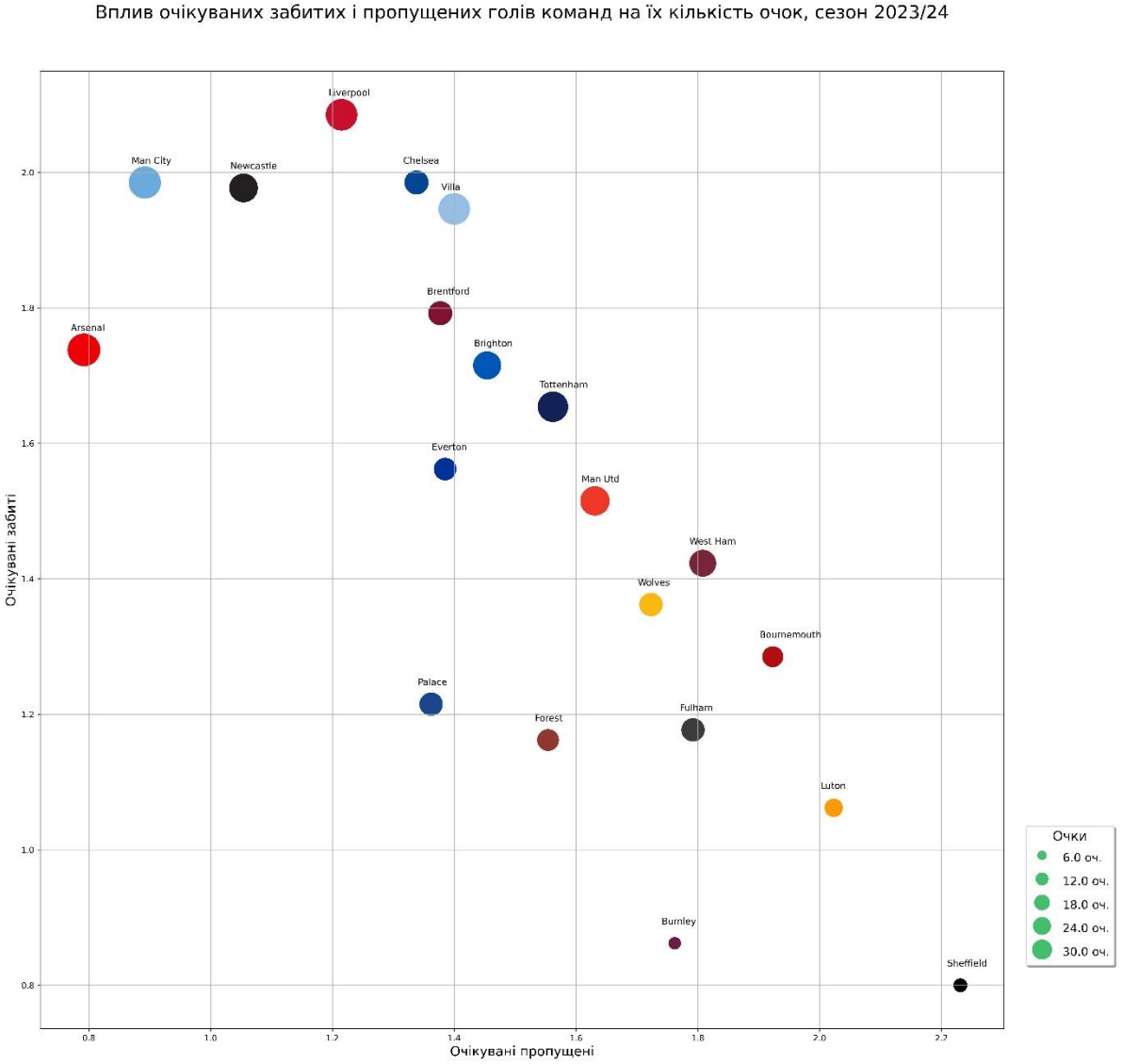


Рисунок 5 Вплив очікуваних забитих і очікуваних пропущених голів за сезон 2023/24

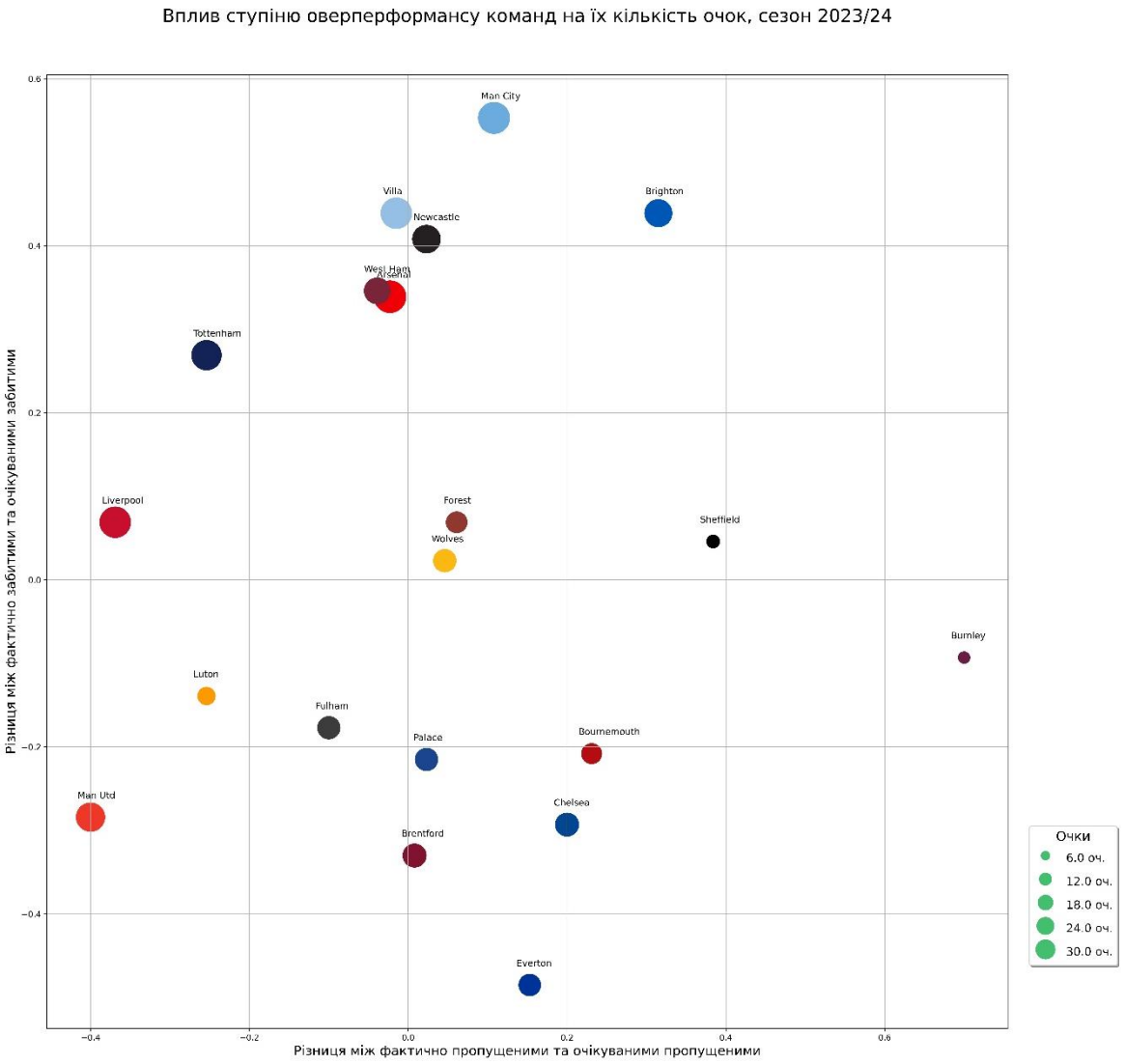


Рисунок 6 Вплив вплив різниці між фактичними та очікуваними голами за сезон 2023/24

2.4.4 Аналіз створених діаграм

За першими двома діаграмами можемо кластеризувати команди на такі групи:

- Лідери (верхній лівий кут): Man City, Liverpool, Arsenal, Newcastle, Villa
- Претенденти на верхню частину таблиці: Brighton, Tottenham, Chelsea, Brentford
- Середняки: Palace, West Ham, Wolves, Everton, Man Utd, Forest, Fulham, Bournemouth
- Аутсайдери: Sheffield, Burnley, Luton

Можемо побачити, що за забитими та пропущеними голами West Ham та Chelsea обидві знаходяться на одному рівні претендентів на верхню частину таблиці. Але поглянувши на другу діаграму стає ясно, що West Ham є все ж середняком, чиє високе місце є результатом яскравого початку сезону , яке вони не зуміли втримати та почали спускатися все нижче за таблицею. В той же час Chelsea має бути в числі лідерів чемпіонату, але їх нестабільність, що пов’язана з нещодавньою зміною керівництва і приходом до команди більше 10 нових гравців за останні півтора року, призводить до того, що фактично вони зараз знаходяться в середині таблиці за очками. Також Bournemouth та Everton показують результати набагато гірше, ніж мали би.

За третьою діаграмою можемо кластеризувати команди на такі групи:

- Команди, що мають великий ступінь оверперформансу: Man City, Liverpool, Arsenal, Tottenham, Newcastle, Villa, Tottenham
- Команди, що виступають приблизно на тому рівні, на якому мали б: Man Utd, Luton, Fulham, Forest, West Ham, Brighton, Sheffield, Palace
- Команди, що показують андерперформанс: Bournemouth, Chelsea, Everton, Brentford, Burnley

Факт досить великого адерперформансу Everton та Bournemouth може значити те, що нинішня їх позиція внизу таблиці в майбутньому зміниться, бо обидві команди забивають менше голів ніж мали б і пропускають більше ніж мали б

Хоча Burnley і показує андерперформанс це не призводить до думки, що ситуація для них може змінитися, бо за очікуваними голами вони все ще в групі аутсайдерів.

West Ham не є командою, яка показує стабільний оверперформанс, тому нинішнє її високе положення положення в лізі є аномалією і команда в найближчий час має повернутися до місця, яке має займати згідно зі своєю групою.

В той же час Newcastle показують гарний прогрес починаючи з середини сезону 2021/2022, а Villa починаючи з минулого сезону, тому їх оперперформанс не дає нам можливості прогнозувати майбутній кардинальний спад в результатах, хоч команди історічно і відносяться до групи тих, що виступають на очікуваному статистикою рівні.

2.4.5 Створення лінійного графіку показників прогресуючих команд

Побудуємо лінійний графік, що буде показувати кількість очок зароблених в середньому за матч протягом останніх 5 років, при цьому розглянемо 5 найбільш прогресуючих команд ліги:

```
N_results=pd.read_csv ('2019-23 all years data.csv')
V_results=pd.read_csv ('2019-23 all years data.csv')
A_results=pd.read_csv ('2019-23 all years data.csv')

N_results_Index = N_results[(N_results['team'] != 'Newcastle')].index
N_results.drop(N_results_Index, inplace=True)
print(N_results, V_results, A_results)

V_results_Index = V_results[(V_results['team'] != 'Villa')].index
V_results.drop(V_results_Index, inplace=True)


A_results_Index = A_results[(A_results['team'] != 'Arsenal')].index
A_results.drop(A_results_Index, inplace=True)

N=[N_results['p p90_19'], N_results['p p90_20'], N_results['p p90_21'],
N_results['p p90_22'], N_results['p p90_23']]
V=[V_results['p p90_19'], V_results['p p90_20'], V_results['p p90_21'],
V_results['p p90_22'], V_results['p p90_23']]
A=[A_results['p p90_19'], A_results['p p90_20'], A_results['p p90_21'],
A_results['p p90_22'], A_results['p p90_23']]
seasons=['2019/20', '2020/21', '2021/22', '2022/23', '2023/24']
print(N, V, A)
plt.figure(figsize=(20,15))

plt.title('Результати команд протягом сезонів 2019/20 - 2023/24', fontsize=20)

plt.plot(seasons, N, '.-', color='#241F20')
plt.plot(seasons, V, '.-', color='#95BFE5')
plt.plot(seasons, A, '.-', color='#EF0107')
plt.xticks(seasons)
plt.ylabel('Очки за 90 хвилин', fontsize=16)
plt.xlabel('Сезон', fontsize=16)
plt.grid()
plt.legend(['Newcastle', 'Villa', 'Arsenal'], fontsize=12)
plt.savefig('5 years Results.jpg', dpi=500)
plt.show()
```

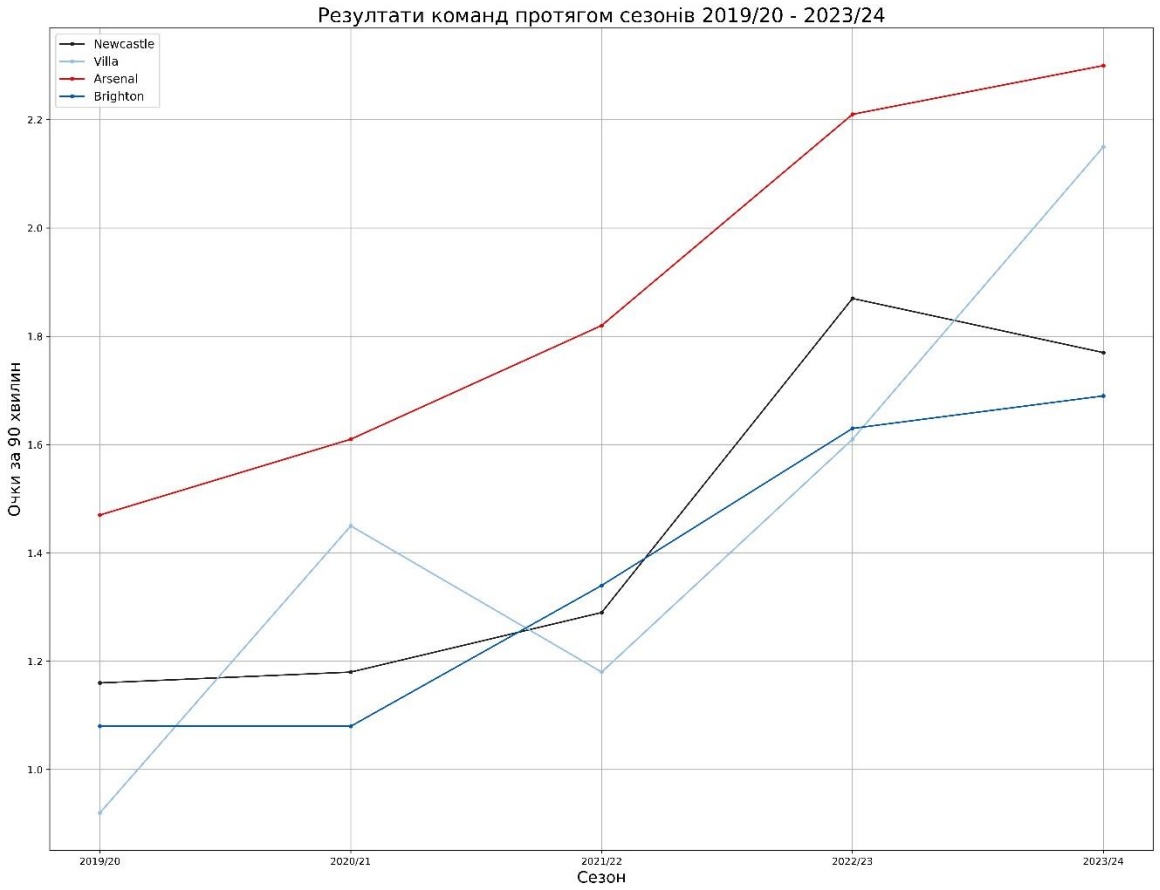


Рисунок 7 Покращення результатів команд протягом 5 сезонів

Отримуємо графік, який є доказом того, що оверперформанс таких команд, як Villa і Newcastle в цьому році не є аномалією, бо ці дві команди показують досить стабільний ріст останні 5 років. Можна помітити аномальність сезону 2021/2022, коли на чолі Villa був недосвідчений тренер Стівен Джерард, але прихід тренера Унаї Емері в сезоні 2021/22 повернув команду на шлях прогресу

2.5 Метод головних компонент

Проведемо аналіз середніх показників команд за всі 5 сезонів за допомогою методу головних компонент:

```
import sys
import pandas
import pylab as pl
from sklearn import preprocessing
from sklearn.decomposition import PCA


def main():
    """Load data."""
    try:
        csvfile1 = 'stats_data_13_games.csv'
        csvfile2 = 'stats_data_22_23.csv'
        csvfile3 = 'stats_data_21_22.csv'
        csvfile4 = 'stats_data_20_21.csv'
        csvfile5 = 'stats_data_19_20.csv'
        csvfile6 = 'Seasons19-23 Average stats1.csv'
    except IndexError:
        print
        '%s\n\nUsage: %s [--3d] <csv_file>' % (__doc__, sys.argv[0])
        return
    data = pandas.read_csv(csvfile6, index_col=(0, 1))

    # first column provides labels
    ylabels = [a for a, _ in data.index]
    labels = [text for _, text in data.index]
    encoder = preprocessing.LabelEncoder().fit(ylabels)

    xdata = data.values
    ydata = encoder.transform(ylabels)
    target_names = encoder.classes_
    plotpca(xdata, ydata, target_names, labels, csvfile1)


def plotpca(xdata, ydata, target_names, items, filename):
    """Make plot."""
    pca = PCA(n_components=2)
    components = pca.fit(xdata).transform(xdata)

    # Percentage of variance explained for each components
    print('explained variance ratio (first two components):',
          pca.explained_variance_ratio_)

    pl.figure(figsize=(21, 21))  # Make a plotting figure
    pl.subplots_adjust(bottom=0.1)

    # NB: a maximum of 7 targets will be plotted
    for i, (c, m, target_name) in enumerate(zip(
            'krbmycgkrmycgbkrmycb', 'ooooooooooooooooo', target_names)):
        pl.scatter(components[ydata == i, 0], components[ydata == i, 1],
                   color=c, marker=m, label=target_name)
        for n, x, y in zip(
                (ydata == i).nonzero()[0],
                components[ydata == i, 0],
                components[ydata == i, 1]):
            pl.annotate(
                items[n],
                xy=(x, y),
                xytext=(5, 5),
```

```
                textcoords='offset points',
                color=c,
                fontsize='large',
                ha='left',
                va='top')

    pl.title('Метод головних компонент для середніх значень всіх сезонів',
fontsize=22)
    pl.xlabel('Головна компонента 1', fontsize=18)
    pl.ylabel('Головна компонента 2', fontsize=18)
    pl.savefig ('19-23/PCA for 2019-2023.jpeg', dpi=500)
    pl.show()

main()
```

В результаті отримуємо значення першої компоненти: 0.95870169 і другої компоненти: 0.01591216 та точкову діаграму:

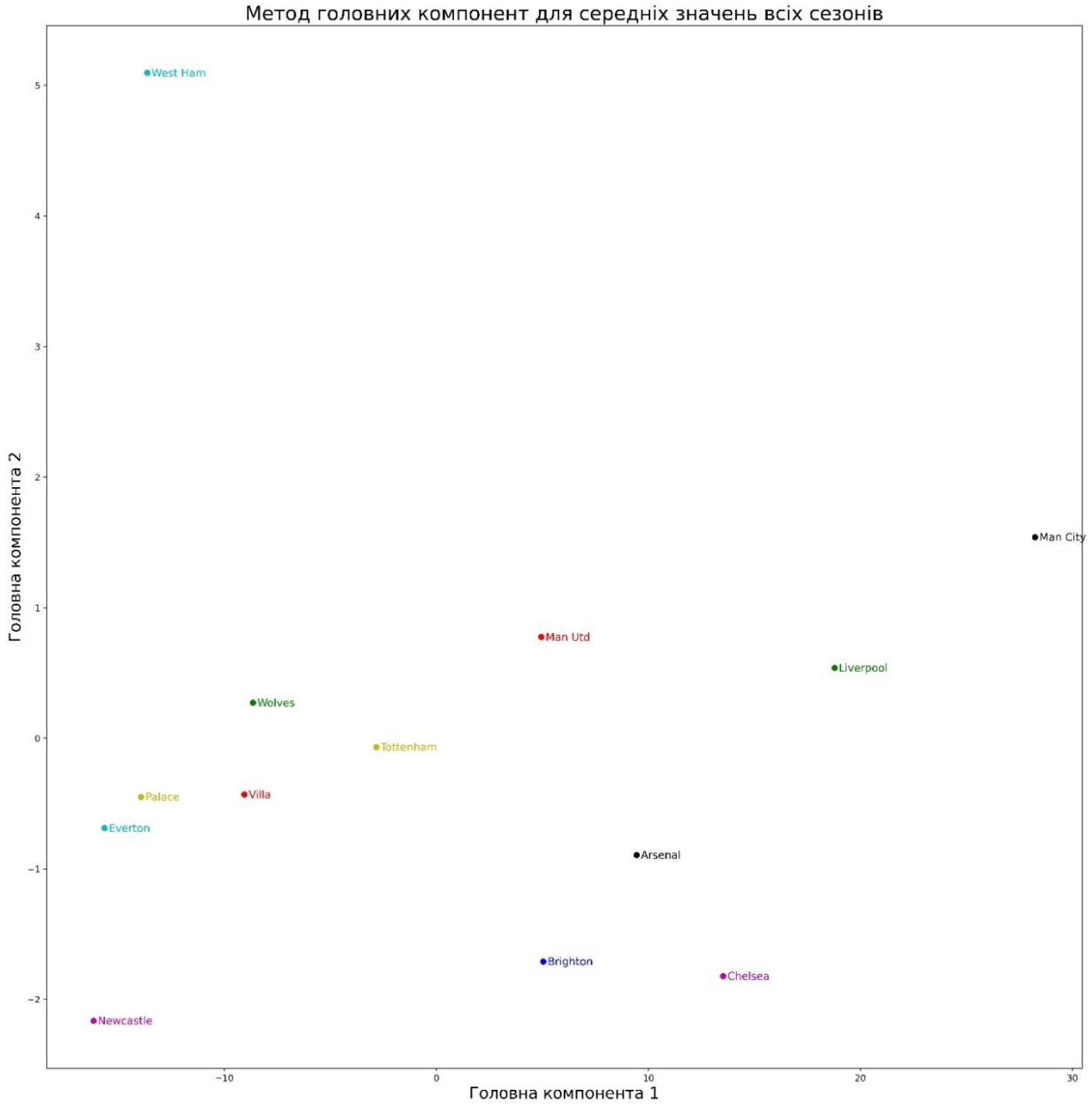


Рисунок 8 Аналіз методом головних компонент

Враховуючи настільки велику різницю між значимістю компонент можемо вважати, що перша компонента прямо відображує умовну силу кожної з команд за цей період часу. Згідно з діаграмою можемо знову кластеризувати команди на такі групи:

- Лідери: Man City
- Команди зі стабільно хорошими показниками: Arsenal, Chelsea, Man Utd, Brighton, Liverpool
- Середняки та аутсайдери: Palace, West Ham, Villa, Newcastle, Wolves, Everton, Tottenham

Можемо зробити висновок, що Liverpool все ж занадто далеко знаходиться від Man City, тому Man City є одноосібним лідером серед команд чемпіонату.

В той же час Tottenham та Brighton міняються групами. Причиною цьому може бути стабільний ріст середньої кількості очок з боку Brighton не зважаючи на історичний андерперформанс. В той час, коли Tottenham стабільно показує оверперформанс, але інші показники гри, такі як паси з просуванням та проходи з просуванням, все ж вказують на те, що команда показує рузльтати краще ніж мала б.

Такі команди, як Villa та Newcastle все ще знаходяться у групі середняків, не зважаючи на чудові результати в цьому та минулому роках. Причиною цього може бути занадто різкий ріст результатів команд, який обумовлено новим керівництвом з великою кількістю грошей в Newcastle, і прихід нового тренеру в Villa, якому дали можливість будувати команду без особливих обмежень і тиску з боку керівництва клубу.

2.6 Побудова дерева рішень

Перевіримо якість наших даних з таблиці середніх значень за 5 сезонів в програмі Deductor Studio Academic. Для цього треба розглянути їх на факт наявності пропущених даних, викидів та екстремальних значень:

№	Столбец	Тип данных	Вид данных	Пропуски		Выбросы		Экстремальные		Кол-во уника...	Качество данных	Резюме
				Кол-во	Действие	Кол-во	Действие	Кол-во	Действие			
1	team	ab Строковый	⋯ Дискретный							13	1,0000	Пригоден
2	w_avg	9.0 Вещественный	— Непрерывный								0,7898	Пригоден
3	t_avg	9.0 Вещественный	— Непрерывный								0,8940	Пригоден
4	l_avg	9.0 Вещественный	— Непрерывный								0,9440	Пригоден
5	p_p90_avg	9.0 Вещественный	— Непрерывный								0,7898	Пригоден
6	gf_avg	9.0 Вещественный	— Непрерывный								0,9061	Пригоден
7	ga_avg	9.0 Вещественный	— Непрерывный								0,8278	Пригоден
8	gd_avg	9.0 Вещественный	— Непрерывный								0,6776	Пригоден
9	xg_for_avg	9.0 Вещественный	— Непрерывный								0,8278	Пригоден
10	xg_ag_avg	9.0 Вещественный	— Непрерывный								0,8561	Пригоден
11	xgd_avg	9.0 Вещественный	— Непрерывный								0,8561	Пригоден
12	pl_used_avg	9.0 Вещественный	— Непрерывный								0,9636	Пригоден
13	avg_age_avg	9.0 Вещественный	— Непрерывный								0,9190	Пригоден
14	poss_avg	9.0 Вещественный	— Непрерывный								0,9311	Пригоден
15	nPg_for_avg	9.0 Вещественный	— Непрерывный								0,8778	Пригоден
16	nPxg_for_avg	9.0 Вещественный	— Непрерывный								0,8278	Пригоден
17	pA_for_avg	9.0 Вещественный	— Непрерывный								0,8398	Пригоден
18	pS_for_avg	9.0 Вещественный	— Непрерывный								0,8115	Пригоден
19	pr_carr_for_avg	9.0 Вещественный	— Непрерывный								0,8940	Пригоден
20	pr_pass_for_avg	9.0 Вещественный	— Непрерывный								0,9440	Пригоден
21	y_cards_for_avg	9.0 Вещественный	— Непрерывный								0,9190	Пригоден
22	r_cards_for_avg	9.0 Вещественный	— Непрерывный								0,9190	Пригоден
23	nPg_ag_avg	9.0 Вещественный	— Непрерывный								0,8940	Пригоден
24	nPxg_ag_avg	9.0 Вещественный	— Непрерывный								0,7648	Пригоден
25	pA_ag_avg	9.0 Вещественный	— Непрерывный								0,9886	Пригоден
26	pS_ag_avg	9.0 Вещественный	— Непрерывный								0,9190	Пригоден
27	pr_carr_ag_avg	9.0 Вещественный	— Непрерывный								0,9061	Пригоден
28	pr_pass_ag_avg	9.0 Вещественный	— Непрерывный								0,8561	Пригоден
29	y_cards_ag_avg	9.0 Вещественный	— Непрерывный								0,8778	Пригоден
30	r_cards_ag_avg	9.0 Вещественный	— Непрерывный								0,9190	Пригоден

Рисунок 9 Аналіз якості даних

За допомогою програми Deductor Studio Academic знаходимо коефіцієнт кореляції Пірсона для всіх стовпців по відношенню до вихідних даних, якими буде кількість очок:

Входные поля		Корреляция с выходными полями
№	Поле	p_p90_avg
1	w_avg	0,994
2	t_avg	-0,611
3	l_avg	-0,961
4	gf_avg	0,967
5	ga_avg	-0,947
6	gd_avg	0,983
7	xg_for_avg	0,945
8	xg_ag_avg	-0,832
9	xgd_avg	0,932
10	pl_used_avg	-0,079
11	avg_age_avg	-0,253
12	poss_avg	0,852
13	nPg_for_avg	0,963
14	nPxg_for_avg	0,945
15	pA_for_avg	0,791
16	pS_for_avg	0,946
17	pr_carr_for_avg	0,860
18	pr_pass_for_avg	0,895
19	y_cards_for_avg	-0,606
20	r_cards_for_avg	-0,195
21	nPg_ag_avg	-0,931
22	nPxg_ag_avg	-0,821
23	pA_ag_avg	-0,402
24	pS_ag_avg	-0,355
25	pr_carr_ag_avg	-0,758
26	pr_pass_ag_avg	-0,863
27	y_cards_ag_avg	-0,188
28	r_cards_ag_avg	-0,215

Рисунок 10 Коефіцієнт корелязії статистичних показників до кількості очок

Згідно з коефіцієнтами кореляції середній вік гравців, кількість використаних гравців за сезон, кількість жовтих та червоних карток у команди та суперника, а також кількість пробитих і забитих суперником пенальті мають досить малий вплив на кількість очок.

Через це при побудові дерева рішень ми залишимо ці стовпці разом зі стовпцем назв команд інформаційними. Стовпець кількості очок за матч буде вихідними даними, а всі інші стовпці стануть вхідними даними.

Побудуємо перше дерево рішень, яке буде базуватися на відсотку перемог команд:

Условие	Следствие	Поддержка	Достоверность
ЕСЛИ		13	1
w_avg < 0,53		10	1
w_avg < 0,4245		7	1
w_avg < 0,3605		4	1
w_avg < 0,324	1,18	2	1
w_avg >= 0,324	1,277	2	1
w_avg >= 0,3605	1,333	3	1
w_avg >= 0,4245	1,619	3	1
w_avg >= 0,53	1,819	3	1

Рисунок 11 Дерево рішень №1

Протестуємо його на даних про поточний сезон. Відфільтруємо дані за умовою, що процент перемог має бути вищий або дорівнювати 42,45%. Бачимо, що всі команди окрім West Ham мають середню кількість очок за матч дійсно більшу за 1,619. Подібна ситуація з West Ham сталася через замалий відсоток нічиїх команди, тобто команда виграє велику кількість матчів, але ті матчі, що програє по ходу зустрічі, не може перевести до нічийного завершення.

team	p p90	w	t	l	xg for	xg ag	gf	ga	pr pass for	pr carr for	pr pass ag	pr carr ag
Arsenal	2,3	69,231	23,077	7,692	1,738	0,792	2,077	0,769	52,308	20,462	22,769	12,308
Man City	2,23	69,231	15,385	15,385	1,985	0,892	2,538	1	51,231	28,077	22,308	14,077
Villa	2,15	69,231	7,692	23,077	1,946	1,4	2,385	1,385	40,462	20,692	32,769	17
Liverpool	2,15	61,538	30,769	7,692	2,085	1,215	2,154	0,846	51,154	22,538	36,846	17,077
Tottenham	2	61,538	15,385	23,077	1,654	1,562	1,923	1,308	53,615	24,615	29,231	15,231
Man Utd	1,85	61,538	0	38,462	1,515	1,631	1,231	1,231	46,385	20,077	32,231	21,231
Newcastle	1,77	53,846	15,385	30,769	1,977	1,054	2,385	1,077	39	18,462	32,538	15,769
Brighton	1,69	46,154	30,769	23,077	1,715	1,454	2,154	1,769	45,308	22,846	31,231	16,385
West Ham	1,54	46,154	15,385	38,462	1,423	1,808	1,769	1,769	32	15,077	50,615	25,462

Рисунок 12 Фільтрація даних згідно з Деревом рішень №1

Побудуємо друге дерево рішень, яке буде базуватися на кількості пасів з просуванням в середньому за матч з боку команди:

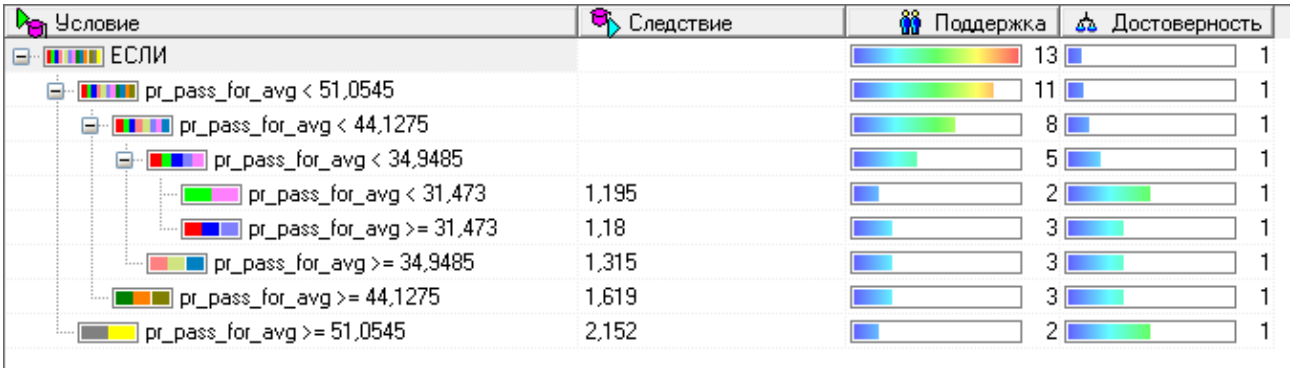


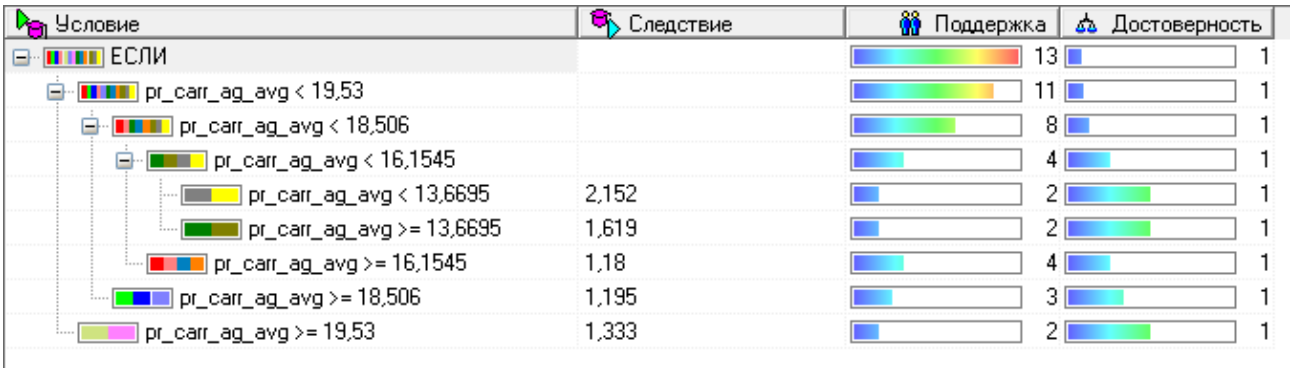
Рисунок 13 Дерево рішень №2

Протестуємо його на даних про поточний сезон. Відфільтруємо дані за умовою, що кількість пасів з просуванням має бути нижчою ніж 34,95. Бачимо, що всі команди окрім West Ham мають середню кількість очок за матч дійсно більшу за 1,315 і по факту підпадають під іншу умову де при кількості пасів меншій за 31,47 кількість очок складає менше ніж 1,195. Це знову ж таки відображує нетиповий для даної команди оверперформанс, який вона показує на початку сезону, і може казати про майбутній спад результатів команди. Bournemouth має третій найвищий показник в даній таблиці, але при цьому його кількість очок за матч є однією з найнижчих, що може вказувати на майбутнє покращення результатів команди

team	p p90	pr carr for	pr pass for	pr carr ag	pr pass ag	xg for	xg ag	games	w	t	l
West Ham	1,54	15,077	32	25,462	50,615	1,423	1,808	13	46,154	15,385	38,462
Palace	1,15	14	31,923	17,769	45,692	1,215	1,362	13	30,769	23,077	46,154
Bournemouth	0,92	16,846	30,231	20,462	41,154	1,285	1,923	13	23,077	23,077	53,846
Wolves	1,15	18,692	30	19,692	44,846	1,362	1,723	13	30,769	23,077	46,154
Everton	1,08	14,462	29,077	19,308	42,615	1,562	1,385	13	30,769	15,385	53,846
Burnley	0,31	16,615	28,462	15,462	35,538	0,862	1,762	13	7,692	7,692	84,615
Forest	1	14,615	28,231	21,538	52,385	1,162	1,554	13	23,077	30,769	46,154
Luton	0,69	15,385	27,923	18,846	48,538	1,062	2,023	13	15,385	23,077	61,538
Shelfield	0,39	8,231	24,462	25,692	47,923	0,8	2,231	13	7,692	15,385	76,923

Рисунок 14 Фільтрація даних згідно з Деревом рішень №2

Побудуємо третє дерево рішень, яке буде базуватися на кількості допущених проходів з просуванням в сторону своїх воріт в середньому за матч з боку команди:



Условие	Следствие	Поддержка	Достоверность
ЕСЛИ		13	1
pr_carr_ag_avg < 19,53		11	1
pr_carr_ag_avg < 18,506		8	1
pr_carr_ag_avg < 16,1545		4	1
pr_carr_ag_avg < 13,6695	2,152	2	1
pr_carr_ag_avg >= 13,6695	1,619	2	1
pr_carr_ag_avg >= 16,1545	1,18	4	1
pr_carr_ag_avg >= 18,506	1,195	3	1
pr_carr_ag_avg >= 19,53	1,333	2	1

Рисунок 15 Дерево рішень №3

Протестуємо його на даних про поточний сезон. Відфільтруємо дані за умовою, що кількість допущених проходів з просуванням в сторону своїх воріт має бути нижчою ніж 16,16. Після фільтрації ми дійсно отримали 4 команди, які є одними з лідерів чемпіонату, але також с списку присутній Burnley. Дана аномалія є лише підтвердженням того, що пункт статистики, який ми розглядаємо, не є єдиним найважливішим при формуванні умовної сили команди. Команда в середньому пропускає за матч на 0,7 голів більше ніж мала б, до того ж вона має дуже слабку атаку, яка створює дуже мало пасів з просуванням та проходів з просуванням в порівнянні з іншими командами

team	p p90	pr carr ag	pr pass ag	pr carr for	pr pass for	xg for	xg ag	gf	ga	p
Arsenal	2,3	12,308	22,769	20,462	52,308	1,738	0,792	2,077	0,769	30
Man City	2,23	14,077	22,308	28,077	51,231	1,985	0,892	2,538	1	29
Tottenham	2	15,231	29,231	24,615	53,615	1,654	1,562	1,923	1,308	26
Burnley	0,31	15,462	35,538	16,615	28,462	0,862	1,762	0,769	2,462	4
Newcastle	1,77	15,769	32,538	18,462	39	1,977	1,054	2,385	1,077	23

Рисунок 16 Фільтрація даних згідно з Деревом рішень №3

2.7 Порівняння зроблених висновків з фактичними подальшими виступами команд

Команди, які ми розглядали в контексті поточного сезону? провели ще по три гри кожна, тому давайте розглянемо результати цих ігор і порівняємо їх з тими прогнозами, які ми робили в минулих розділах:

Everton, з боку якого спостерігався великий ступінь андерперформансу, в останніх трьох матчах здобув 3 перемоги і 9 очок, що на 5 очок менше, ніж вони заробили в перших 13 матчах чемпіонату. Це показує, що ближче до середини сезону команда почала грати на тому рівні, який є її прогнозованим, і зараз вона має значення середньої кількості очок за матч 1,44 (було 1,08)

10.12. 16:00	**Евертон**	2	(0)	В
	Челсі	0	(0)	
07.12. 21:30	**Евертон**	3	(0)	В
	Ньюкасл Юнайтед	0	(0)	
02.12. 19:30	Ноттінгем	0	(0)	В
	Евертон	1	(0)	

Рисунок 17 Результати команди Евертон

Bournemouth, який також показував великий андерперформанс, в останніх своїх матчах заробив 7 очок, чим покращив своє турнірне становище і став більше відповідати своїм статистичним показникам з середньої кількістю очок за матч 1,19 (було 0,92)

09.12. 17:00	Манчестер Юнайтед	0	(0)	В
	Борнмут	3	(1)	
06.12. 21:30	Кристал Пелес	0	(0)	В
	Борнмут	2	(1)	
03.12. 16:00	Борнмут	2	(1)	Н
	Астон Вілла	2	(1)	

Рисунок 18 Результати команди Борнмут

West Ham в свою чергу почав виступати гірше і за останні 3 матчі заробив тільки 4 очки, при чому очки були втрачені саме в матчах з командами

«середняками», що вказує на те, що це і є та група команд, до якої West Ham має належати. Середня кількість очок команди за матч хоч і несильно, але знизилася до 1,5 (було 1,54)

10.12. 16:00	**Фулгем**	5	(3)	П
	Вест Гем	0	(0)	
07.12. 22:15	Тоттенгем	1	(1)	В
	Вест Гем	2	(0)	
03.12. 16:00	Вест Гем	1	(1)	Н
	Кристал Пелес	1	(0)	

Рисунок 19 Результати команди Вест Гем

Burnley, який ми відмічали як аномальну команду на етапі формування дерева рішень також почав набирати очки – 4 за останні три матчі, що підвищило значення середньої кількості очок за матч до 0,5 (було 0,31)

09.12. 17:00	Брайтон	1	(0)	Н
	Бернлі	1	(1)	
05.12. 21:30	**Вулвергемптон**	1	(1)	П
	Бернлі	0	(0)	
02.12. 17:00	**Бернлі**	5	(2)	В
	Шеффілд Юнайтед	0	(0)	

Рисунок 20 Результати команди Бернлі

2.8 Висновки до розділу 2

У розділі 2 ми за допомогою парсингу отримали статистичні дані команд Англійської Прем’єр Ліги та провели їх аналіз за допомогою методу візуалізації, методу головних компонент, коефіцієнту кореляції Пірсона та побудови дерева рішень. Для цього ми використовували мову програмування Python та програмне забезпечення Deductor Studio Academic.

При побудові точкових діаграм, які підходили для нашої роботи через підтримку великих масивів даних, ми відстежили залежність результатів виступів команд від очікуваних та фактичних голів. Також на діаграмах було відображено залежність результатів виступів команд в залежності від того, наскільки їх забиті та пропущені голи відповідають фактичним.

Побудова лінійного графіку дозволила нам простежити екстремальність росту результатів найбільш прогресуючих команд, що дозволило нам проводити більш адекватний аналіз їх результатів

Після проведення аналізу даних методом головних компонент ми прийшли до висновку, що всі статистичні показники з таблиці можна звести до однієї головної компоненти, яка буде відображуватися умовну силу кожної з команд по відношенню до інших

Коефіцієнт кореляції Пірсона дав розуміння про те, які саме з показників мають найменший вплив на результати команд, через що в подальшому будуванні дерева рішень дані показники не використовувалися

Побудова дерев рішень за допомогою програмного забезпечення Deductor Studio Academic дала нам розуміння того, за якими правилами ми можемо оцінювати поточні виступи команд в лізі, що дало нам змогу спрогнозувати покращення та погіршення результатів конкретних команд на такій короткій дистанції як 3 матчі.

ВИСНОВКИ

Якісний аналіз даних є невід’ємною частиною успіху в будь-якій сфері життя, бізнесу чи спорту. Правильний глибокий аналіз дозволяє більш чітко розуміти ситуацію, в якій знаходиться спортивний клуб чи ліга загалом, що в свою чергу дає розуміння того, які подальші дії необхідні для покращення, стабілізації чи модернізації тої чи іншої ланки спорту. Якщо розглядати саме керівництво спортивних клубів, то це дозволяє формувати очікування від результатів клуба, які відповідають дійсності, що може позитивно позначитися на стабільності клубів, наприклад, не звільняти тренерів під час турбулентних періодів або ж не проводити різких змін складу. Яскравим прикладом відсутності подібного аналізу є новий власник команди Chelsea, за часів якого клуб майже повністю змінив свій склад та вже змінив 4 тренерів і все це за період в 18 місяців. Подібні необачні дії призвели до ще більшого погіршення результатів команди і повної стагнації бренду в цілому. Саме для уникнення подібних ситуацій та проведення зважених прогнозів і необхідний аналіз даних

У ході аналізу результатів клубів Англійської Прем’єр Ліги за останні 5 сезонів було використано декілька методів аналізу: методі візуалізації, метод головних компонент, дерева рішень. За допомогою візуалізації даних ми кластеризували команди і виявили причетні ним риси, що дало змогу спрогнозувати майбутні зміни в їх результатах. За допомогою методу головних компонент визначили умовний рейтинг сили команд, що брали участь у всіх 5 сезонах. За допомогою побудованих дерев рішень ми отримали інструменти для визначення того, наскільки відповідають поточні результати команд їх рівню гри, що дало ще більше можливостей прогнозувати майбутні зміни в їх результатах

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “Premier League Stats” URL: https://shorturl.at/iESX6
2. “Flashscore” URL: https://www.flashscore.ua/
3. В. Є. Бахрушин, Методи аналізу даних : навчальний посібник для студентів.: видав. КПУ, 2011. - 268 с.
4. Edward Tufte, "The Visual Display of Quantitative Information". Graphics Press. 2001. – 200 с.
5. Julie Steele and Noah Iliinsky, "Beautiful Visualization: Looking at Data through the Eyes of Experts". O'Reilly Media. 2010 – 415 с.
6. Чубуркова І.А., “Data Mining”.: видав. КНЕУ. 2020 – 326 с.
7. Max Kuhn and Kjell Johnson, "Applied Predictive Modeling".: Springer. 2013. – 613 с.
8. “Machine Learning Yearning By Andrew NG” URL: https://info.deeplearning.ai/machine-learning-yearning-book

Додаток А

Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення	Найменування	Кількість аркушів	Примітки
1				
2		Документація		
3				
4	САУ.КР.23.01.ПЗ	Пояснювальна записка	55	Формат А4
5				
6		Демонстраційний матеріал		Презентація на CD-R
7				
8		Копія роботи	1	Диск CD-R
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

					САУ.КР.23.01.ДА.ПЗ.				
Змін.	Аркуш	№ докум.	Підпис	Дата					
Розроб.		Швидкий Р.О.			**Матеріали кваліфікаційної роботи**	Літ.		Аркуш	Аркушів
К. розд.		Коряшкіна Л.С.							
Керівн.		Коряшкіна Л.С.				НТУ «ДП», 12; 124м-22-1			
Н.контр.		Хом'як Т. В.							
Зав. каф.		Желдак Т. А.							

Додаток В

	team	games	w	t	l	p	p p90	gf	ga	gd	xg for	xg ag	xgd	pl used	avg age	poss	nPg for	pS for	pA for	nPxg for	pr carr for	pr pass for	y cards for	r cards for	nPg ag	pS ag	pA ag	nPxg ag	pr carr ag	pr pass ag	y cards ag	r cards ag
0	Arsenal	13	69.231	23.077	7.692	30	2.3	2.077	0.769	1.308	1.738	0.792	0.94	23	25.2	60.9	1.538	1.392	0.462	1.711	20.462	52.308	1.462	0.154	0.692	0.077	0.077	0.788	12.308	22.769	2.615	0.077
1	Villa	13	69.231	7.692	23.077	28	2.15	2.385	1.385	1.0	1.946	1.4	0.54	23	27.2	52.5	1.923	1.769	0.231	1.933	20.692	40.462	2.462	0.0	1.231	0.0	0.0	1.4	17.0	32.769	2.462	0.154
2	Bournemouth	13	23.077	23.077	53.846	12	0.92	1.077	2.154	-1.077	1.285	1.923	-0.64	25	25.8	43.8	1.077	1.285	0.0	1.285	16.846	30.231	1.615	0.077	1.923	0.154	0.231	1.91	20.462	41.154	2.538	0.077
3	Brentford	13	30.769	30.769	38.462	16	1.23	1.462	1.385	0.077	1.792	1.377	0.41	25	26.7	46.8	1.231	1.685	0.154	1.783	13.692	36.846	2.385	0.0	1.308	0.077	0.077	1.373	19.385	39.154	2.231	0.231
4	Brighton	13	46.154	30.769	23.077	22	1.69	2.154	1.769	0.385	1.715	1.454	0.26	26	27.1	60.8	1.846	1.592	0.154	1.706	22.846	45.308	2.692	0.154	1.385	0.231	0.231	1.441	16.385	31.231	3.077	0.154
5	Burnley	13	7.692	7.692	84.615	4	0.31	0.769	2.462	-1.692	0.862	1.762	-0.9	25	25.2	49.9	0.692	0.8	0.077	0.858	16.615	28.462	2.0	0.231	2.154	0.154	0.154	1.753	15.462	35.538	2.385	0.077
6	Chelsea	13	30.769	30.769	38.462	16	1.23	1.692	1.538	0.154	1.985	1.338	0.64	24	25.2	59.2	1.308	1.685	0.308	1.967	24.231	42.385	3.538	0.154	1.385	0.154	0.154	1.329	16.385	32.769	2.923	0.231
7	Palace	13	30.769	23.077	46.154	15	1.15	1.0	1.385	-0.385	1.215	1.362	-0.15	22	27.8	45.0	0.923	1.154	0.077	1.211	14.0	31.923	1.769	0.0	1.154	0.154	0.154	1.353	17.769	45.692	2.538	0.077
8	Everton	13	30.769	15.385	53.846	14	1.08	1.077	1.538	-0.462	1.562	1.385	0.18	23	27.6	40.8	1.077	1.562	0.0	1.562	14.462	29.077	2.154	0.077	1.077	0.308	0.308	1.367	19.308	42.615	1.846	0.0
9	Fulham	13	30.769	23.077	46.154	15	1.15	1.0	1.692	-0.692	1.177	1.792	-0.62	23	29.5	49.5	0.769	1.054	0.154	1.168	16.154	36.154	2.846	0.154	1.231	0.308	0.308	1.774	16.692	36.0	1.769	0.0
10	Liverpool	13	61.538	30.769	7.692	28	2.15	2.154	0.846	1.308	2.085	1.215	0.86	22	27.3	57.2	1.769	1.831	0.231	2.072	22.538	51.154	1.923	0.308	0.769	0.0	0.0	1.215	17.077	36.846	2.077	0.077
11	Luton	13	15.385	23.077	61.538	9	0.69	0.923	1.769	-0.846	1.062	2.023	-0.96	23	27.2	36.4	0.692	0.946	0.154	1.053	15.385	27.923	2.0	0.0	1.615	0.077	0.077	2.019	18.846	48.538	1.692	0.154
12	Man City	13	69.231	15.385	15.385	29	2.23	2.538	1.0	1.538	1.985	0.892	1.09	25	26.9	62.2	2.308	1.746	0.231	1.972	28.077	51.231	1.846	0.154	0.846	0.077	0.077	0.888	14.077	22.308	3.462	0.077
13	Man Utd	13	61.538	0.0	38.462	24	1.85	1.231	1.231	0.0	1.515	1.631	-0.12	26	27.2	53.9	1.077	1.4	0.154	1.506	20.077	46.385	2.0	0.0	1.0	0.154	0.154	1.622	21.231	32.231	2.615	0.077
14	Newcastle	13	53.846	15.385	30.769	23	1.77	2.385	1.077	1.308	1.977	1.054	0.92	29	28.0	53.2	2.154	1.8	0.231	1.964	18.462	39.0	2.846	0.0	1.077	0.0	0.0	1.054	15.769	32.538	3.0	0.154
15	Forest	13	23.077	30.769	46.154	13	1.0	1.231	1.615	-0.385	1.162	1.554	-0.39	27	26.5	39.5	1.154	1.1	0.077	1.158	14.615	28.231	2.769	0.154	1.462	0.154	0.154	1.545	21.538	52.385	2.308	0.231
16	Sheffield	13	7.692	15.385	76.923	5	0.39	0.846	2.615	-1.769	0.8	2.231	-1.43	27	26.9	38.8	0.462	0.677	0.154	0.791	8.231	24.462	3.0	0.077	2.462	0.077	0.154	2.222	25.692	47.923	1.538	0.077
17	Tottenham	13	61.538	15.385	23.077	26	2.0	1.923	1.308	0.615	1.654	1.562	0.09	24	25.5	59.7	1.692	1.654	0.0	1.654	24.615	53.615	2.923	0.231	1.0	0.231	0.231	1.549	15.231	29.231	3.231	0.231
18	West Ham	13	46.154	15.385	38.462	20	1.54	1.769	1.769	0.0	1.423	1.808	-0.38	21	28.8	42.7	1.615	1.362	0.077	1.419	15.077	32.0	2.692	0.077	1.462	0.231	0.308	1.79	25.462	50.615	1.308	0.0
19	Wolves	13	30.769	23.077	46.154	15	1.15	1.385	1.769	-0.385	1.362	1.723	-0.36	22	27.1	47.2	1.231	1.308	0.077	1.358	18.692	30.0	3.308	0.231	1.308	0.385	0.385	1.701	19.692	44.846	2.615	0.077

Рисунок 21 Таблиця даних команд за сезон 2023/24

	team	games	w	t	l	p	p p90	gf	ga	gd	xg for	xg ag	xgd	pl used	avg age	poss	nPg for	pS for	pA for	nPxg for	pr carr for	pr pass for	y cards for	r cards for	nPg ag	pS ag	pA ag	nPxg ag	pr carr ag	pr pass ag	y cards ag	r cards ag
0	Arsenal	38	36.842	36.842	26.316	56	1.47	1.474	1.263	0.211	1.237	1.455	-0.22	29	25.8	53.8	1.395	1.174	0.079	1.232	21.447	43.184	2.316	0.132	1.079	0.184	0.211	1.029	19.974	36.026	2.079	0.026
1	Villa	38	23.684	21.053	55.263	35	0.92	1.079	1.763	-0.684	1.166	1.784	-0.62	28	25.7	44.1	1.026	1.1	0.026	1.164	16.816	32.816	1.842	0.026	1.605	0.132	0.158	1.226	22.053	47.974	2.368	0.184
2	Bournemouth	38	23.684	18.421	57.895	34	0.89	1.053	1.711	-0.658	1.179	1.639	-0.46	27	25.2	44.1	0.895	1.097	0.105	1.173	15.816	33.053	2.053	0.079	1.579	0.132	0.158	1.121	19.553	43.947	1.763	0.079
3	Brighton	38	23.684	36.842	39.474	41	1.08	1.026	1.421	-0.395	1.195	1.447	-0.25	25	26.4	52.2	0.895	1.153	0.026	1.194	17.605	44.079	1.553	0.053	1.316	0.053	0.053	1.134	20.342	35.316	1.605	0.026
4	Burnley	38	39.474	23.684	36.842	54	1.42	1.132	1.316	-0.184	1.247	1.311	-0.06	22	28.0	41.9	1.0	1.192	0.079	1.242	11.5	23.421	1.763	0.0	1.184	0.105	0.132	0.963	22.553	40.632	1.105	0.079
5	Chelsea	38	52.632	15.789	31.579	66	1.74	1.816	1.421	0.395	1.821	1.016	0.81	27	25.5	60.4	1.632	1.695	0.184	1.81	25.868	49.921	1.579	0.0	1.289	0.053	0.053	0.721	12.211	30.289	2.0	0.079
6	Palace	38	28.947	26.316	44.737	43	1.13	0.816	1.316	-0.5	0.942	1.5	-0.56	25	29.4	44.8	0.684	0.884	0.079	0.937	15.684	32.184	1.632	0.053	1.263	0.0	0.026	1.053	20.237	44.474	2.316	0.132
7	Everton	38	34.211	26.316	39.474	49	1.29	1.158	1.474	-0.316	1.337	1.271	0.07	24	25.5	49.0	1.079	1.316	0.026	1.336	16.395	37.632	2.0	0.079	1.316	0.079	0.105	0.9	17.763	36.974	1.605	0.053
8	Leicester City	38	47.368	21.053	31.579	62	1.63	1.763	1.079	0.684	1.637	1.197	0.44	24	26.1	57.1	1.579	1.492	0.132	1.629	19.447	42.421	1.132	0.079	0.816	0.211	0.263	0.768	17.184	31.947	1.816	0.079
9	Liverpool	38	84.211	7.895	7.895	99	2.61	2.237	0.868	1.368	1.813	0.992	0.82	24	26.6	62.9	2.053	1.711	0.132	1.805	25.447	51.079	1.0	0.026	0.816	0.026	0.026	0.779	13.526	24.974	1.237	0.0
10	Man City	38	68.421	7.895	23.684	81	2.13	2.684	0.921	1.763	2.421	0.953	1.47	24	26.9	66.2	2.474	2.2	0.158	2.412	31.553	64.737	1.684	0.105	0.816	0.079	0.079	0.608	11.789	21.079	1.395	0.026
11	Man Utd	38	47.368	31.579	21.053	66	1.74	1.737	0.947	0.789	1.637	0.992	0.64	29	24.8	55.8	1.447	1.347	0.263	1.622	24.368	47.053	1.921	0.0	0.842	0.079	0.079	0.663	17.053	31.684	2.079	0.053
12	Newcastle	38	28.947	28.947	42.105	44	1.16	1.0	1.526	-0.526	0.934	1.726	-0.79	28	26.5	39.0	1.0	0.916	0.0	0.934	14.132	22.211	1.737	0.079	1.421	0.053	0.053	1.326	21.263	49.526	1.5	0.053
13	Norwich City	38	13.158	15.789	71.053	21	0.55	0.684	1.974	-1.289	0.974	1.75	-0.78	30	26.0	49.3	0.605	0.932	0.053	0.971	17.711	30.763	1.789	0.079	1.789	0.105	0.184	1.226	22.263	47.184	1.526	0.053
14	Sheffield	38	36.842	31.579	31.579	54	1.42	1.026	1.026	0.0	1.132	1.342	-0.21	26	26.9	43.1	0.921	1.111	0.026	1.13	12.974	35.579	1.658	0.053	1.0	0.026	0.079	1.003	22.105	37.368	1.211	0.0
15	Southampton	38	39.474	18.421	42.105	52	1.37	1.342	1.579	-0.237	1.476	1.389	0.09	25	25.9	48.9	1.289	1.371	0.053	1.473	16.526	35.132	1.447	0.105	1.474	0.079	0.105	1.018	16.947	38.237	1.763	0.079
16	Tottenham	38	42.105	28.947	28.947	59	1.55	1.605	1.237	0.368	1.261	1.426	-0.17	29	26.7	52.0	1.395	1.176	0.079	1.256	20.211	37.105	2.211	0.079	1.053	0.132	0.184	0.895	18.816	40.895	2.184	0.026
17	Watford	38	21.053	26.316	52.632	34	0.89	0.947	1.684	-0.737	1.287	1.529	-0.24	26	28.9	42.9	0.737	1.121	0.184	1.276	15.211	30.737	2.105	0.079	1.395	0.211	0.237	1.037	22.211	43.526	1.737	0.026
18	West Ham	38	26.316	23.684	50.0	39	1.03	1.289	1.632	-0.342	1.279	1.679	-0.4	29	27.4	44.1	1.184	1.195	0.105	1.273	17.526	36.237	1.684	0.026	1.421	0.132	0.158	1.282	22.5	45.368	1.184	0.0
19	Wolves	38	39.474	36.842	23.684	59	1.55	1.342	1.053	0.289	1.424	1.0	0.42	21	26.6	48.3	1.184	1.342	0.105	1.418	22.579	35.211	1.605	0.053	1.0	0.026	0.079	0.687	18.474	37.132	2.237	0.132

Рисунок 22 Таблиця даних команд за сезон 2019/20

	team	games	w	t	l	p	p p90	gf	ga	gd	xg for	xg ag	xgd	pl used	avg age	poss	nPg for	pS for	pA for	nPxg for	pr carr for	pr pass for	y cards for	r cards for	nPg ag	pS ag	pA ag	nPxg ag	pr carr ag	pr pass ag	y cards ag	r cards ag
0	Arsenal	38	47.368	18.421	34.211	61	1.61	1.447	1.026	0.421	1.363	1.132	0.23	29	25.9	53.5	1.237	1.237	0.158	1.354	19.947	46.316	1.289	0.132	0.868	0.053	0.079	1.068	15.895	30.921	1.947	0.053
1	Villa	38	42.105	18.421	39.474	55	1.45	1.447	1.211	0.237	1.382	1.345	0.04	24	25.2	48.5	1.237	1.261	0.132	1.374	16.553	37.368	1.868	0.105	1.053	0.132	0.158	1.221	19.105	40.763	1.921	0.184
2	Brighton	38	23.684	36.842	39.474	41	1.08	1.053	1.211	-0.158	1.339	0.929	0.41	27	25.8	51.1	0.868	1.153	0.158	1.33	18.026	39.895	1.289	0.158	1.0	0.158	0.184	0.784	15.447	34.263	1.342	0.053
3	Burnley	38	26.316	23.684	50.0	39	1.03	0.868	1.447	-0.579	1.034	1.439	-0.4	25	28.3	42.1	0.763	0.971	0.079	1.029	10.237	26.632	1.263	0.0	1.342	0.079	0.079	1.376	23.263	43.816	0.842	0.053
4	Chelsea	38	50.0	26.316	23.684	67	1.76	1.526	0.947	0.579	1.642	0.797	0.85	27	26.0	60.9	1.263	1.434	0.211	1.63	23.553	47.316	1.342	0.079	0.842	0.079	0.105	0.713	13.079	27.947	1.474	0.079
5	Palace	38	31.579	21.053	47.368	44	1.16	1.079	1.737	-0.658	0.897	1.532	-0.64	24	29.1	40.6	0.947	0.816	0.079	0.892	13.842	29.789	1.474	0.053	1.658	0.079	0.105	1.45	17.526	47.553	1.158	0.105
6	Everton	38	44.737	21.053	34.211	59	1.55	1.237	1.263	-0.026	1.203	1.318	-0.12	29	26.3	46.4	1.079	1.103	0.105	1.197	15.895	31.158	1.553	0.053	1.158	0.079	0.105	1.234	20.895	43.789	1.737	0.026
7	Fulham	38	13.158	34.211	52.632	28	0.74	0.711	1.395	-0.684	1.066	1.384	-0.32	28	25.2	49.7	0.605	0.945	0.079	1.061	19.132	35.632	1.763	0.079	1.184	0.184	0.184	1.242	14.711	34.921	1.053	0.026
8	Leeds United	38	47.368	13.158	39.474	59	1.55	1.632	1.421	0.211	1.463	1.524	-0.06	23	26.1	57.3	1.474	1.382	0.105	1.457	21.342	42.974	1.605	0.026	1.158	0.211	0.237	1.334	18.947	31.605	1.289	0.053
9	Leicester City	38	52.632	15.789	31.579	66	1.74	1.789	1.316	0.474	1.468	1.211	0.26	27	26.5	54.3	1.421	1.218	0.263	1.453	17.526	38.526	1.605	0.0	1.158	0.105	0.105	1.129	14.921	33.0	1.605	0.026
10	Liverpool	38	52.632	23.684	23.684	69	1.82	1.789	1.105	0.684	1.776	1.132	0.65	28	26.8	62.1	1.553	1.655	0.158	1.767	23.316	54.132	1.053	0.0	1.0	0.105	0.211	0.971	14.184	25.579	1.474	0.053
11	Man City	38	71.053	13.158	15.789	86	2.26	2.184	0.842	1.342	1.795	0.795	1.0	24	26.1	63.5	2.026	1.611	0.132	1.787	28.842	53.974	1.211	0.053	0.605	0.211	0.263	0.587	14.132	22.368	1.605	0.053
12	Man Utd	38	55.263	28.947	15.789	74	1.95	1.921	1.158	0.763	1.582	1.089	0.49	29	25.6	55.7	1.579	1.353	0.263	1.567	23.079	48.211	1.684	0.026	1.0	0.105	0.105	1.011	15.079	32.342	1.684	0.079
13	Newcastle	38	31.579	23.684	44.737	45	1.18	1.211	1.632	-0.421	1.142	1.534	-0.39	27	27.1	38.8	1.0	0.995	0.158	1.133	12.289	23.763	1.711	0.079	1.447	0.105	0.132	1.432	22.395	46.605	1.579	0.079
14	Sheffield	38	18.421	5.263	76.316	23	0.61	0.526	1.658	-1.132	0.858	1.637	-0.78	27	26.7	41.7	0.421	0.776	0.079	0.853	12.921	31.474	1.921	0.079	1.5	0.079	0.105	1.555	21.789	42.263	1.026	0.026
15	Southampton	38	31.579	18.421	50.0	43	1.13	1.237	1.789	-0.553	1.084	1.397	-0.31	29	26.6	52.0	1.105	0.963	0.132	1.076	16.474	31.447	1.368	0.079	1.553	0.211	0.237	1.213	14.605	37.026	1.526	0.079
16	Tottenham	38	47.368	21.053	31.579	62	1.63	1.789	1.184	0.605	1.397	1.292	0.1	24	27.2	51.6	1.605	1.295	0.132	1.389	16.658	35.0	1.5	0.053	0.842	0.263	0.263	1.084	19.026	39.842	2.026	0.026
17	West Brom	38	13.158	28.947	57.895	26	0.68	0.921	2.0	-1.079	0.942	1.787	-0.84	30	26.4	37.9	0.763	0.861	0.105	0.936	11.605	27.026	1.342	0.105	1.737	0.184	0.289	1.561	24.5	47.421	1.316	0.026
18	West Ham	38	50.0	21.053	28.947	65	1.71	1.632	1.237	0.395	1.458	1.282	0.18	24	27.8	43.2	1.526	1.371	0.053	1.455	17.316	33.737	1.316	0.079	1.053	0.079	0.132	1.179	21.737	44.684	1.368	0.026
19	Wolves	38	31.579	23.684	44.737	45	1.18	0.947	1.368	-0.421	0.961	1.303	-0.34	27	26.3	49.3	0.789	0.879	0.105	0.955	21.053	33.763	1.447	0.026	1.105	0.184	0.211	1.137	18.368	41.421	1.632	0.158

Рисунок 23 Таблиця даних команд за сезон 2020/21

	team	games	w	t	l	p	p p90	gf	ga	gd	xg for	xg ag	xgd	pl used	avg age	poss	nPg for	pS for	pA for	nPxg for	pr carr for	pr pass for	y cards for	r cards for	nPg ag	pS ag	pA ag	nPxg ag	pr carr ag	pr pass ag	y cards ag	r cards ag
0	Arsenal	38	57.895	7.895	34.211	69	1.82	1.605	1.263	0.342	1.592	1.203	0.39	27	24.4	52.8	1.447	1.45	0.132	1.584	19.316	43.553	1.763	0.105	1.105	0.132	0.158	1.076	14.632	31.342	1.816	0.079
1	Villa	38	34.211	15.789	50.0	45	1.18	1.368	1.421	-0.053	1.158	1.289	-0.13	31	26.1	46.5	1.237	1.105	0.079	1.153	17.184	34.211	2.132	0.053	1.263	0.105	0.158	1.161	19.5	39.842	2.184	0.026
2	Brentford	38	34.211	18.421	47.368	46	1.21	1.263	1.474	-0.211	1.205	1.276	-0.07	29	25.4	44.8	1.053	1.082	0.158	1.196	11.842	31.289	1.711	0.079	1.289	0.132	0.132	1.174	19.316	40.079	1.316	0.079
3	Brighton	38	31.579	39.474	28.947	51	1.34	1.105	1.158	-0.053	1.216	1.129	0.09	26	26.2	54.4	0.947	1.071	0.105	1.21	17.789	40.711	1.895	0.053	1.079	0.079	0.105	1.047	16.895	33.237	2.105	0.0
4	Burnley	38	18.421	36.842	44.737	35	0.92	0.895	1.395	-0.5	1.045	1.503	-0.46	23	28.5	40.2	0.816	1.003	0.026	1.043	10.211	25.579	1.789	0.053	1.211	0.132	0.132	1.4	22.763	45.026	1.158	0.0
5	Chelsea	38	55.263	28.947	15.789	74	1.95	2.0	0.868	1.132	1.668	0.874	0.79	26	27.4	61.8	1.763	1.489	0.211	1.656	23.526	51.763	1.658	0.026	0.684	0.158	0.158	0.761	13.026	26.421	1.816	0.105
6	Palace	38	28.947	39.474	31.579	48	1.26	1.316	1.211	0.105	1.221	1.071	0.15	24	27.1	51.1	1.158	1.055	0.158	1.212	15.579	32.342	1.816	0.026	1.079	0.079	0.105	0.989	14.816	36.895	1.947	0.079
7	Everton	38	28.947	15.789	55.263	39	1.03	1.132	1.737	-0.605	1.084	1.458	-0.37	33	26.7	40.0	0.947	0.942	0.158	1.075	13.474	28.947	2.158	0.158	1.553	0.053	0.053	1.418	19.342	47.342	1.895	0.0
8	Leeds United	38	23.684	28.947	47.368	38	1.0	1.105	2.079	-0.974	1.168	1.784	-0.62	29	26.1	52.3	0.974	1.063	0.132	1.16	17.658	39.026	2.658	0.079	1.921	0.132	0.132	1.684	18.421	35.763	1.947	0.079
9	Leicester City	38	36.842	26.316	36.842	52	1.37	1.632	1.553	0.079	1.258	1.561	-0.3	28	26.5	52.0	1.579	1.216	0.053	1.255	16.368	35.684	1.447	0.026	1.342	0.184	0.237	1.384	19.263	39.368	1.579	0.0
10	Liverpool	38	73.684	21.053	5.263	92	2.42	2.474	0.684	1.789	2.334	0.889	1.45	27	27.7	62.7	2.289	2.176	0.184	2.323	23.289	59.026	1.316	0.026	0.658	0.0	0.0	0.889	12.342	24.0	1.658	0.079
11	Man City	38	76.316	15.789	7.895	93	2.45	2.605	0.684	1.921	2.334	0.647	1.68	26	27.0	67.9	2.342	2.142	0.184	2.323	30.132	63.711	1.105	0.026	0.658	0.026	0.026	0.629	10.316	19.921	1.789	0.079
12	Man Utd	38	42.105	26.316	31.579	58	1.53	1.5	1.5	0.0	1.468	1.395	0.07	29	27.2	52.7	1.395	1.366	0.079	1.463	20.632	41.263	2.026	0.053	1.342	0.079	0.132	1.3	17.684	36.947	1.263	0.026
13	Newcastle	38	34.211	26.316	39.474	49	1.29	1.158	1.632	-0.474	1.003	1.503	-0.5	29	27.5	40.4	1.0	0.945	0.079	0.998	14.316	24.289	2.132	0.053	1.447	0.132	0.158	1.384	20.368	45.395	1.974	0.079
14	Norwich City	38	13.158	18.421	68.421	22	0.58	0.605	2.211	-1.605	0.85	1.987	-1.14	28	26.2	42.9	0.474	0.789	0.079	0.845	14.895	27.132	1.5	0.026	1.921	0.237	0.316	1.745	21.632	46.658	1.632	0.053
15	Southampton	38	23.684	34.211	42.105	40	1.05	1.132	1.763	-0.632	1.179	1.579	-0.4	25	25.7	47.7	0.974	1.095	0.105	1.173	19.158	34.895	1.711	0.053	1.658	0.053	0.053	1.539	17.026	39.395	1.684	0.026
16	Tottenham	38	57.895	13.158	28.947	71	1.87	1.816	1.053	0.763	1.611	1.034	0.58	25	26.5	51.8	1.526	1.508	0.132	1.603	20.816	38.395	1.816	0.026	0.974	0.053	0.053	0.992	17.184	37.974	2.184	0.158
17	Watford	38	15.789	13.158	71.053	23	0.61	0.895	2.026	-1.132	1.053	1.697	-0.64	30	28.4	40.7	0.842	1.011	0.026	1.051	14.658	28.158	1.658	0.079	1.789	0.158	0.184	1.553	19.579	45.842	1.684	0.026
18	West Ham	38	42.105	21.053	36.842	56	1.47	1.579	1.342	0.237	1.353	1.408	-0.06	25	28.2	47.9	1.421	1.232	0.105	1.347	17.0	38.132	1.368	0.079	1.237	0.053	0.158	1.284	22.237	40.421	1.289	0.079
19	Wolves	38	39.474	15.789	44.737	51	1.34	1.0	1.132	-0.132	0.987	1.497	-0.51	26	27.3	49.5	0.895	0.968	0.026	0.986	19.026	33.763	1.632	0.053	0.868	0.237	0.263	1.295	20.526	40.0	2.368	0.079

Рисунок 24 Таблиця даних команд за сезон 2021/22

	team	games	w	t	l	p	p p90	gf	ga	gd	xg for	xg ag	xgd	pl used	avg age	poss	nPg for	pS for	pA for	nPxg for	pr carr for	pr pass for	y cards for	r cards for	nPg ag	pS ag	pA ag	nPxg ag	pr carr ag	pr pass ag	y cards ag	r cards ag
0	Arsenal	38	68.421	15.789	15.789	84	2.21	2.316	1.132	1.184	1.892	1.105	0.79	26	24.7	59.3	2.132	1.818	0.079	1.887	21.684	53.921	1.342	0.0	1.026	0.079	0.132	1.0	13.421	23.868	2.211	0.026
1	Villa	38	47.368	18.421	34.211	61	1.61	1.342	1.211	0.132	1.321	1.382	-0.06	26	27.0	49.3	1.211	1.242	0.079	1.316	16.763	32.684	2.105	0.026	0.974	0.132	0.158	1.261	15.974	34.816	2.237	0.079
2	Bournemouth	38	28.947	15.789	55.263	39	1.03	0.974	1.868	-0.895	1.016	1.682	-0.67	31	26.3	40.4	0.974	1.016	0.0	1.016	13.263	26.474	1.789	0.0	1.632	0.158	0.184	1.537	21.316	52.316	1.342	0.0
3	Brentford	38	39.474	36.842	23.684	59	1.55	1.526	1.211	0.316	1.495	1.313	0.18	25	26.2	43.8	1.289	1.332	0.184	1.484	10.316	29.763	1.5	0.026	1.079	0.053	0.053	1.282	19.5	40.526	1.605	0.026
4	Brighton	38	47.368	21.053	31.579	62	1.63	1.895	1.395	0.5	1.929	1.321	0.61	29	26.3	60.2	1.632	1.813	0.158	1.92	21.289	48.658	1.526	0.0	1.079	0.158	0.211	1.15	13.868	27.0	2.421	0.026
5	Chelsea	38	28.947	28.947	42.105	44	1.16	1.0	1.237	-0.237	1.303	1.382	-0.08	32	26.3	58.7	0.895	1.242	0.079	1.298	21.816	45.711	2.132	0.079	1.132	0.026	0.026	1.361	15.605	31.658	1.816	0.026
6	Palace	38	28.947	31.579	39.474	45	1.18	1.053	1.289	-0.237	1.034	1.266	-0.23	26	26.7	46.3	0.974	0.968	0.026	1.032	14.632	32.711	2.158	0.079	1.105	0.105	0.105	1.182	15.526	41.895	2.184	0.079
7	Everton	38	21.053	31.579	47.368	36	0.95	0.895	1.5	-0.605	1.189	1.729	-0.54	28	26.6	42.8	0.763	1.126	0.079	1.184	13.158	27.684	2.132	0.053	1.368	0.132	0.158	1.608	19.789	44.158	2.0	0.026
8	Fulham	38	39.474	18.421	42.105	52	1.37	1.447	1.395	0.053	1.216	1.679	-0.46	29	28.2	48.8	1.237	1.029	0.132	1.208	17.421	34.895	2.105	0.026	1.158	0.184	0.211	1.524	14.684	37.0	1.5	0.132
9	Leeds United	38	18.421	26.316	55.263	31	0.82	1.263	2.053	-0.789	1.247	1.768	-0.52	29	25.3	47.0	1.158	1.184	0.026	1.246	14.605	37.605	2.342	0.079	1.895	0.132	0.158	1.645	15.684	38.658	1.737	0.079
10	Leicester City	38	23.684	18.421	57.895	34	0.89	1.342	1.789	-0.447	1.332	1.668	-0.34	28	26.1	47.7	1.211	1.205	0.079	1.327	15.263	34.316	1.658	0.053	1.553	0.105	0.132	1.576	18.974	41.447	1.684	0.053
11	Liverpool	38	50.0	26.316	23.684	67	1.76	1.974	1.237	0.737	1.911	1.339	0.57	28	27.4	60.8	1.816	1.829	0.053	1.908	18.921	52.421	1.5	0.026	1.132	0.053	0.105	1.266	13.289	28.974	1.553	0.026
12	Man City	38	73.684	13.158	13.158	89	2.34	2.474	0.868	1.605	2.068	0.845	1.22	24	27.2	64.7	2.184	1.871	0.237	2.054	25.974	54.553	1.158	0.026	0.763	0.079	0.079	0.784	11.421	22.632	2.105	0.053
13	Man Utd	38	60.526	15.789	23.684	75	1.97	1.526	1.132	0.395	1.782	1.326	0.45	26	26.9	53.7	1.395	1.718	0.079	1.777	18.711	41.974	2.053	0.053	1.053	0.053	0.079	1.263	18.342	35.658	1.316	0.0
14	Newcastle	38	50.0	36.842	13.158	71	1.87	1.789	0.868	0.921	1.895	1.042	0.85	27	27.3	52.3	1.526	1.774	0.158	1.886	18.263	42.579	1.605	0.026	0.763	0.053	0.132	0.939	15.789	31.605	2.105	0.053
15	Forest	38	23.684	28.947	47.368	38	1.0	1.0	1.789	-0.789	1.034	1.689	-0.66	33	26.5	37.6	0.868	0.911	0.079	1.029	11.447	22.605	2.211	0.0	1.632	0.158	0.237	1.513	21.395	50.105	1.711	0.0
16	Southampton	38	15.789	18.421	65.789	25	0.66	0.947	1.921	-0.974	0.992	1.605	-0.61	36	24.7	44.5	0.895	0.908	0.053	0.989	15.474	29.605	1.921	0.0	1.737	0.053	0.079	1.545	17.158	45.658	1.658	0.053
17	Tottenham	38	47.368	15.789	36.842	60	1.58	1.842	1.658	0.184	1.503	1.308	0.2	29	27.6	50.0	1.658	1.379	0.132	1.495	18.211	38.132	2.026	0.079	1.553	0.079	0.105	1.226	18.026	39.895	2.079	0.053
18	West Ham	38	28.947	18.421	52.632	40	1.05	1.105	1.447	-0.342	1.295	1.395	-0.1	25	28.2	42.1	0.921	1.15	0.158	1.286	16.158	33.526	1.158	0.0	1.316	0.105	0.158	1.274	21.184	45.289	1.421	0.0
19	Wolves	38	28.947	21.053	50.0	41	1.08	0.816	1.526	-0.711	0.968	1.576	-0.61	32	26.7	49.9	0.658	0.908	0.079	0.963	16.053	35.342	2.237	0.158	1.447	0.053	0.105	1.492	18.474	42.0	1.974	0.0

Рисунок 25 Таблиця даних команд за сезон 2022/23