НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

**КУРСОВА РОБОТА**

з дисципліни «Аналіз даних в інформаційних системах»

на тему: «Аналіз гоночної серії FORMULA 1»

Студентів 2 курсу групи ІП-21

Спеціальності: 121

«Інженерія програмного забезпечення»

Загребельний О.А.

Голованьов Г.О.

«ПРИЙНЯВ» з оцінкою

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доц. Ліхоузова Т.А. / доц. Олійник Ю.О.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Підпис                    Дата

Київ - 2024 рік

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

Кафедра інформатики та програмної інженерії

Дисципліна Аналіз даних в інформаційно-управляючих системах

Спеціальність 121 "Інженерія програмного забезпечення"

Курс 2 Група ІП-21 Семестр 4

**ЗАВДАННЯ**

**на курсову роботу студента**

|  |
| --- |
| Загребельного О.А., Голованьова Г.О. |

|  |  |
| --- | --- |
| 1.Тема роботи | Аналіз гоночної серії FORMULA 1 |
|  | |
|  | |

|  |  |
| --- | --- |
| 2.Строк здачі студентом закінченої роботи | 29.05.2024 |

|  |  |
| --- | --- |
| 3. Вхідні дані до роботи | методичні вказівки до курсової роботи, обрані дані з сайтів |
| https://github.com/f1db/f1db | |
| https://docs.fastf1.dev/ | |

4.Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

|  |
| --- |
| 1.Постановка задачі |
| 2.Аналіз предметної області |
| 3.Розробка сховища даних |
| 4.Інтелектуальний аналіз даних |

5.Перелік графічного матеріалу ( з точним зазначенням обов’язкових креслень )

|  |  |
| --- | --- |
|  | |
|  | |
|  | |
|  | |
| 6.Дата видачі завдання | 30.03.2024 |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Назва етапів курсової роботи | Термін виконання етапів роботи | Підписи керівника, студента |
| 1. | Отримання теми курсової роботи | 30.03.2024 |  |
| 2. | Визначення зовнішніх джерел даних | 06.04.2024 |  |
| 3. | Пошук та вивчення літератури з питань курсової роботи | 13.04.2024 |  |
| 4. | Розробка моделі сховища даних | 20.04.2024 |  |
| 4. | Розробка ETL процесів | 27.04.2024 |  |
| 5. | Обґрунтування методів інтелектуального аналізу даних | 01.05.2024 |  |
| 6. | Застосування та порівняння ефективності методів інтелектуального аналізу даних | 15.05.2024 |  |
| 7. | Підготовка пояснювальної записки | 22.05.2024 |  |
| 8. | Здача курсової роботи на перевірку | 29.05.2024 |  |
| 9. | Захист курсової роботи | 03.06.2024 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  |  | Загребельний О.А. |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |
| Студент | |  |  | Голованьов Г.О. |
|  | | (підпис) |  | (прізвище, ім’я, по батькові) |

|  |  |  |  |
| --- | --- | --- | --- |
| Керівник |  |  | доц. Ліхоузова Т.А |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |
| Керівник |  |  | доц. Олійник Ю.О. |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |

"29" травня 2024 р.

**АНОТАЦІЯ**

Пояснювальна записка до курсової роботи: \_\_ сторінок, \_\_ рисунки, \_\_ таблиць, \_\_ посилань.

Об’єкт дослідження: інтелектуальний аналіз даних.

Предмет дослідження: створення програмного забезпечення, що проводить аналіз даних з подальшим прогнозуванням та графічним відображенням результатів.

Мета роботи: проектування та реалізація сховища даних та ETL процесів, а також реалізація програмного забезпечення для отримання даних зі сховища та їх подальшого аналізу та прогнозування.

Дана курсова робота включає в себе: опис проектування, створення та заповнення сховища даних за даною задачею за допомогою фізичної моделі бази даних, опис створення програмного забезпечення для інтелектуального аналізу даних, їх графічного відображення та прогнозування за допомогою різних моделей.

СХОВИЩЕ ДАННИХ, МОДЕЛЬ ПРОГНОЗУВАННЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ФІЗИЧНА МОДЕЛЬ БАЗИ ДАНИХ, ELT ПРОЦЕСИ, МОДЕЛЬ SARIMAX, ЧАСОВИЙ ПРОМІЖОК, ДОВІРЧІ ІНТЕРВАЛИ, МОДЕЛЬ ARIMA.

**Зміст**

[вступ 5](#_Toc419641933)

[1.Постановка задачі 7](#_Toc419641934)

[2.Аналіз ПРЕДМЕТНОЇ ОБЛАСТІ 9](#_Toc419641935)

[3.розробка сховища даних 11](#_Toc419641936)

[3.1.РОЗРОБКА МОДЕЛІ СХОВИЩА ДАНИХ……... 11](#_Toc419641937)

[3.2.РОЗРОБКА ETL ПРОЦЕСІВ 19](#_Toc419641938)

[3.3.ЗАВАНТАЖЕННЯ ДАНИХ ЗА ДОПОМОГОЮ ETL ПРОЦЕСІВ…. 20](#_Toc419641938)

[4.ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ 37](#_Toc419641939)

[4.1.ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ 37](#_Toc419641940)

[4.2. МАТЕМАТИЧНА МОДЕЛЬ ПРОГНОЗУВАННЯ ЧАСОВИХ ПРОМІЖКІВ 38](#_Toc419641940)

[4.3.ПРАКТИЧНЕ ЗАСТОСУВАННЯ МОДЕЛЕЙ ПРОГНОЗУВАННЯ ЧАСОВИХ ПРОМІЖКІВ 40](#_Toc419641940)

4.4.ПОРІВНЯННЯ ЕФЕКТИВНОСТІ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ 56

[Висновки 58](#_Toc419641949)

[Перелік посилань 60](#_Toc419641950)

ДОДАТок А ТЕКСТИ ПРОГРАМНОГО КОДУ 61

**Вступ**

У сучасному світі автоспорт, зокрема серія перегонів Formula 1, є одним із найпопулярніших та найпрестижніших видів спорту, що привертає увагу мільйонів глядачів по всьому світу. Формула 1 не лише захоплює своїми динамічними гонками та високотехнологічними автомобілями, але й приносить величезні прибутки та стимулює розвиток інновацій у різних галузях. Щорічно серія перегонів Formula 1 привертає нових вболівальників та встановлює нові рекорди за відвідуваністю та переглядами.

У рамках даної курсової роботи розроблено сховище даних, яке містить інформацію про результати перегонів Formula 1, базуючись на фізичній моделі бази даних. Основу сховища було створено за допомогою SQL скриптів, а робота з базою даних реалізована через програмне забезпечення для імплементації ETL процесів та інтелектуального аналізу даних.

Як система керування сховищем даних для даної роботи використовується PostgreSQL, а мова програмування для реалізації застосунку - Python3. Використовуючи ці інструменти, було розроблено програмне забезпечення, яке дозволяє отримувати дані зі сховища, проводити їх аналіз, прогнозувати результати перегонів та графічно відображати отримані результати.

Ця робота спрямована на дослідження та аналіз даних перегонів Formula 1, що дозволить не лише зрозуміти поточні тенденції, але й зробити прогнози щодо майбутніх перегонів, використовуючи різні моделі прогнозування, такі як K-nearest neighbours та Random Forest.

**1.Постановка задачі**

Під час виконання курсової роботи необхідно виконати комплекс завдань, спрямованих на створення ефективного інструменту для аналізу та прогнозування результатів перегонів Formula 1. Насамперед, потрібно створити сховище даних типу «сніжинка», яке міститиме щонайменше шість таблиць вимірів та одну таблицю фактів.

Наступним кроком є створення ETL процесів для завантаження даних до сховища, їх отримання зі сховища за допомогою запитів, а також оновлення та додавання даних до таблиць.

Реалізація спроектованого сховища даних буде здійснюватися з використанням PostgreSQL, що забезпечить високу продуктивність та надійність системи. Для подальшого аналізу даних необхідно створити програмне забезпечення, яке дозволить отримувати вибірки даних зі створеного сховища, графічно відображати їх, проводити інтелектуальний аналіз та здійснювати передбачення за допомогою різних моделей прогнозування.

Аналіз вибірки даних передбачає перевірку на сезонність, стаціонарність та шум за допомогою доповненого тесту Дікі-Фуллера, а також декомпозицію даних. Для покращення якості прогнозів необхідно провести згладжування даних методами ковзного середнього, вейвлет-згладжування та швидкого згладжування Фур'є.

У роботі будуть використовуватись моделі прогнозування, такі як k-nearest neighbours та random forest, для аналізу даних. Важливо застосувати найоптимальнішу модель прогнозування на згладжених даних та порівняти отримані результати з прогнозуванням на початкових даних.

Додатково, необхідно провести автокореляцію та часткову автокореляцію для виявлення повторюваних ділянок даних, незважаючи на наявність шумів. Аналіз отриманих результатів та порівняння різних методів прогнозування дозволить визначити найоптимальніший метод. Важливим аспектом є знаходження довірчих інтервалів з використанням найкращої моделі прогнозування.

Для реалізації застосунку буде використовуватись мова програмування Python 3. Весь проект повинен бути завершений до дедлайну, початку сесії, з дотриманням єдиного стилю написання коду (coding style), що забезпечить його читабельність та підтримуваність у майбутньому.

**2.АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

Сьогодні автоспорт, зокрема серія перегонів Formula 1, є одним із найпопулярніших видів спорту у світі, який постійно привертає увагу мільйонів глядачів. Formula 1, або "Королева автоспорту", не тільки захоплює своїми видовищними гонками та інноваційними технологіями, але й є значущою складовою світової економіки, генеруючи величезні доходи від трансляцій, спонсорства та продажу квитків. Гонки Formula 1 не лише пропонують захоплююче видовище, але й виступають платформою для впровадження новітніх технологій та інженерних рішень.

Аналіз результатів перегонів Formula 1 є складним завданням, яке включає багато факторів, таких як погодні умови, характеристики трас, технічний стан автомобілів та майстерність пілотів. Для команд та аналітиків важливо виявляти закономірності, що впливають на результати перегонів, аби оптимізувати стратегії, покращувати результати та передбачати майбутні успіхи. Вміння прогнозувати результати перегонів на основі історичних даних дозволяє командам краще готуватися до майбутніх змагань та приймати обґрунтовані рішення щодо розвитку техніки і тактики.

У рамках даної курсової роботи буде розроблено програмне забезпечення для аналізу та прогнозування результатів перегонів Formula 1. Функціональність цього програмного забезпечення включає:

* проектування сховища даних;
* створення ETL процесів для завантаження та оновлення даних;
* створення вибірки даних зі сховища;
* інтелектуальний аналіз даних;
* використання декількох моделей прогнозування даних;
* прогнозування результатів перегонів;
* визначення факторів, що впливають на результати перегонів;
* графічне відображення отриманих результатів та їх аналіз.

Сховище даних дозволить зберігати великі обсяги історичних даних про результати перегонів, що забезпечить основу для детального аналізу та прогнозування. ETL процеси забезпечать актуальність та повноту даних, необхідних для аналітичної роботи. Інтелектуальний аналіз даних, включаючи перевірку на сезонність, стаціонарність та шум, а також згладжування даних, дозволить виявити приховані тенденції та шаблони. Використання моделей прогнозування, таких як k-nearest neighbours та random forest, дозволить робити точні передбачення та оцінювати вплив різних факторів на результати перегонів. Графічне відображення результатів допоможе візуалізувати дані та зрозуміти основні висновки аналізу.

**3.РОЗРОБКА СХОВИЩА ДАНИХ**

3.1 Розробка моделі сховища даних

Для виконання курсової роботи було обрано 2 джерела відкритих даних на сайті https://github.com , що включають в себе:

* Загальна інформація про перегони:   
  https://github.com/f1db/f1db
* Інформація про час кожного гонщика на всіх гран-прі: https://docs.fastf1.dev/

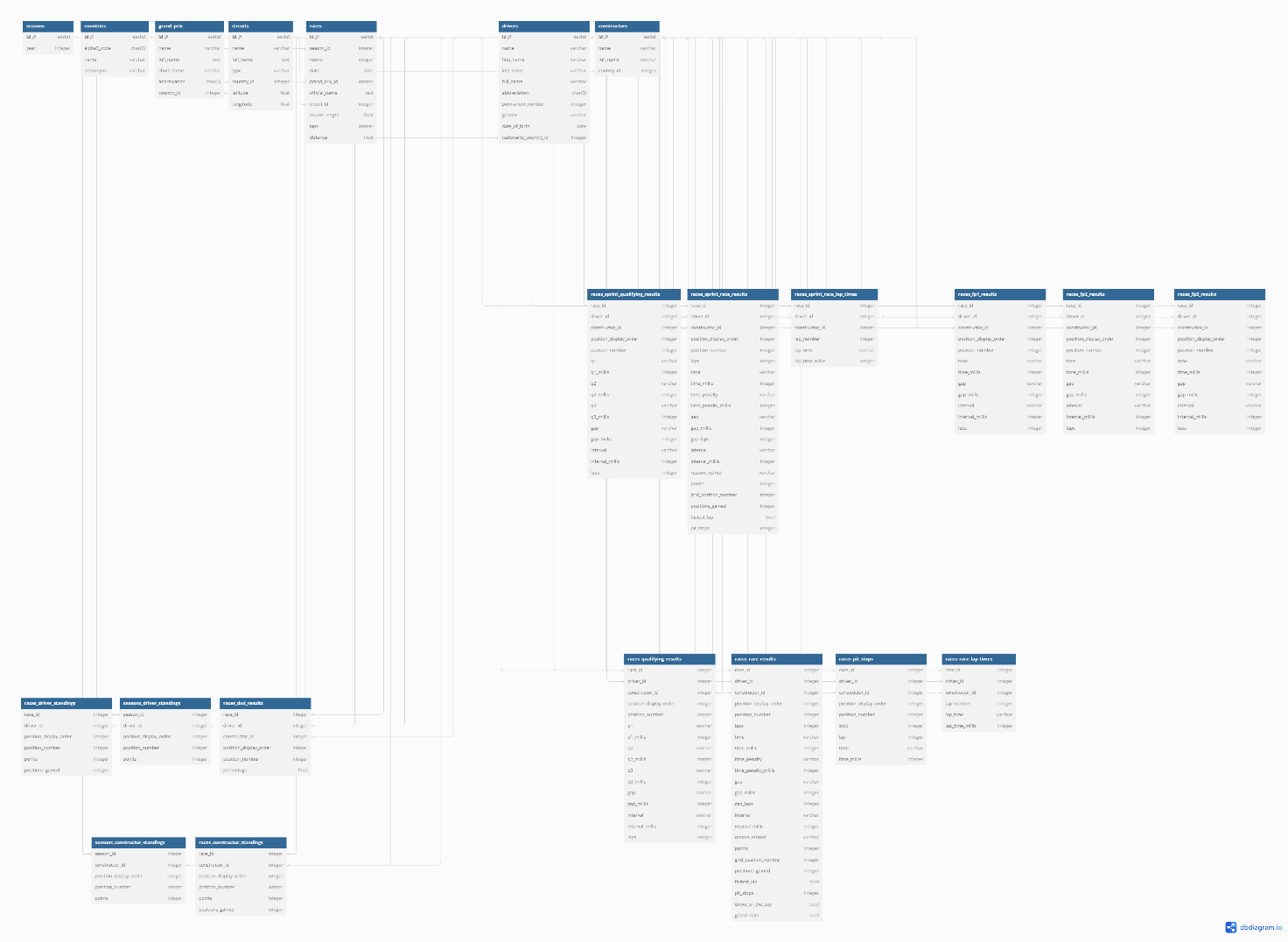
На основі детального опису та проведеного аналізу предметної області було розроблено наступну модель сховища даних гоночної серії Формула-1 за типом „сніжинка” (рисунок 3.1).

Рисунок 3.1 - Проектування моделі сховища за типом сніжинка

У моделі сховища за типом сніжинка спроектовано 15 таблиці фактів та 7 таблиць вимірів. Нижче наведено опис полів кожної таблиці сховища даних гоночної серії Формула-1 (таблиці 3.1-3.22).

Таблиця 3.1 - Таблиця виміру сезони **seasons**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| id | Унікальний ідентифікатор сезону (автоматично згенерований) |
| year | Рік сезону |

Таблиця 3.2 - Таблиця виміру країни **countries**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| id | Унікальний ідентифікатор країни (автоматично згенерований) |
| alpha3\_code | Трилітерний код країни |
| name | Назва країни |
| demonym | Демонім (назва для жителів країни) |

Таблиця 3.3 - Таблиця виміру кіл **circuits**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| id | Унікальний ідентифікатор траси (автоматично згенерований) |
| name | Назва траси |
| full\_name | Повна назва траси |
| type | Тип траси |
| country\_id | Зовнішній ключ, що посилається на id країни |
| latitude | Широта |
| longitude | Довгота |

Таблиця 3.4 - Таблиця виміру конструкторів **constructors**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| id | Унікальний ідентифікатор конструктора (автоматично згенерований) |
| name | Назва конструктора |
| full\_name | Повна назва конструктора |
| country\_id | Зовнішній ключ, що посилається на id країни |

Таблиця 3.5 - Таблиця виміру пілотів **drivers**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| id | Унікальний ідентифікатор гонщика (автоматично згенерований) |
| name | Прізвище гонщика |
| first\_name | Ім'я гонщика |
| last\_name | По батькові гонщика |
| full\_name | Повне ім'я гонщика |
| abbreviation | Абревіатура |
| permanent\_number | Постійний номер гонщика |
| gender | Стать |
| date\_of\_birth | Дата народження |
| nationality\_country\_id | Зовнішній ключ, що посилається на id країни громадянства гонщика |

Таблиця 3.6 - Таблиця виміру гран-прі **grand\_prix**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| id | Унікальний ідентифікатор гран-прі (автоматично згенерований) |
| name | Назва гран-прі |
| full\_name | Повна назва гран-прі |
| short\_name | Скорочена назва гран-прі |
| abbreviation | Абревіатура |
| country\_id | Зовнішній ключ, що посилається на id країни |

Таблиця 3.7 - Таблиця виміру гонки **races**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| id | Унікальний ідентифікатор гонки (автоматично згенерований) |
| season\_id | Зовнішній ключ, що посилається на id сезону |
| round | Номер етапу |
| date | Дата гонки |
| grand\_prix\_id | Зовнішній ключ, що посилається на id гран-прі |
| official\_name | Офіційна назва |
| circuit\_id | Зовнішній ключ, що посилається на id траси |
| course\_length | Довжина траси |
| laps | Кількість кіл |
| distance | Дистанція |

Таблиця 3.8 - Таблиця виміру гонки в першій вільній практиці **races\_fp1\_results**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| time | Час гонки |
| time\_millis | Час у мілісекундах |
| gap | Різниця в часі до лідера |
| gap\_millis | Різниця в часі до лідера у мілісекундах |
| interval | Інтервал між гонками |
| interval\_millis | Інтервал між гонками у мілісекундах |
| laps | Кількість кіл, що пройдено |

Таблиця 3.9 - Таблиця виміру гонки в другій вільній практиці **races\_fp2\_results**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| time | Час гонки |
| time\_millis | Час у мілісекундах |
| gap | Різниця в часі до лідера |
| gap\_millis | Різниця в часі до лідера у мілісекундах |
| interval | Інтервал між гонками |
| interval\_millis | Інтервал між гонками у мілісекундах |
| laps | Кількість кіл, що пройдено |

Таблиця 3.10 - Таблиця виміру гонки в третій вільній практиці **races\_fp3\_results**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| time | Час гонки |
| time\_millis | Час у мілісекундах |
| gap | Різниця в часі до лідера |
| gap\_millis | Різниця в часі до лідера у мілісекундах |
| interval | Інтервал між гонками |
| interval\_millis | Інтервал між гонками у мілісекундах |
| laps | Кількість кіл, що пройдено |

Таблиця 3.11 - Таблиця виміру результати гоночної кваліфікації **races\_qualifying\_results**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| q1 | Час в кваліфікації 1 |
| q1\_millis | Час в кваліфікації 1 у мілісекундах |
| q2 | Час в кваліфікації 2 |
| q2\_millis | Час в кваліфікації 2 у мілісекундах |
| q3 | Час в кваліфікації 3 |
| q3\_millis | Час в кваліфікації 3 у мілісекундах |
| gap | Різниця в часі до лідера |
| gap\_millis | Різниця в часі до лідера у мілісекундах |
| interval | Інтервал між гонками |
| interval\_millis | Інтервал між гонками у мілісекундах |
| laps | Кількість кіл, що пройдено |

Таблиця 3.12 - Таблиця виміру результати спринтової кваліфікації **races\_sprint\_qualifying\_results**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| q1 | Час в кваліфікації 1 |
| q1\_millis | Час в кваліфікації 1 у мілісекундах |
| q2 | Час в кваліфікації 2 |
| q2\_millis | Час в кваліфікації 2 у мілісекундах |
| q3 | Час в кваліфікації 3 |
| q3\_millis | Час в кваліфікації 3 у мілісекундах |
| gap | Різниця в часі до лідера |
| gap\_millis | Різниця в часі до лідера у мілісекундах |
| interval | Інтервал між гонками |
| interval\_millis | Інтервал між гонками у мілісекундах |
| laps | Кількість кіл, що пройдено |

Таблиця 3.13 - Таблиця виміру результати гонки **races\_race\_results**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| laps | Кількість пройдених кол |
| time | Час гонки |
| time\_millis | Час гонки у мілісекундах |
| time\_penalty | Час штрафу |
| time\_penalty\_millis | Час штрафу у мілісекундах |
| gap | Різниця в часі до лідера |
| gap\_millis | Різниця в часі до лідера у мілісекундах |
| gap\_laps | Кількість кол відставання |
| interval | Інтервал до наступного гонщика |
| interval\_millis | Інтервал до наступного гонщика у мілісекундах |
| reason\_retired | Причина виходу з гонки |
| points | Очки |
| grid\_position\_number | Позиція на старті |
| positions\_gained | Кількість місць, що покращено за гонку |
| fastest\_lap | Прапорець, що позначає найшвидший круг |
| pit\_stops | Кількість зупинок на боксах |
| driver\_of\_the\_day | Прапорець, що позначає гонщика дня |
| grand\_slam | Прапорець, що позначає виграш "гранд-слему" |

Таблиця 3.14 - Таблиця виміру результати спринту **races\_sprint\_race\_results**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| laps | Кількість пройдених кол |
| time | Час гонки |
| time\_millis | Час гонки у мілісекундах |
| time\_penalty | Час штрафу |
| time\_penalty\_millis | Час штрафу у мілісекундах |
| gap | Різниця в часі до лідера |
| gap\_millis | Різниця в часі до лідера у мілісекундах |
| gap\_laps | Кількість кол відставання |
| interval | Інтервал до наступного гонщика |
| interval\_millis | Інтервал до наступного гонщика у мілісекундах |
| reason\_retired | Причина виходу з гонки |
| points | Очки |
| grid\_position\_number | Позиція на старті |
| positions\_gained | Кількість місць, що покращено за гонку |
| fastest\_lap | Прапорець, що позначає найшвидший круг |
| pit\_stops | Кількість зупинок на боксах |
| driver\_of\_the\_day | Прапорець, що позначає гонщика дня |
| grand\_slam | Прапорець, що позначає виграш "гранд-слему" |

Таблиця 3.15 - Таблиця виміру результати піт-стопів **races\_pit\_stops**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| stop | Номер зупинки |
| lap | Номер круга, на якому відбулась зупинка |
| time | Час зупинки |
| time\_millis | Час зупинки у мілісекундах |

Таблиця 3.16 - Таблиця виміру колові часи у спринті **races\_sprint\_race\_lap\_times**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| lap\_number | Номер круга |
| lap\_time | Час проходження круга |
| lap\_time\_millis | Час проходження круга у мілісекундах |

Таблиця 3.17 - Таблиця виміру колові часи у гонці **races\_race\_lap\_times**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| lap\_number | Номер круга |
| lap\_time | Час проходження круга |
| lap\_time\_millis | Час проходження круга у мілісекундах |

Таблиця 3.18 - Таблиця виміру позиції конструкторів під час певної гонки **races\_constructor\_standings**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| points | Кількість очок |
| positions\_gained | Кількість позицій, які були покращені |

Таблиця 3.19 - Таблиця виміру гонщик дня (Drive Of the Day) **races\_dod\_results**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| percentage | Відсоток голосів за гонщика дня |

Таблиця 3.20 - Таблиця виміру позиції гонщиків під час певної гонки **races\_driver\_standings**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| race\_id | Зовнішній ключ, що посилається на id гонки |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| points | Кількість очок |
| positions\_gained | Кількість позицій, які були покращені |

Таблиця 3.21 - Таблиця виміру позиції конструкторів після завершення сезону **seasons\_constructor\_standings**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| season\_id | Зовнішній ключ, що посилається на id сезону |
| constructor\_id | Зовнішній ключ, що посилається на id конструктора |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| points | Кількість очок |

Таблиця 3.22 - Таблиця виміру позиції гонщиків після завершення сезону **seasons\_driver\_standings**:

|  |  |
| --- | --- |
| **Поле** | **Пояснення** |
| season\_id | Зовнішній ключ, що посилається на id сезону |
| driver\_id | Зовнішній ключ, що посилається на id гонщика |
| position\_display\_order | Порядковий номер позиції |
| position\_number | Номер позиції |
| points | Кількість очок |

На основі спроектованої моделі було створено сховище даних гоночної серії Формула-1, що включає 7 таблиць вимірів та 15 таблиці фактів реалізованих за допомогою нижче наведених скриптів використовуючи PostgreSQL версії 12.1.

Створення таблиці виміру сезони в сховищі даних:

CREATE TABLE seasons (

    id SERIAL PRIMARY KEY,

    year INTEGER

);

Створення таблиці виміру країни в сховищі даних:

CREATE TABLE countries (

    id SERIAL PRIMARY KEY,

    alpha3\_code CHAR(3),

    name VARCHAR,

    demonym VARCHAR

);

Створення таблиці виміру кола в сховищі даних:

CREATE TABLE circuits (

    id SERIAL PRIMARY KEY,

    name VARCHAR,

    full\_name TEXT,

    type VARCHAR,

    country\_id INTEGER REFERENCES countries(id),

    latitude FLOAT,

    longitude FLOAT

);

Створення таблиці виміру конструктори в сховищі даних:

CREATE TABLE constructors (

    id SERIAL PRIMARY KEY,

    name VARCHAR,

    full\_name VARCHAR,

    country\_id INTEGER REFERENCES countries(id)

);

Створення таблиці виміру пілоти в сховищі даних:

CREATE TABLE drivers (

    id SERIAL PRIMARY KEY,

    name VARCHAR,

    first\_name VARCHAR,

    last\_name VARCHAR,

    full\_name VARCHAR,

    abbreviation CHAR(3),

    permanent\_number INTEGER,

    gender VARCHAR,

    date\_of\_birth DATE,

    nationality\_country\_id INTEGER REFERENCES countries(id)

);

Створення таблиці виміру гран-прі в сховищі даних:

CREATE TABLE grand\_prix (

    id SERIAL PRIMARY KEY,

    name VARCHAR,

    full\_name TEXT,

    short\_name VARCHAR,

    abbreviation CHAR(3),

    country\_id INTEGER REFERENCES countries(id)

);

Створення таблиці виміру гонки в сховищі даних:

CREATE TABLE races (

    id SERIAL PRIMARY KEY,

    season\_id INTEGER REFERENCES seasons(id),

    round INTEGER,

    date DATE,

    grand\_prix\_id INTEGER REFERENCES grand\_prix(id),

    official\_name TEXT,

    circuit\_id INTEGER REFERENCES circuits(id),

    course\_length FLOAT,

    laps INTEGER,

    distance FLOAT

);

Створення таблиці виміру результати першої вільної практики в сховищі даних:

CREATE TABLE races\_fp1\_results (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    time VARCHAR,

    time\_millis INTEGER,

    gap VARCHAR,

    gap\_millis INTEGER,

    interval VARCHAR,

    interval\_millis INTEGER,

    laps INTEGER

);

Створення таблиці виміру результати другої вільної практики в сховищі даних:

CREATE TABLE races\_fp2\_results (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    time VARCHAR,

    time\_millis INTEGER,

    gap VARCHAR,

    gap\_millis INTEGER,

    interval VARCHAR,

    interval\_millis INTEGER,

    laps INTEGER

);

Створення таблиці виміру результати третьої вільної практики в сховищі даних:

CREATE TABLE races\_fp3\_results (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    time VARCHAR,

    time\_millis INTEGER,

    gap VARCHAR,

    gap\_millis INTEGER,

    interval VARCHAR,

    interval\_millis INTEGER,

    laps INTEGER

);

Створення таблиці виміру результати кваліфікації в сховищі даних:

CREATE TABLE races\_qualifying\_results (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    q1 VARCHAR,

    q1\_millis INTEGER,

    q2 VARCHAR,

    q2\_millis INTEGER,

    q3 VARCHAR,

    q3\_millis INTEGER,

    gap VARCHAR,

    gap\_millis INTEGER,

    interval VARCHAR,

    interval\_millis INTEGER,

    laps INTEGER

);

Створення таблиці виміру результати гонки в сховищі даних:

CREATE TABLE races\_race\_results (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    laps INTEGER,

    time VARCHAR,

    time\_millis INTEGER,

    time\_penalty VARCHAR,

    time\_penalty\_millis INTEGER,

    gap VARCHAR,

    gap\_millis INTEGER,

    gap\_laps INTEGER,

    interval VARCHAR,

    interval\_millis INTEGER,

    reason\_retired VARCHAR,

    points INTEGER,

    grid\_position\_number INTEGER,

    positions\_gained INTEGER,

    fastest\_lap BOOLEAN,

    pit\_stops INTEGER,

    driver\_of\_the\_day BOOLEAN,

    grand\_slam BOOLEAN

);

Створення таблиці виміру піт-стопи під час гонки в сховищі даних:

CREATE TABLE races\_pit\_stops (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    stop INTEGER,

    lap INTEGER,

    time VARCHAR,

    time\_millis INTEGER

);

Створення таблиці виміру результати кваліфікації на спринт в сховищі даних:

CREATE TABLE races\_sprint\_qualifying\_results (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    q1 VARCHAR,

    q1\_millis INTEGER,

    q2 VARCHAR,

    q2\_millis INTEGER,

    q3 VARCHAR,

    q3\_millis INTEGER,

    gap VARCHAR,

    gap\_millis INTEGER,

    interval VARCHAR,

    interval\_millis INTEGER,

    laps INTEGER

);

Створення таблиці виміру результат спринт-гонки в сховищі даних:

CREATE TABLE races\_sprint\_race\_results (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    laps INTEGER,

    time VARCHAR,

    time\_millis INTEGER,

    time\_penalty VARCHAR,

    time\_penalty\_millis INTEGER,

    gap VARCHAR,

    gap\_millis INTEGER,

    gap\_laps INTEGER,

    interval VARCHAR,

    interval\_millis INTEGER,

    reason\_retired VARCHAR,

    points INTEGER,

    grid\_position\_number INTEGER,

    positions\_gained INTEGER,

    fastest\_lap BOOLEAN,

    pit\_stops INTEGER

);

Створення таблиці виміру позиції конструкторів в таблиці очків під час гонки в сховищі даних:

CREATE TABLE races\_constructor\_standings (

    race\_id INTEGER REFERENCES races(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    points INTEGER,

    positions\_gained INTEGER

);

Створення таблиці виміру результати пілота дня в сховищі даних:

CREATE TABLE races\_dod\_results (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    percentage FLOAT

);

Створення таблиці виміру позиції пілотів в таблиці очків під час гонки в сховищі даних:

CREATE TABLE races\_driver\_standings (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    points INTEGER,

    positions\_gained INTEGER

);

Створення таблиці виміру позиції конструкторів після завершення сезону в сховищі даних:

CREATE TABLE seasons\_constructor\_standings (

    season\_id INTEGER REFERENCES seasons(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    points INTEGER

);

Створення таблиці виміру позиції пілотів після завершення сезону в сховищі даних:

CREATE TABLE seasons\_driver\_standings (

    season\_id INTEGER REFERENCES seasons(id),

    driver\_id INTEGER REFERENCES drivers(id),

    position\_display\_order INTEGER,

    position\_number INTEGER,

    points INTEGER

);

Створення таблиці виміру результати кіл під час спринт-гонки в сховищі даних:

CREATE TABLE races\_sprint\_race\_lap\_times (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    lap\_number INTEGER,

    lap\_time VARCHAR,

    lap\_time\_millis INTEGER

);

Створення таблиці виміру результати кіл під час гонки в сховищі даних:

CREATE TABLE races\_race\_lap\_times (

    race\_id INTEGER REFERENCES races(id),

    driver\_id INTEGER REFERENCES drivers(id),

    constructor\_id INTEGER REFERENCES constructors(id),

    lap\_number INTEGER,

    lap\_time VARCHAR,

    lap\_time\_millis INTEGER

);

3.2 Розробка ETL процесів

Для проектування сховища даних і подальшої роботи з ним необхідно враховувати набір методів, що реалізують процес перекладу вихідних даних в аналітичний формат, який підтримується сховищем даних і не порушує цілісність системи. Для цього ми обговоримо ключові особливості процесу ETL, включаючи:

**Процес завантаження.**

На цьому етапі ви перетягуєте всі дані про якість в ETL для подальшої обробки. Вхідні дані можуть бути джерелом різних типів і форматів, можуть створюватися в різних додатках і навіть використовувати різні кодування, тому в майбутньому для вирішення проблеми аналізу, запобігання впливу різних факторів, що ускладнюють коректний аналіз даних, дані будуть використовуватися в наступних цілях: зберігається в базі даних. Його необхідно перетворити в єдиний універсальний формат.

**Процес валідації даних.**

Основним завданням на цьому етапі є перевірка точності та повноти даних, після чого складається звіт про помилку для виправлення та узагальнення даних. Основне очищення даних у процесі ETL в основному є технічним, і основним завданням є підготовка даних до завантаження в сховище. Вторинна очистка в аналітичних системах визначається користувачем і призначена для підготовки даних для вирішення конкретних аналітичних задач. З цієї причини обидва етапи очищення однаково важливі і необхідні.

**Перетворення даних.**

Цей етап процесу ETL необхідний для підготовки даних до розміщення в сховищі даних і надання їм найбільш зручного формату для подальшого аналізу. У той же час слід враховувати вимоги до рівня якості даних. Тому в процесі перетворення можна використовувати найрізноманітніші інструменти, які можуть бути як найпростішими засобами ручного редагування даних, так і реалізацією досить складних методів обробки і очищення даних.

**Завантаження даних в сховище.**

Після того, як дані отримані з різних джерел і перетворені, агреговані і очищені, виконується останній етап ETL. Процес завантаження складається з перенесення даних з проміжної таблиці в структуру сховища даних. Таким чином, щоб пришвидшити завантаження даних, вам потрібно заздалегідь перевірити цілісність та точність даних, а потім розробити та вибрати оптимальний процес заповнення сховища. При завантаженні даних в сховище передається не вся інформація з джерела, а тільки інформація, яка змінилася за минулий з моменту попереднього завантаження час. В нашому випадку майже половина даних, які ми отримали з джерела, є непотрібні, тому під час завантаження їх до сховища були виконані роботи щодо цього.

3.3 Завантаження даних за допомогою ETL процесів

Для заповнення сховища даних відеоігор було реалізовано наступне:

1. Було зведено таблиці вимірів і заімпортовані до сховища, а саме:
   1. A screenshot of a computer

      Description automatically generatedCountries (країни)

Рисунок 3.2 – Вхідні дані таблиці **Countries**

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a black background with white text

Description automatically generated

Рисунок 3.3 – Вихідні дані

* 1. Circuits (кола)

A screenshot of a computer

Description automatically generated

Рисунок 3.4 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

Рисунок 3.5 – Вихідні дані

A screenshot of a computer

Description automatically generated

* 1. Constructors (конструктори)

A screenshot of a computer

Description automatically generated

Рисунок 3.6 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

Рисунок 3.7 – Вихідні дані

A screenshot of a computer

Description automatically generated

* 1. Drivers (пілоти)

A black screen with white text

Description automatically generated

Рисунок 3.8 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a computer

Description automatically generated

Рисунок 3.9 – Вихідні дані

* 1. Grand Prix (гран-прі)

A screenshot of a computer

Description automatically generated

Рисунок 3.10 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a computer

Description automatically generated

Рисунок 3.11 – Вихідні дані

* 1. Seasons (сезони)

A screenshot of a cell phone

Description automatically generated

Рисунок 3.12 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a cell phone

Description automatically generated

Рисунок 3.13 – Вихідні дані

* 1. Races (гонки)

A screenshot of a computer

Description automatically generated

Рисунок 3.14 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a computer

Description automatically generated

Рисунок 3.15 – Вихідні дані

б) Було зведено таблиці фактів і заімпортовані до сховища, а саме:

* 1. Races FP1 Results(Результати першої вільної практики)

A screenshot of a computer

Description automatically generated

Рисунок 3.16 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a computer

Description automatically generated

Рисунок 3.17 – Вхідні дані

* 1. Races FP2 Results(Результати другої вільної практики)

A screenshot of a computer

Description automatically generated

Рисунок 3.18 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a computer

Description automatically generated

Рисунок 3.19 – Вхідні дані

* 1. Races FP3 Results(Результати третьої вільної практики)

A screenshot of a computer

Description automatically generated

Рисунок 3.20 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a computer

Description automatically generated

Рисунок 3.21 – Вхідні дані

* 1. Races Qualifying Results(Результати кваліфікації)

A black screen with white text

Description automatically generated

Рисунок 3.22 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A black screen with white text

Description automatically generated

Рисунок 3.23 – Вхідні дані

* 1. Races Race Result(Результати гонки)

A screen shot of a computer

Description automatically generated

Рисунок 3.24 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screen shot of a black screen

Description automatically generated

Рисунок 3.25 – Вхідні дані

* 1. Races Pit Stops(Пітстопи під час гонки)

A black screen with white text

Description automatically generated

Рисунок 3.26 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a computer

Description automatically generated

Рисунок 3.27 – Вхідні дані

* 1. Races Sprint Qualifying Results(Результати кваліфікації на спринт-гонку)

A screenshot of a computer

Description automatically generated

Рисунок 3.28 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A black screen with white text

Description automatically generated

Рисунок 3.29 – Вхідні дані

* 1. Races Sprint Race Results(Результати спринт-гонки)

A screenshot of a computer

Description automatically generated

Рисунок 3.30 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A black screen with white text

Description automatically generated

Рисунок 3.31 – Вхідні дані

* 1. Races Constructor Standings(Позиції конструкторів під час сезону)

A screenshot of a computer

Description automatically generated

Рисунок 3.32 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a computer

Description automatically generated

Рисунок 3.33 – Вхідні дані

* 1. Races Driver Of the Day Results(Пілот дня)

A screenshot of a computer

Description automatically generated

Рисунок 3.34 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a black screen

Description automatically generated

Рисунок 3.35 – Вхідні дані

* 1. Races Driver Standings(Позиція пілотів під час сезону)

A screenshot of a computer

Description automatically generated

Рисунок 3.36 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a black screen

Description automatically generated

Рисунок 3.37 – Вхідні дані

* 1. Seasons Constructor Standing(Позиція конструкторів після завершення сезону)

A screenshot of a computer

Description automatically generated

Рисунок 3.38 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a black screen

Description automatically generated

Рисунок 3.39 – Вхідні дані

* 1. Seasons Driver Standings(Позиція пілотів після завершення сезону)

A screenshot of a black screen

Description automatically generated

Рисунок 3.40 – Вхідні дані

Приводимо дані у потрібний нам формат для сховища, видаляємо непотрібні поля та перейменуємо стовпці. Отже маємо такий результат:

A screenshot of a black screen

Description automatically generated

Рисунок 3.41 – Вхідні дані

в) Було створено функції завантаження даних у сховище даних. Тому вcі таблиці в сховищі тепер заповнені:

A computer code on a black background

Description automatically generated  
Рисунок 3.42 – Функція завантаження даних у сховище

1. За допомогою функції insert\_data() було завантажено circuits у створену таблицю виміру circuits (рисунок 3.43).

A screenshot of a computer

Description automatically generated

Рисунок 3.43 - Завантаження даних у таблицю circuits

1. A screenshot of a computer

   Description automatically generatedЗа допомогою функції insert\_data() було завантажено constructors у створену таблицю виміру constructors (рисунок 3.44)

Рисунок 3.44 - Завантаження даних у таблицю constructors

1. За допомогою функції insert\_data() було завантажено countries у створену таблицю виміру countries (рисунок 3.45).

A screenshot of a computer

Description automatically generated

Рисунок 3.45 - Завантаження даних у таблицю countries

1. За допомогою функції insert\_data() було завантажено drivers у створену таблицю виміру drivers (рисунок 3.46).

A screenshot of a computer

Description automatically generated

Рисунок 3.46 - Завантаження даних у таблицю drivers

1. За допомогою функції insert\_data() було завантажено grand\_prix у створену таблицю виміру grand\_prix (рисунок 3.47).

A screenshot of a computer

Description automatically generated

Рисунок 3.47 - Завантаження даних у таблицю grand\_prix

1. За допомогою функції insert\_data() було завантажено races у створену таблицю виміру races (рисунок 3.48).

A screenshot of a computer

Description automatically generated

Рисунок 3.48 - Завантаження даних у таблицю races

1. За допомогою функції insert\_data() було завантажено races\_constructor\_standings у створену таблицю виміру races\_constructor\_standings (рисунок 3.49).

A screenshot of a computer

Description automatically generated

Рисунок 3.49 - Завантаження даних у таблицю races\_constructor\_standings

1. За допомогою функції insert\_data() було завантажено races\_dod\_results у створену таблицю виміру races\_dod\_results (рисунок 3.50).

A screenshot of a computer

Description automatically generated

Рисунок 3.50 - Завантаження даних у таблицю races\_dod\_results

1. За допомогою функції insert\_data() було завантажено races\_driver\_standings у створену таблицю виміру races\_driver\_standings (рисунок 3.51).

A screenshot of a computer

Description automatically generated

Рисунок 3.51 - Завантаження даних у таблицю races\_driver\_standings

1. За допомогою функції insert\_data() було завантажено races\_fp1\_results у створену таблицю виміру races\_fp1\_results (рисунок 3.52).

A screenshot of a computer

Description automatically generated

Рисунок 3.52 - Завантаження даних у таблицю races\_fp1\_results

1. За допомогою функції insert\_data() було завантажено races\_fp2\_results у створену таблицю виміру races\_fp2\_results (рисунок 3.53).

A screenshot of a computer

Description automatically generated

Рисунок 3.53 - Завантаження даних у таблицю races\_fp2\_results

1. За допомогою функції insert\_data() було завантажено races\_fp3\_results у створену таблицю виміру races\_fp3\_results (рисунок 3.54).

A screenshot of a computer

Description automatically generated

Рисунок 3.54 - Завантаження даних у таблицю races\_fp3\_results

1. За допомогою функції insert\_data() було завантажено races\_pit\_stops у створену таблицю виміру races\_pit\_stops (рисунок 3.55).

A screenshot of a computer

Description automatically generated

Рисунок 3.55 - Завантаження даних у таблицю races\_pit\_stops

1. За допомогою функції insert\_data() було завантажено races\_qualifying\_results у створену таблицю виміру races\_qualifying\_results (рисунок 3.56).

A screenshot of a computer

Description automatically generated

Рисунок 3.56 - Завантаження даних у таблицю races\_qualifying\_results

1. Оскільки ми використвуємо дані з бібліотеки пайтон fastf1, то немає потреби імпортувати інформацію races\_race\_lap\_times в сховище, тому можливість продемонструвати є й у вигляді зчитування в датафрейм датасету (рисунок 3.57).

A screenshot of a black screen

Description automatically generated

Рисунок 3.57 - Вміст races\_race\_lap\_times

1. За допомогою функції insert\_data() було завантажено races\_race\_results у створену таблицю виміру races\_race\_results (рисунок 3.57).

A screenshot of a computer

Description automatically generated

Рисунок 3.57 - Завантаження даних у таблицю races\_race\_results

1. За допомогою функції insert\_data() було завантажено races\_sprint\_qualifying\_results у створену таблицю виміру races\_sprint\_qualifying\_results (рисунок 3.58).

A screenshot of a computer

Description automatically generated

Рисунок 3.58 - Завантаження даних у таблицю races\_sprint\_qualifying\_results

1. Оскільки ми використвуємо дані з бібліотеки пайтон fastf1, то немає потреби імпортувати інформацію races\_sprint\_race\_lap\_times в сховище, тому можливість продемонструвати є й у вигляді зчитування в датафрейм датасету (рисунок 3.59).

A screenshot of a computer

Description automatically generated

Рисунок 3.59 – Вміст races\_sprint\_race\_lap\_times

1. За допомогою функції insert\_data() було завантажено races\_sprint\_race\_results у створену таблицю виміру races\_sprint\_race\_results (рисунок 3.60).

A screenshot of a computer

Description automatically generated

Рисунок 3.60 - Завантаження даних у таблицю races\_sprint\_race\_results

1. За допомогою функції insert\_data() було завантажено seasons у створену таблицю виміру seasons (рисунок 3.61).

A screenshot of a table

Description automatically generated

Рисунок 3.61 - Завантаження даних у таблицю seasons

1. За допомогою функції insert\_data() було завантажено seasons\_constructor\_standings у створену таблицю виміру seasons\_constructor\_standings (рисунок 3.62).

A screenshot of a computer

Description automatically generated

Рисунок 3.62 - Завантаження даних у таблицю seasons\_constructor\_standings

1. За допомогою функції insert\_data() було завантажено seasons\_driver\_standings у створену таблицю виміру seasons\_driver\_standings (рисунок 3.63).

A screenshot of a table

Description automatically generated

Рисунок 3.63 - Завантаження даних у таблицю seasons\_driver\_standings

**4.ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ**

4.1 Обґрунтування вибору методів інтелектуального аналізу даних

Для обробки інформації та виявлення в ній моделей та тенденцій для прийняття рішень необхідно використовувати інтелектуальний аналіз даних. В даній курсовій роботі для передбачення аналізу популярності ігор та доходу компаній було обрано методи прогнозування даних. Спираючись на результати попередніх даних ігор за допомогою математичних методів, можна оцінити майбутні показники, що є актуальним для багатьох інтернет-магазинів відеоігор для виявлення закономірностей. В якості моделей для проведення прогнозування даних відеоігор було обрано Exponential Smoothing, ARIMA, SARIMAX, Holt.

Вибір був зроблений на основі суттєвих факторів, що полегшують проведення прогнозування. Дані моделі дають змогу аналітикам обробляти структурні розриви та зсуви даних, що змінюються з плином часу, оперативно змінювати параметри самої моделі, що включає в себе коваріацію шумів, матриць переходів та спостереження. Обрані моделі для прогнозування дають високі показники ефективності у роботі з даними з різних джерел в одній моделі. Подальший аналіз даних на кожній з моделей та порівняння їх ефективності буде описано в наступних підрозділах[5].

4.2 Математична модель прогнозування часових проміжків

Для дослідження часових рядів головними чинниками є опис та моделювання їх структури. Побудована модель може використовуватися для екстраполяції або прогнозування часового ряду, ….

Модель ARIMA– це досить поширений клас статистичних моделей для аналізу та …[1].

Модель ARIMA для нестаціонарного часового ряду має вигляд:

де – стаціонарний часовий ряд, – параметри моделі, – оператор різниці часового ряду порядку k.

Модель SARIMAX …

Exponential Smoothing – це модель прогнозування, яка враховує як лінійний тренд у вибірці, так і сезонні варіації у даних за допомогою використання трьох додаткових параметрів. Дана модель описується чотирма формулами:

* експоненціально згладженний ряд
* оцінка тренду
* оцінка сезонності
* прогнозування на p-періодів вперед

де α, β, γ — константи згладжувания для рівня, тренду и сезонності, відповідно; s – тривалість періоду сезонного коливання[8].

Holt - модель прогнозування, … наступним чином:

(t,k)=(t)+k⋅(t),

де (t) – оцінка поточного (t-го) значення ряду; (t) – оцінка поточної зміни.

Далі визначається величина відхилення прогнозованого значення від реального e. При k=1 маємо:

e(t+1)=X(t+1)−(t,1),

(t)= (t−1)+ (t−1)+⋅e(t),

де - коефіцієнт згладжування (адаптації), що змінюється в межах від 0 до 1.

Параметри моделі обчислюються послідовно … на більш ранніх етапах[11].

4.3 Практичне застосування моделей прогнозування часових проміжків

Опираючись на теоретичний матеріал попереднього підпункту, опишемо практичне використання моделей прогнозування часових проміжків. Для початку роботи, оберемо потрібну вибірку даних, що включає в себе кількість гравців та кількість продажів певної гри з плином часу. В якості розробника обраних ігор буде виступати відома компанія … (рисунок 4.1).



Рисунок 4.1 – Вибірка даних для прогнозування

Відобразимо графічно зміну кількості гравців та кількості продажів гри у вигляді часового ряду (рисунки 4.2-4.3).

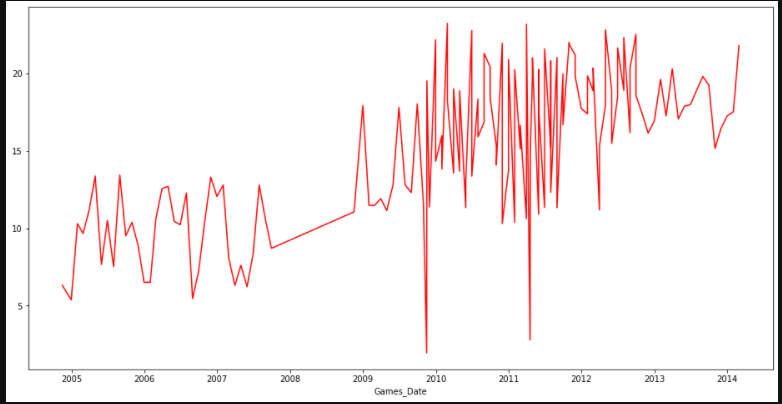


Рисунок 4.2 – Графічне відображення кількості продажів ігор у вигляді часового ряду

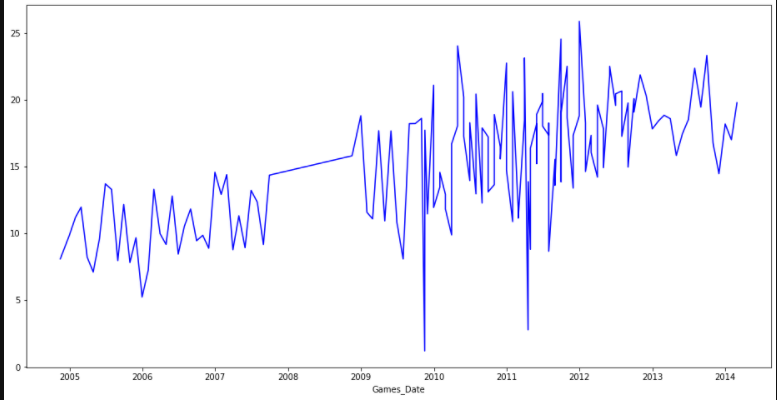


Рисунок 4.3 – Графічне відображення кількості гравців у вигляді часового ряду

Для перевірки на тренд, сезонність та шум застосуємо декомпозицію обраних даних (рисунки 4.4-4.5)

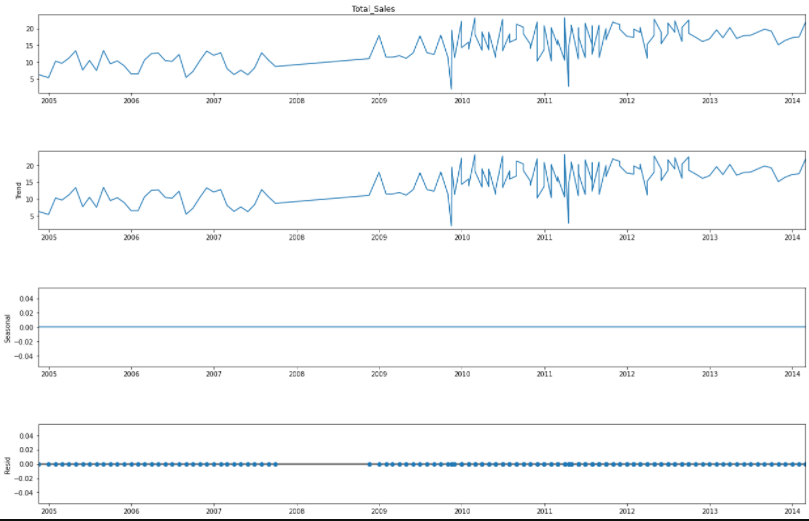


Рисунок 4.4 – Графічне зображення тренду, сезонності та залишкових атрибутів на прикладі даних кількості продажів ігор

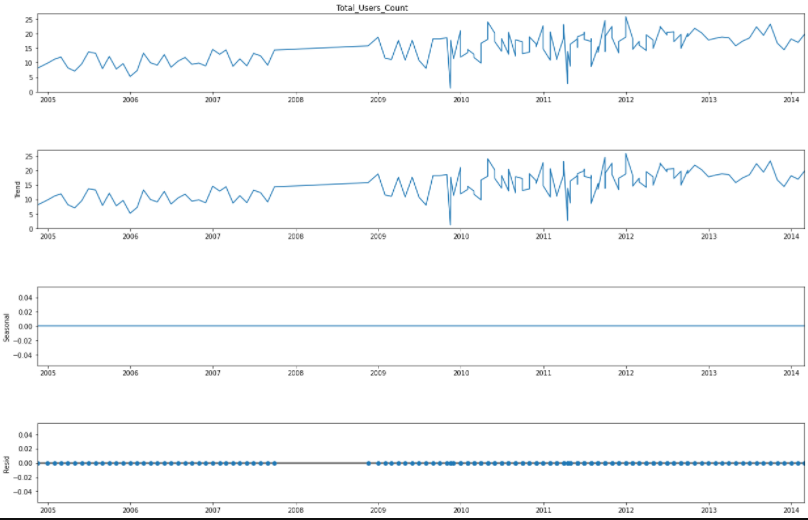


Рисунок 4.5 – Графічне зображення тренду, сезонності та залишкових атрибутів на прикладі даних кількості гравців

Дані графіки наочно демонструють те, що обрані дані є несезонними, а отже це дає змогу сказати, що дані є стаціонарними.

Для перевірки даних на сезонність було реалізовано розширений тест Дікі-Фуллера[10], який підтверджує результати отримані після декомпозиції даних. Отже, дані щодо кількості гравців та продажів ігор є стаціонарними (рисунки 4.6-4.7).

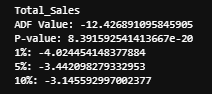


Рисунок 4.6 – Результат розширеного тесту Дікі-Фуллера для перевірки на сезонність кількості продажів ігор

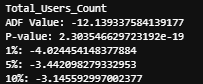


Рисунок 4.7 – Результат розширеного тесту Дікі-Фуллера для перевірки на сезонність кількості гравців

Отримані результати демонструють, що запропоновані у курсовій роботі моделі прогнозування підходитимуть під обрані дані для аналізу[9].

Для зменшення шуму …. В ролі вхідних даних в даній курсовій роботі виступають кількість гравців та кількість продажів ігор (рисунки 4.7 – 4.12).

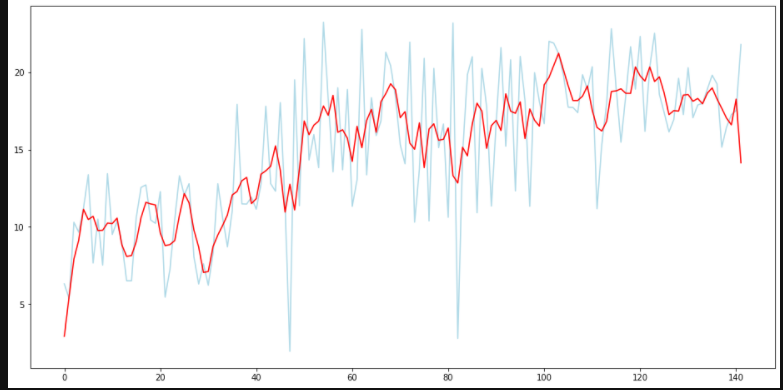


Рисунок 4.8 – Графічне відображення згладжування даних кількості продажів ігор на основі експоненціального згладжування

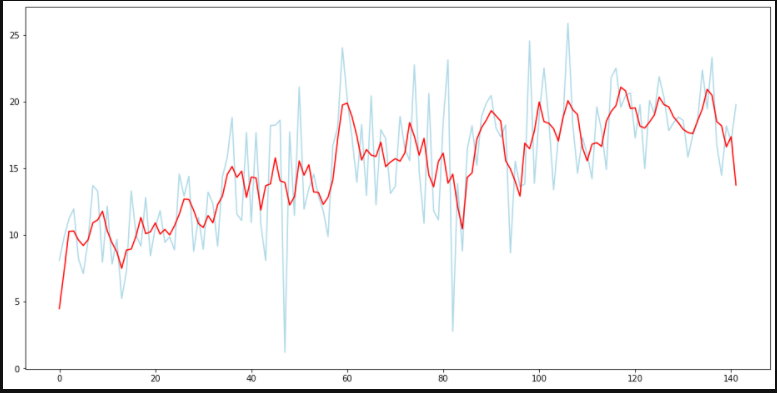


Рисунок 4.9 – Графічне відображення згладжування даних кількості гравців на основі експоненціального згладжування

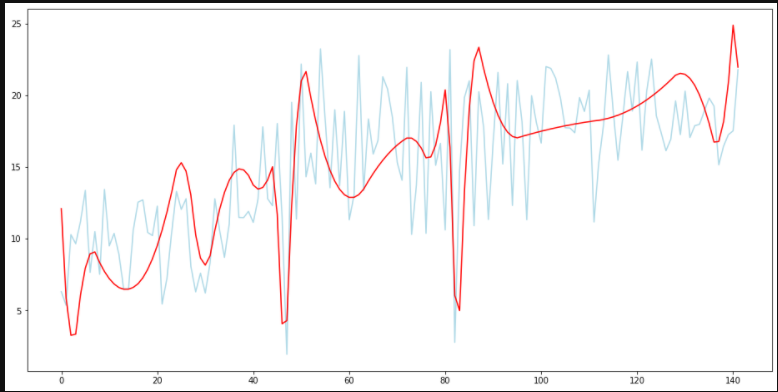


Рисунок 4.10 – Графічне відображення згладжування даних кількості продажів ігор на основі Вейвлет згладжування

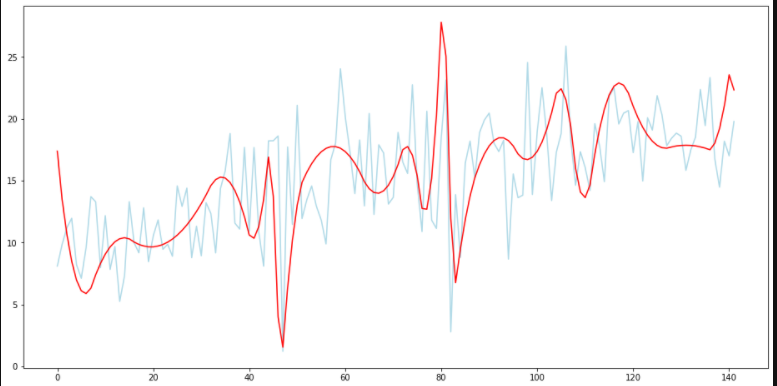


Рисунок 4.11 – Графічне відображення згладжування даних кількості гравців на основі Вейвлет згладжування

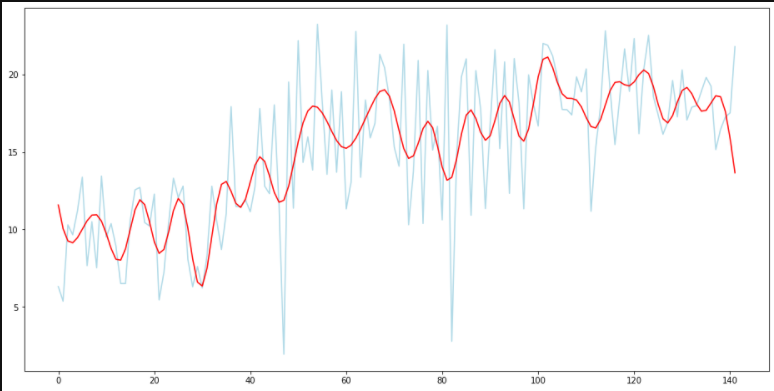


Рисунок 4.12 - Графічне відображення згладжування даних кількості продажів ігор на основі перетворення Фур'є

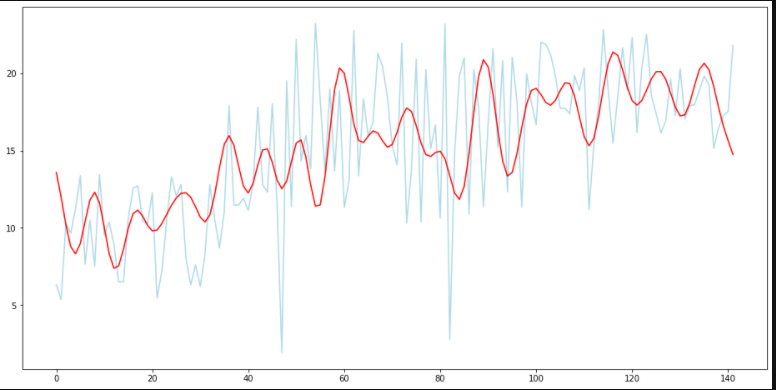


Рисунок 4.13 - Графічне відображення згладжування даних кількості гравців на основі перетворення Фур'є

З вище продемонстрованих графіків можна зробити висновок, що найкраще відтворює траєкторію кривої вхідних даних метод експоненціального згладжування, який і буде використовуватись для подальшого прогнозування даних відеоігор[8].

Перейдемо безпосередньо до графічного представлення реалізації прогнозування на обраних моделях такі як Exponential Smoothing, ARIMA, SARIMAX, Holt. На рисунках чорним кольором позначено вибірку тренувальних даних, а червоним та чорним – дані вибірки та прогнозовані дані відповідно.

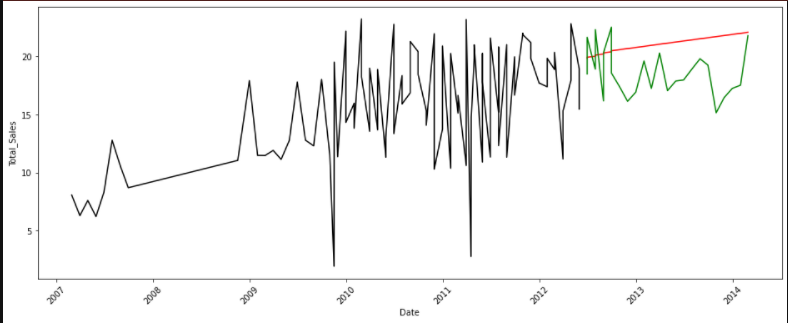


Рисунок 4.14 – Прогнозування вхідних даних кількості продажів ігор за моделлю Holt

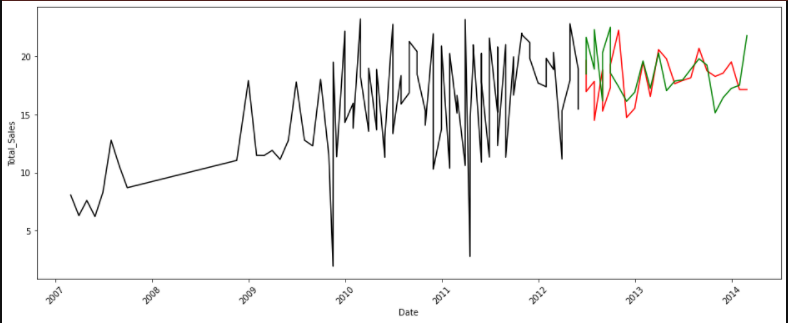


Рисунок 4.15 – Прогнозування вхідних даних кількості продажів ігор за моделлю Exponential Smoothing

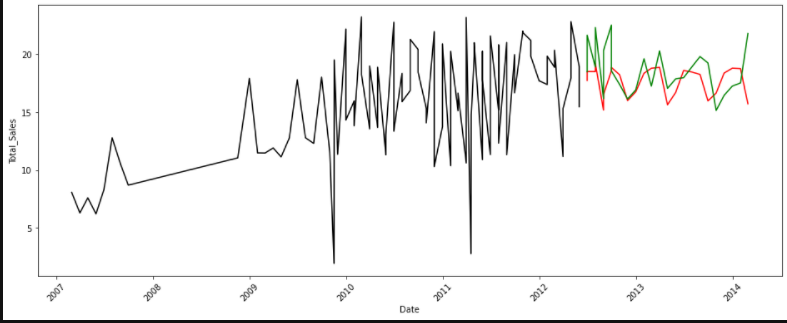


Рисунок 4.16 – Прогнозування вхідних даних кількості продажів ігор за моделлю SARIMAX

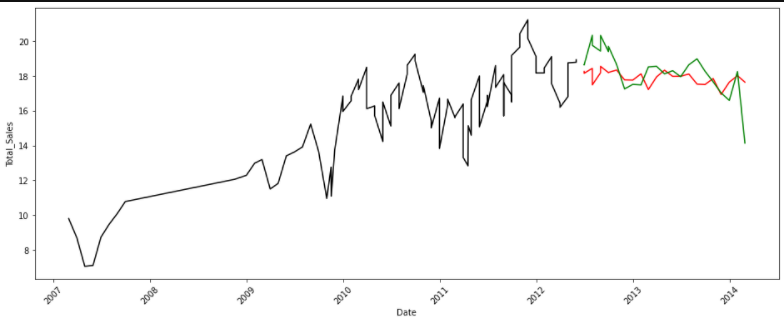


Рисунок 4.17 – Прогнозування згладжувальних даних кількості продажів ігор за моделлю SARIMAX

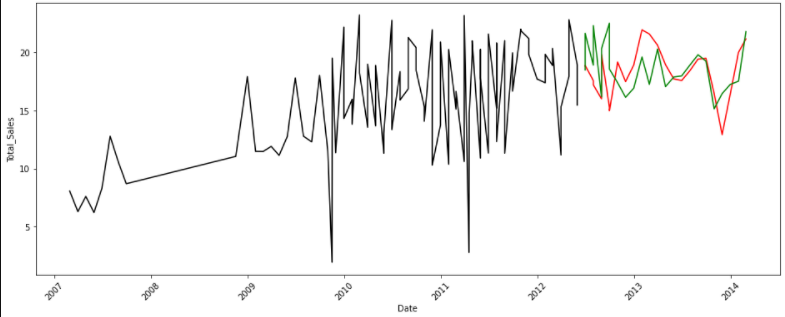


Рисунок 4.18 – Прогнозування вхідних даних кількості продажів ігор за моделлю ARIMA

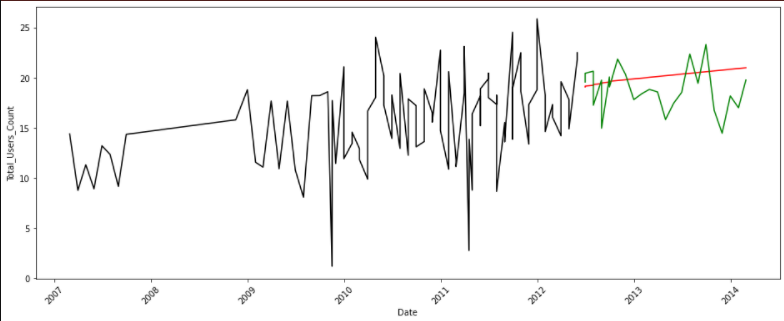


Рисунок 4.19 – Прогнозування вхідних даних кількості гравців за моделлю Holt

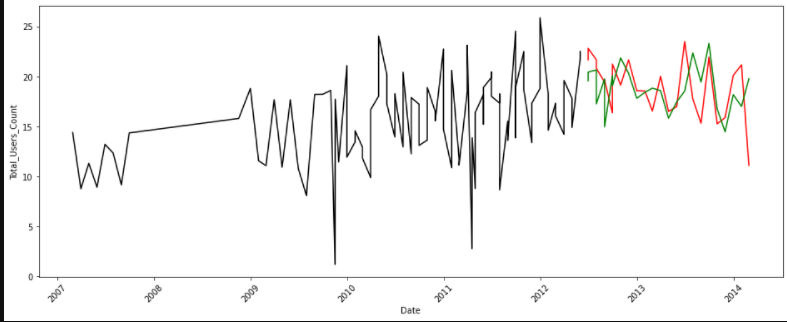


Рисунок 4.20 – Прогнозування вхідних даних кількості гравців за моделлю Exponential Smoothing

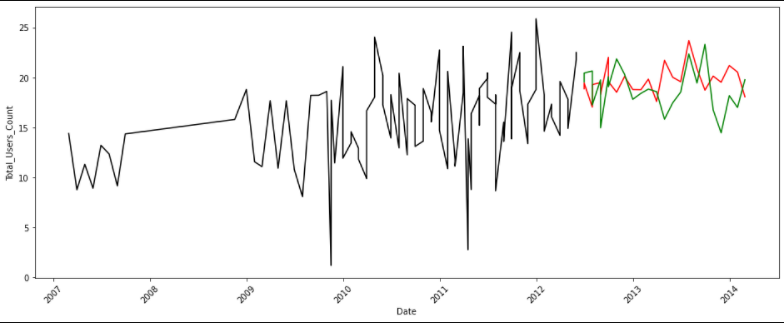


Рисунок 4.21 – Прогнозування вхідних даних кількості гравців за моделлю SARIMAX

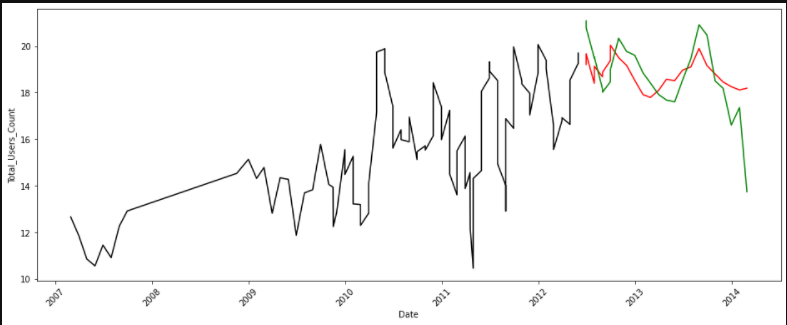


Рисунок 4.22 – Прогнозування згладжувальних даних кількості гравців за моделлю SARIMAX

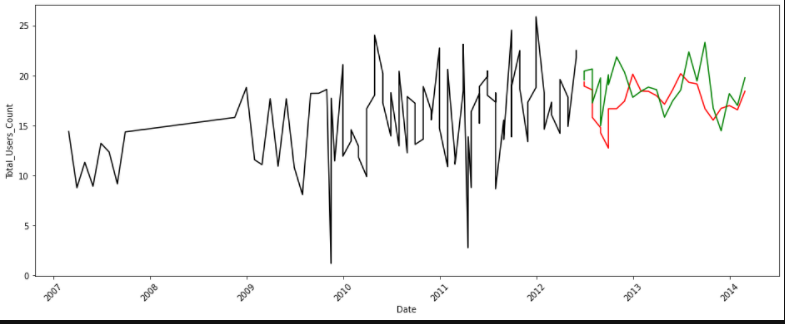


Рисунок 4.23 – Прогнозування вхідних даних кількості гравців за моделлю ARIMA

Для моделі SARIMAX виведемо діагностику для прогнозування кількості гравців та продажу ігор, що демонструє наступне (рисунки 4.24-4.25):

* Standardized residual for „T” … яка не пояснюється в моделі[7].

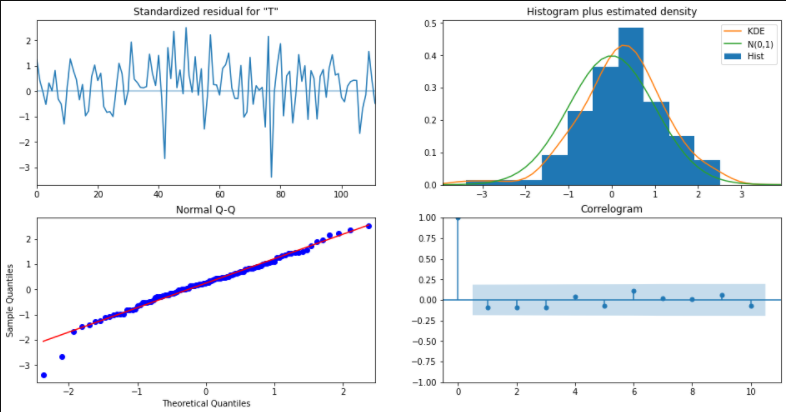


Рисунок 4.24 – Вивід діагностики для моделі SARIMAX на основі даних продажів ігор

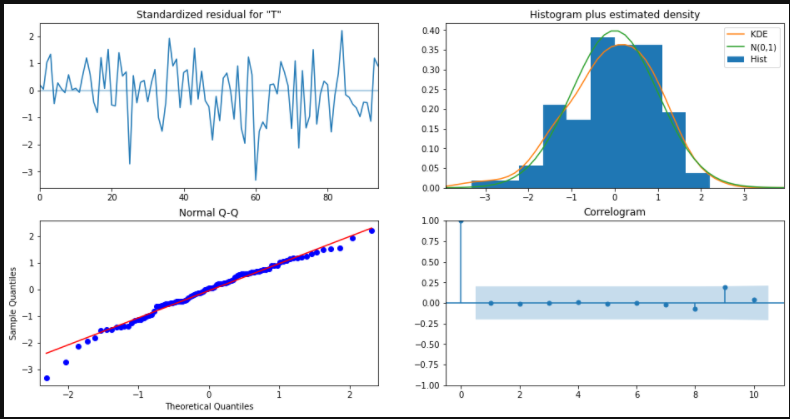


Рисунок 4.25 – Вивід діагностики для моделі SARIMAX на основі даних кількості гравців

Виведемо графіки автокореляції та часткової автокореляції для обраних у курсовій роботі даних для моделі SARIMAX(рисунки 4.26 - 4.27).

У цьому випадку автокореляція демонструє спостереження часового проміжку зі значенням того з цього ж проміжку в попередній інтервал часу. Часткова автокореляція дає змогу нам отримати взаємозв’язок між спостереженням в часовому проміжку зі спостереженнями на попередніх часових проміжках без урахування взаємозв’язків між проміжними спостереженнями[6].

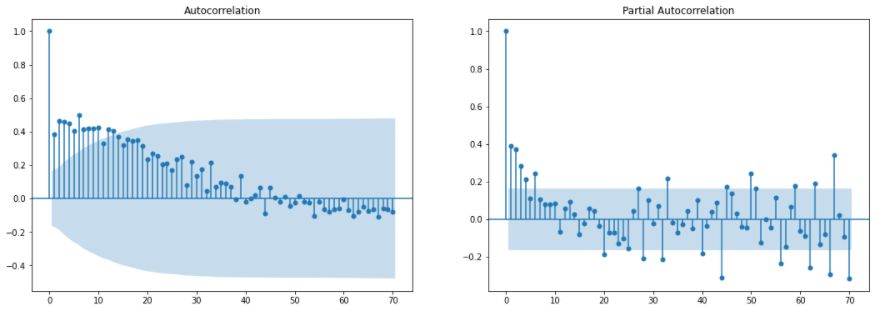


Рисунок 4.26 – Графіки автокореляції та часткової автокореляції на вибірці даних продажів ігор для моделі SARIMAX

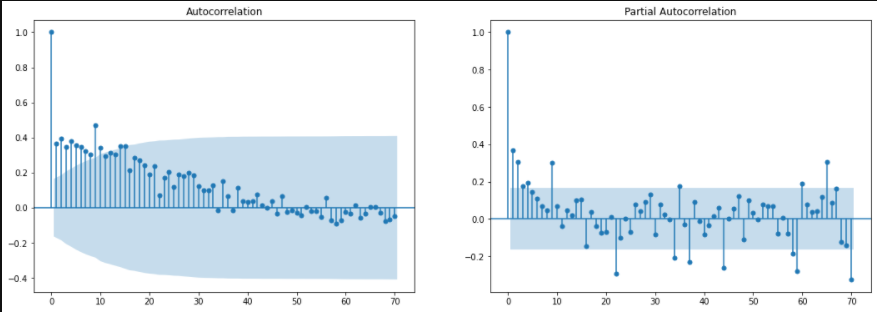


Рисунок 4.27 – Графіки автокореляції та часткової автокореляції на вибірці даних кількості гравців для моделі SARIMAX

…..

Враховуючи вищеописану реалізацію можна зробити висновки, проаналізувавши дані та вибрати найоптимальніший метод.

4.4 Порівняння ефективності методів інтелектуального аналізу

Аналізуючи прибутки з продажів ігор чи кількості гравців у певний момент часу важливо отримувати найбільш точний результат, щоб передбачати майбутні результати. Для аналітичного визначення точності результатів, отриманих на основі моделей для прогнозування, та вибору найоптимальнішого методу знайдемо середньоквадратичні помилки MSE для кожної з моделей вибірки даних кількості продажів ігор та кількості гравців.

Для кількості продажів ігор отримано наступні результати:

* Модель Exponential Smoothing – MSE = 8.952499933006333
* Модель ARIMA – MSE = 6.406898744038264
* Модель SARIMAX на вихідних даних – MSE = 4.864811488884299
* Модель SARIMAX на згладжувальних даних –
* …

З отриманих результатів середньоквадратичної помилки, а також враховуючи рисунки 4.14 - 4.23, можна зробити висновок, що найоптимальнішим методом для прогнозування що кількості продажів ігор, що кількості гравців є метод, реалізований на основі моделі …. Варто зазначити, що після проведення згладжування даних,….

**ВИСНОВКИ**

В результаті виконання курсової роботи було розроблено сховище даних відеоігор типу „сніжинка”, що містить дві таблиці фактів та вісім таблиць вимірів. Реалізовано ETL процеси для завантаження даних, …. Для реалізації поставленої задачі було використано PostgreSQL версії 12.1, мова програмування ….

На основі детального опису та проведеного аналізу предметної області інтелектуального аналізу даних для прогнозування кількості продажів ігор та факторів впливу на популярність відеоігор було отримано результати аналізу вибірки даних на сезонність, стаціонарність, шум за допомогою доповненого тесту …. Підтвердженням даних висновків є результати середньоквадратичних помилок, а для найкращого результату …. Результати досліджень показують, що …

Отже, поставлені задачі були виконані, а також планується розширення функціоналу для взаємодії з сховищем даних відеоігор з використанням більшої кількості факторів, які ….

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Бокс Дж., Дженкинс Г.М. Анализ временных рядов. Прогноз и управление. –– Мир, 1974.
2. Scott Lundberg and Su-In Lee, “A Unified Approach to Interpreting Model Predictions”, 2017
3. Коровкин С. Д., Левенец И. А., Ратманова И. Д., Старых В. А., Щавелёв Л. В. Решение проблемы комплексного оперативного анализа информации хранилищ данных. // СУБД. – 1997. – № 5-6. – 47-51с.
4. Lloyd Shapley, “A value of n-person games. Contributions to the Theory of Games”, 1953
5. Tobias Lang and Matthias Rettenmeier, “Understanding Consumer Behavior with Recurrent Neural Networks”, 2017
6. Wikipedia. ARIMA // Википедия, свободная энциклопедия. –– 2014. –– URL: http://goo.gl/bNmcX8.
7. Wikipedia. Модель авторегрессии скользящего среднего // Википедия, свободная энциклопедия. –– 2014. –– URL: http://goo.gl/TSHFst.
8. Альперович М. Введение в OLAP и многомерные базы данных. – М.: Вильямс. 2005. – 267с.
9. Jerome Friedman, “Greedy function approximation: A gradient boosting machine”, 2001
10. NeuroProject. Методы прогнозирования. –– 2014. –– URL: http://goo.gl/9Jn0UV.
11. Tony Duan, Anand Avati, Daisy Ding, Khanh Thai, Sanjay Basu, Andrew Ng, and Alejandro Schuler, “NGBoost: Natural Gradient Boosting for Probabilistic Prediction”, 2020

**Додаток А Тексти програмного коду**

*студента групи ХХХ ІІ курсу*

*ХХХ*

(Обсяг програми (документа), арк., Кб)

*28 арк, 64 Кб*

(Вид носія даних)

*SSD*

(Найменування програми (документа))

*Тексти програмного коду* прогнозування доходів та факторів на популярність індустрії відеоігор

import pandas as pd

import numpy as np

import datetime

…