

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ  
ТАРАСА ШЕВЧЕНКА**  
**Факультет інформаційних технологій**

Кафедра технологій управління

Спеціальність 122 – Комп’ютерні науки,  
освітня програма «Інформаційна аналітика та впливи»

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему:

“Розробка моделей та інформаційної технології прогнозування показників  
теледивлення за допомогою методів науки про дані”

**Студента 2-го курсу групи ІАВ-21**

Олександра ІЛЬЧУКА

**Науковий керівник**

к.е.н., доцент кафедри  
технологій управління

Ігор МІРОШНИЧЕНКО

\_\_\_\_\_

(підпис студента)

\_\_\_\_\_

(дата)

\_\_\_\_\_

(підпис)

|   |                      |        |
|---|----------------------|--------|
| <b>Попередній захист:</b>                         |                      |        |
| (Висновок: «До захисту в Екзаменаційній комісії») |                      |        |
| Завідувач кафедри<br>технологій управління _____  |                      |        |
| (підпис)  | (прізвище, ініціали) | (дата) |

**Київ – 2025**

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА  
Факультет інформаційних технологій**

Кафедра технологій управління

Освітньо-кваліфікаційний рівень Магістр

Спеціальність 122 – Комп'ютерні науки

Освітня програма Інформаційна аналітика та впливи

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

професор Віктор МОРОЗОВ

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент Олександр ІЛЬЧУК

Група ІАВ–21

**1. Тема кваліфікаційної роботи**

Розробка моделей та інформаційної технології прогнозування показників теледивлення за допомогою методів науки про дані

Затверджена наказом по від «\_»\_\_\_\_\_20\_\_р. № \_\_.

**Строк подання студентом готової роботи – “19” травня\_2025р.**

**Цільова установка та вихідні дані до роботи**

Дослідження методів прогнозування показників теледивлення та прогнозування показника Affinity, створення інтелектуальної системи автоматичного прогнозування показників.

## 2. Зміст роботи

Робота містить вступ, огляд наукової літератури на тему, огляд існуючих досліджень та методів їх проведення, відповідно до визначеної мети - постановку завдання. Робота містить опис існуючих математичних та власних математичних моделей, за допомогою яких відбувається досягнення мети дослідження. У роботі наведено опис побудова власної моделі та інтелектуальної системи на її основі. Представлено обґрунтування вибору певних моделей та методів дослідження і моделювання. Робота містить висновки, список використаних наукових джерел та додатки.

## 3. Перелік графічного матеріалу (слайдів)

Загалом робота містить 49 рисунків, 25 слайдів.

Перелік слайдів: мета (1 слайд), задачі дослідження (1 слайд), об'єкт і предмет (1 слайд), наукова новизна та практична цінність (1 слайд), основи ТБ та планування (6 слайдів), побудова та оцінювання моделей (9 слайдів), впровадження моделі (2 слайди), відгук з підприємства (1 слайд), висновки (1 слайд).

## 4. Календарний план виконання роботи:

| №  | Назва частин роботи  | %  | Виконання роботи |          |
|----|--|----|------------------|----------|
|    |  |    | За планом        | Фактично |
| 1. | Вибір теми кваліфікаційної магістерської роботи, дослідження актуальності обраної теми, наявності наукових матеріалів з теми | 3  | 01.10.24         | 01.10.24 |
| 2. | Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників                                  | 2  | 27.12.24         | 27.12.24 |
| 3. | Формування переліку Нормативних матеріалів, літератури з проблематики дипломної роботи                                       | 10 | 10.01.25         | 12.01.25 |

|     |  |    |                        |          |
|-----|--|----|------------------------|----------|
| 4.  | Складання розгорнутого плану виконання та представлення кваліфікаційної роботи   | 5  | 18.01.25               | 25.01.25 |
| 5.  | Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.   | 5  | 29.01.25 –<br>30.01.25 | 30.01.25 |
| 6.  | Підготовка розділу 1 «Постановка задачі дослідження та аналіз її вирішення». Визначення наповнення розділу, висновків.                         | 10 | 12.02.25               | 13.02.25 |
| 7.  | Підготовка розділу 2 «Математичні моделі та методика прогнозування часових рядів». Визначення наповнення розділу, висновків.                   | 14 | 10.03.25               | 10.03.25 |
| 8.  | Підготовка розділу 3 «Розробка моделей прогнозування показника Affinity». Визначення наповнення розділу, висновків.                            | 14 | 02.04.25               | 02.04.25 |
| 9.  | Підготовка розділу 4 «Технологія автоматичного прогнозування показника Affinity». Визначення наповнення розділу, висновків.                    | 13 | 09.04.25               | 09.04.25 |
| 10. | Оформлення кваліфікаційної роботи. Підготовка аналізу результатів роботи, висновків. Перевірка відповідності початковій меті та задачам роботи | 15 | 12.04.25               | 12.04.25 |
| 11. | Передача кваліфікаційної роботи науковому керівникові  | 2  | 20.04.25               | 23.04.25 |

|     |   |   |          |          |
|-----|---|---|----------|----------|
| 12. | Передача кваліфікаційної роботи рецензенту для рецензування | 2 | 01.05.25 | 01.05.25 |
| 13. | Попередній захист кваліфікаційної роботи                    | 5 | 03.05.25 | 03.05.25 |

Дата видачі завдання «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

Керівник роботи: к.е.н., доцентр кафедри технологій управління Ігор МІРОШНИЧЕНКО

\_\_\_\_\_  
(підпис)

Завдання прийняв до виконання студент групи ІАВ-21 Олександр ІЛЬЧУК

\_\_\_\_\_  
(підпис)

## ЗМІСТ

|  |    |
|--|----|
| АНОТАЦІЯ.....  | 10 |
| ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ .....   | 12 |
| ВСТУП.....   | 13 |
| Розділ 1. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ЇЇ<br>ВИРІШЕННЯ.....                                     | 18 |
| 1.1 Аналіз об'єкту дослідження.....  | 18 |
| 1.1.1 Роль телебачення у медіапросторі .....   | 18 |
| 1.1.2 Процес вимірювання теледивлення .....  | 19 |
| 1.1.3 Основні показники теледивлення .....   | 22 |
| 1.2 Постановка задачі прогнозування показників теледивлення.....   | 25 |
| 1.2.1 Особливості часових рядів теледивлення .....   | 25 |
| 1.2.2 Практичні задачі та кейси прогнозування: маркетинг, програмне<br>планування, закупівлі реклами ..... | 25 |
| 1.2.3 Виклики у прогнозуванні: сезонність, тренди, особливі події,<br>різкі зміни, розмір даних.....       | 26 |
| 1.3 Огляд сучасних методів прогнозування в телеметрії .....  | 27 |
| 1.3.1 Класичні статистичні методи: ARIMA .....   | 27 |
| 1.3.2 Експоненційні моделі згладжування: ETS .....   | 28 |
| 1.3.3 Байєсівсько-евристична модель Prophet.....   | 28 |
| 1.3.4 Глибинне навчання: LSTM .....  | 29 |
| 1.3.5 Порівняння ефективності підходів для даних теледивлення ....   | 30 |
| 1.4 Постановка задачі .....  | 30 |

|   |    |
|---|----|
| 1.5 Висновки до розділу 1 .....   | 33 |
| Розділ 2. МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИКА ПРОГНОЗУВАННЯ<br>ЧАСОВИХ РЯДІВ ..... | 35 |
| 2.1 ARIMA-модель для прогнозування часових рядів .....                        | 35 |
| 2.2 Метод експоненційного згладжування (ETS) .....                            | 36 |
| 2.3 Модель LSTM.....  | 38 |
| 2.4 Prophet.....  | 40 |
| 2.5 Показники якості прогнозу .....   | 42 |
| 2.5.1 MAE.....  | 42 |
| 2.5.2 MAPE.....   | 43 |
| 2.5.3 RMSE.....   | 43 |
| 2.5.4 Коефіцієнт детермінації ( $R^2$ ) .....                                 | 43 |
| 2.5.5 MASE.....   | 43 |
| 2.6 Процес охайного (tidy) прогнозування (tidy forecasting workflow)..        | 44 |
| 2.6.1 Загальні принципи <i>tidy</i> -підходу .....                            | 45 |
| 2.6.2 Етап 1. Підготовка та структуризація даних .....                        | 46 |
| 2.6.3 Етап 2. Дослідження та візуальний аналіз .....                          | 46 |
| 2.6.4 Етап 3. Попередня обробка та інженерія ознак .....                      | 47 |
| 2.6.5 Етап 4. Визначення та навчання моделей .....                            | 47 |
| 2.6.6 Етап 5. Оцінювання точності та порівняння моделей .....                 | 47 |
| 2.6.7 Етап 6. Перенавчання, прогноз та інтеграція у бізнес-процеси            | 48 |
| 2.7 Висновки до розділу.....  | 48 |
| РОЗДІЛ 3. РОЗРОБКА МОДЕЛЕЙ ПРОГНОЗУВАННЯ ПОКАЗНИКА<br>AFFINITY .....          | 51 |

|   |            |
|---|------------|
| 3.1 Отримання даних теледивлення .....                              | 51         |
| 3.1.1 Створення аудиторії .....                                     | 51         |
| 3.1.2 Створення звіту (Report Advertising) .....                    | 52         |
| 3.2 Обробка даних .....   | 55         |
| 3.2.1 Завантаження вхідних даних.....                               | 56         |
| 3.2.2 Обробка даних .....   | 57         |
| 3.2.3 Об'єднання даних цільової та закупівельної аудиторій .....    | 57         |
| 3.3 Візуалізація часових рядів .....                                | 58         |
| 3.4 Модель ARIMA.....   | 62         |
| 3.5 Модель ETS.....   | 71         |
| 3.6 Модель LSTM.....  | 74         |
| 3.7 Модель Prophet .....  | 79         |
| 3.8 Оцінка моделей.....   | 83         |
| 3.9 Висновки до розділу.....  | 87         |
| <b>РОЗДІЛ 4. ТЕХНОЛОГІЯ АВТОМАТИЧНОГО ПРОГНОЗУВАННЯ</b>             |            |
| <b>ПОКАЗНИКА AFFINITY .....</b>                                     | <b>89</b>  |
| 4.1 Автоматизація даних теледивлення в програмі Yumianalytics ..... | 89         |
| 4.2 Розробка алгоритму автоматичного оновлення даних Affinity.....  | 93         |
| 4.3 Розробка візуалізації для PowerBI .....                         | 95         |
| 4.4 Перспективи майбутніх досліджень .....                          | 100        |
| 4.5 Висновки до розділу.....  | 102        |
| <b>ВИСНОВОК .....</b>   | <b>104</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>                             | <b>106</b> |
| <b>ДОДАТКИ .....</b>  | <b>108</b> |

Додаток А ..... 108

**АНОТАЦІЯ**  
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ІМЕНІ ТАРАСА ШЕВЧЕНКА**  
**Факультет інформаційних**  
**технологій**

Кафедра технологій управління  
Спеціальність 122 – Комп'ютерні  
науки,

освітня програма “Інформаційна аналітика та впливи”

Дипломна робота магістра ІЛЬЧУКА Олександра Михайловича.

Тема роботи – «Розробка моделей та інформаційної технології прогнозування показників теледивлення за допомогою методів науки про дані».

Мета дипломної роботи магістра – підвищення точності планів рекламних кампаній за допомогою впровадження інтелектуальної технології прогнозування показників теледивлення.

Об'єкт дослідження – процес визначення показника теледивлення Affinity.

Предметом дослідження є методи, моделі та програмні засоби прогнозування показників теледивлення з використанням інструментів науки про дані.

Наукова новизна роботи – вперше запропоновано та реалізовано комплексний гібридний підхід прогнозування теледивлення, який поєднує статистичні методи (ARIMA), глибоке навчання (LSTM) та інструмент Prophet. Розроблено інформаційну платформу, адаптовану до специфіки українського ринку та інтегровану в робочі процеси рекламного агентства «АІТІ», що дозволило суттєво покращити точність прогнозів та ефективність аналітичної роботи.

У роботі досліджуються існуючі підходи до прогнозування даних часових рядів показника Affinity. Розробляється інформаційна технологія автоматизованого отримання даних та створення прогнозу. Це допоможе планерам рекламних кампаній не тільки робити більш точні прогнози зміни показників, а й оптимальніше використовувати рекламний бюджет.

Практичне значення одержаних результатів:

Дана система може бути використана в медійних агентствах та на телевізійних каналах, що зробить процес прогнозування різних показників більш доступним та точним.

У результаті дослідження були отримані такі результати:

- Аналіз та опис процесу отримання даних телевізійного вимірювання, визначення основних показників;
- Опис існуючих математичних моделей, що будуть використані в системі, а також розробка власних;
- Опис процесу прогнозування та результатів його роботи на тестових цільових аудиторіях;
- Оцінка розроблених моделей за допомогою метрик.

Дипломна робота складається зі вступу, основної частини, яка включає чотири розділи, висновків та списку використаних джерел. Всього налічує 112 сторінок та перелік посилань з 19 джерел.

Ключові слова: теледивлення, GRP, Affinity, ARIMA, LSTM, ETS, Prophet, Yumianalytics.

## ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

ADF — Augmented Dickey–Fuller test  
AFF % — Affinity Index  
ACF — Autocorrelation Function  
ARIMA — Autoregressive Integrated Moving Average  
BA — Buying (Base) Audience  
ETS — Error–Trend–Seasonality State-Space Model  
EqGRP — Equivalent Gross Rating Point  
GRP — Gross Rating Point  
KPSS — Kwiatkowski–Phillips–Schmidt–Shin test  
LSTM — Long Short-Term Memory  
MAE — Mean Absolute Error  
MAPE — Mean Absolute Percentage Error  
MSE — Mean Squared Error  
RMSE — Root Mean Squared Error  
PACF — Partial Autocorrelation Function  
Prophet — Facebook Prophet  
 $R^2$  — Coefficient of Determination  
RAT % — Rating (відсоток перегляду)  
SARIMA — Seasonal ARIMA  
STL — Seasonal–Trend decomposition using Loess  
TA — Target Audience  
TRP — Target Rating Point

## ВСТУП

Телебачення залишається важливим каналом масової комунікації, а прогнозування його показників є ключовим для ефективного планування медіастратегій та рекламних кампаній. Медіакомпанії та рекламодавці щодня спираються на телевізійні рейтинги для оцінки успіху програм та перевірки досягнення цілей медіапланів. Хоча рейтинги вимірюють насамперед минулі та поточні показники перегляду, їх активно використовують і для прогнозування майбутніх значень: саме на основі таких прогнозів встановлюються ціни на рекламні блоки, формується портфель програм і готуються пропозиції на майбутній телевізійний сезон.

Водночас сучасний медіапростір перебуває в стані стрімких змін: глядачі все більше дивляться контент на смартфонах і планшетах, використовують популярні стрімінгові сервіси (Netflix, Prime Video тощо) та функції відкладеного перегляду. Люди формують власні мультиплатформні звички, тому їх поведінка стає значно складнішою й менш прогнозованою. Ці тенденції ускладнюють традиційні підходи до прогнозування телевізійних показників і породжують потребу у застосуванні нових методів аналізу даних та штучного інтелекту.

Розвиток технологій науки про дані (Data Science) створює нові можливості для побудови точних прогнозних моделей. Завдяки сучасним статистичним та комп'ютерним методам можна оперативнo обробляти великі обсяги історичних даних про телеперегляд та враховувати додаткові чинники (час доби, день тижня, рекламні кампанії тощо). Такі підходи забезпечують вищу точність прогнозів порівняно з простими емпіричними методами, що особливо цінні для рекламно-аналітичних задач. Отже, розробка вдосконалених моделей прогнозування показників теледивлення є актуальною й затребуваною задачею сучасної індустрії реклами та медіа.

**Метою роботи є підвищення точності планів рекламних кампаній за допомогою впровадження інтелектуальної технології прогнозування показників теледивлення.**

Для досягнення цієї мети було поставлено такі задачі:

1. Провести огляд і аналіз існуючих підходів до прогнозування телевізійних аудиторних показників, зокрема традиційних статистичних моделей та сучасних алгоритмів машинного навчання.
2. Зібрати та попередньо обробити дані про телеперегляди (часові ряди показників перегляду), що надаються ТОВ «АІТІ». Встановити характер трендів, сезонності та автокореляційних властивостей даних.
3. Розробити та реалізувати алгоритми прогнозування на основі класичних методів (наприклад, ARIMA) та алгоритмів глибокого навчання (LSTM), а також пакету Prophet, що оптимально підходять для часових рядів зі складними сезонними компонентами.
4. Здійснити налаштування та перевірити правильність побудованих моделей, порівняти їхню точність (MAE, RMSE) та вибрати оптимальний підхід.
5. Створити прототип інформаційної системи (ІТ-технології) для автоматизації процесу прогнозування, інтегрувати моделі у прикладний програмний комплекс агенції «АІТІ».
6. Провести тестове впровадження системи в діяльність ТОВ «Рекламне агентство “АІТІ”» та оцінити її вплив на ефективність медіапланування.

**Об’єктом дослідження є процес визначення показника теледивлення Affinity – відношення цільової аудиторії до закупівельної.** До об’єкта належить система взаємозв’язків між часом перегляду, властивостями контенту, та поведінкою аудиторії.

**Предметом дослідження є методи, моделі та програмні засоби прогнозування показників теледивлення з використанням інструментів науки про дані.** Зокрема це алгоритми обробки часових рядів і побудови прогнозних

моделей (статистичних та машинного навчання) і програмна реалізація інформаційної технології, що їх об'єднує.

Моделі та методи дослідження, використані в роботі:

- **Статистичні методи аналізу** – дослідження стаціонарності часових рядів, побудова автокореляційних функцій, сезонне розкладення. Для цих цілей використовували пакети `pandas`, `statsmodels` та інші інструменти Python.
- **Класичне моделювання ARIMA** – реалізовано моделі ARIMA (Autoregressive Integrated Moving Average), які інтегрують авторегресійні та ковзні середні компоненти для прогнозу майбутніх значень часового ряду. ARIMA є одним із найпоширеніших підходів до прогнозування часових рядів. За потреби застосовано сезонні модифікації SARIMA для врахування циклів (тижневих, місячних).
- У машинному навчанні використано рекурентні нейронні мережі **LSTM (Long Short-Term Memory)** для моделювання довгострокових залежностей у часових рядах. LSTM-мережі спеціально спроектовані для запам'ятовування інформації протягом тривалих інтервалів часу, що робить їх придатними для прогнозування, де контекст попередніх даних істотно впливає на майбутнє. Навчання LSTM виконано із застосуванням TensorFlow/Keras на підготованому наборі послідовностей перегляду.
- **Прогнозування з Prophet** – використано бібліотеку Prophet (Facebook/Meta) як швидкий і зручний інструмент для автоматичного прогнозування з адитивною моделлю трендів і сезонності. Prophet дозволяє врахувати нелінійні тренди та різноманітні циклічні компоненти (річні, тижневі, щоденні), а також коригування на святкові дні, що актуально для телепрограм, що мають чіткі цикли показу.

Результати прогнозів перевірялися за допомогою відкладених вибірок та метрик якості (MAE, RMSE). Моделі налаштовувалися з урахуванням обмеженості даних та специфіки телевізійних показників. Таким чином, застосовано поєднання класичних статистичних методів і сучасних алгоритмів

машинного навчання для побудови моделі прогнозування показників теледивлення. Усі моделі реалізовано у середовищі Python із використанням відкритих бібліотек для обробки даних та машинного навчання.

У дипломній роботі вперше запропоновано та реалізовано комплексний гібридний підхід до прогнозування показників теледивлення, який поєднує класичні статистичні методи, зокрема ARIMA, з алгоритмами глибокого навчання (LSTM), а також сучасним прогнозним інструментом Prophet. Завдяки такому комбінуванню моделей досягнуто значно вищої точності прогнозу порівняно із застосуванням кожного з методів окремо.

Важливим аспектом новизни є адаптація прогнозних моделей до специфічних умов локального ринку, зокрема, українського телевізійного середовища. Моделі були спеціально налаштовані з урахуванням особливостей сезонних патернів та структурних змін, що виникли після переходу на нові технології вимірювання аудиторії, на основі даних ТОВ «Рекламне агентство «АІТІ»».

У рамках роботи вперше буде створено прикладну інформаційну платформу для автоматизованого прогнозування телепоказників, інтегровану в робочі процеси агентства «АІТІ». Розроблена система дає змогу оперативно оновлювати вихідні дані, автоматично сформувати точні прогнози та представляти результати у зрозумілій візуальній формі, суттєво спрощуючи аналітичну роботу планерів рекламних кампаній.

Результати проведених експериментів засвідчили покращення точності прогнозів порівняно з раніше використовуваними в агентстві методами. Крім того, вперше у вітчизняній практиці для прогнозування показників теледивлення було застосовано сучасні інструменти науки про дані — бібліотеку Prophet та технологію рекурентних нейронних мереж LSTM. Це розширює можливості аналітиків у прогнозуванні та відкриває перспективи для подальших досліджень та вдосконалень методик.

**Практичне значення роботи** полягає в тому, що розроблена система прогнозування вже впроваджена в діяльність ТОВ «Рекламне агентство “АІТІ”». Застосування моделей прогнозування дозволяє агентству планувати розміщення реклами з урахуванням очікуваної аудиторії, підвищувати ефективність медіапланів та оптимізувати витрати клієнтів. Зокрема, автоматизована інформаційна технологія прогнозування допомагає рекламним планерам приймати рішення про розподіл рекламних ресурсів наперед, знижуючи ризик невідповідності обраних програм і цільових аудиторій. Впровадження результатів роботи дозволило підвищити точність оцінок, необхідних для закупки рекламних рейтингів, що у свою чергу сприяє зростанню ефективності рекламних кампаній користувачів агентства.

Основні результати дипломного дослідження були апробовані шляхом доповіді на міжнародній науковій конференції **“Urban Transformation in the EU”** (17 квітня 2025 року, м. Київ). Конференція включала секцію, присвячену застосуванню аналітичних технологій у медіа- та телекомунікаціях, де автор представив ключові результати прогнозування телевізійної аудиторії та опис інформаційної технології.

За результатами виконаної роботи підготовлено матеріали для зазначеної конференції **“Urban Transformation in the EU”**. Інформацію про реалізовані моделі прогнозування та їх апробацію подано у вигляді тез доповіді на вищезазначеному заході.

# РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ЇЇ ВИРІШЕННЯ

## 1.1 Аналіз об'єкту дослідження

### 1.1.1 Роль телебачення у медіапросторі

Телебачення традиційно є одним із головних каналів масової інформації, який формує суспільну думку, інформує громадськість про актуальні події та задовольняє потребу в розвагах. Воно охоплює широку аудиторію, незалежно від віку, соціального статусу чи регіону проживання. У багатьох країнах світу телебачення залишається провідним медіа за загальним обсягом споживаного контенту, незважаючи на розвиток цифрових технологій та поширення інтернету.

Протягом останніх двох десятиліть характер споживання телевізійного контенту зазнав істотних змін. Якщо раніше домінувало лінійне телебачення — перегляд програм згідно з сіткою мовлення телеканалу, — то сьогодні все більшої популярності набувають моделі, що передбачають споживання контенту «на вимогу» (Video on Demand).

Глядачі очікують персоналізації та доступу до улюблених програм на різних пристроях — телевізорах, смартфонах, планшетах, комп'ютерах. Паралельно з цим розвиваються онлайн-платформи й стримінгові сервіси, такі як YouTube, Netflix, Megogo, що стають альтернативними або додатковими джерелами відеоконтенту. Це призводить до розмивання аудиторії традиційного телебачення та необхідності адаптувати методи її вимірювання до нових умов.

Тим не менш, певні типи контенту — зокрема новини, прямі трансляції спортивних подій, політичних шоу, національних чи культурних заходів — все ще зберігають високу популярність саме у форматі прямого ефіру. Це створює

потребу у точному та оперативному вимірюванні теледивлення для забезпечення актуальних управлінських рішень у сфері медіа.

### 1.1.2 Процес вимірювання теледивлення

Телевізійне вимірювання базується на панельному дослідженні, що є однією з найефективніших методологій вивчення глядацької поведінки. В Україні провідною організацією, що здійснює вимірювання теледивлення, є компанія Nielsen, яка працює на замовлення Індустріального телевізійного комітету (ІТК). **Телевізійна панель** – це спеціально сформована вибірка домогосподарств, за допомогою якої відстежується перегляд телевізійного контенту на індивідуальному рівні.

**Панельне дослідження** передбачає, що участь у ньому бере постійна група людей. Це дозволяє детально аналізувати зміни глядацьких вподобань, переміщення аудиторії між телеканалами, а також оцінювати ефективність рекламних кампаній. Завдяки такому підходу можна зрозуміти, як і коли саме глядачі перемикаються з однієї передачі на іншу, що дозволяє точніше будувати комунікаційні стратегії.

Дані збираються на рівні так званих **телевізійних домогосподарств**. Телевізійним домогосподарством вважається будь-яке житло, де є хоча б один робочий телевізор із платформою мовлення, який переглядався хоча б один раз за останні шість місяців. До вибірки входять лише ті домогосподарства, мешканці яких ведуть спільне господарство та проживають разом на постійній основі. Водночас не враховується **інституційне населення** — люди, які проживають у студентських гуртожитках, готелях, лікарнях, військових частинах, тюрмах, будинках для осіб похилого віку та інших інституціях.

Для формування панелі застосовується **установче дослідження**, яке в Україні проводить Індустріальний телевізійний комітет, метою якого є визначення соціально-демографічних характеристик потенційних учасників і їх

телевізійного обладнання. На цьому етапі визначаються домогосподарства, які найбільш репрезентативно відображають генеральну сукупність. Репрезентативність означає, що поведінка обраних людей відповідає типовій поведінці населення в межах визначених демографічних та технічних характеристик:

- Регіон та розмір населеного пункту;
- Кількість телевізорів у домогосподарстві;
- Розмір домогосподарства;
- Наявність людей похилого віку;
- Тип прийому сигналу (ефірний, кабельний, супутниковий).

Додатково, на індивідуальному рівні враховують стать та вікову групу (наприклад, 4-17, 18-34, 35-54, 55+ років).

Вимірювання телеперегляду здійснюється за допомогою спеціальних пристроїв – **піплметрів**. Це спеціальне зовнішнє обладнання, яке фіксує факт увімкнення телевізора, джерело сигналу, а також реєструє присутність глядача. Глядачі лише повинні підтвердити свою присутність у кімнаті за допомогою реєстрації. Сама реєстрація перегляду є автоматичною, за допомогою технології **аудиометчингу**.

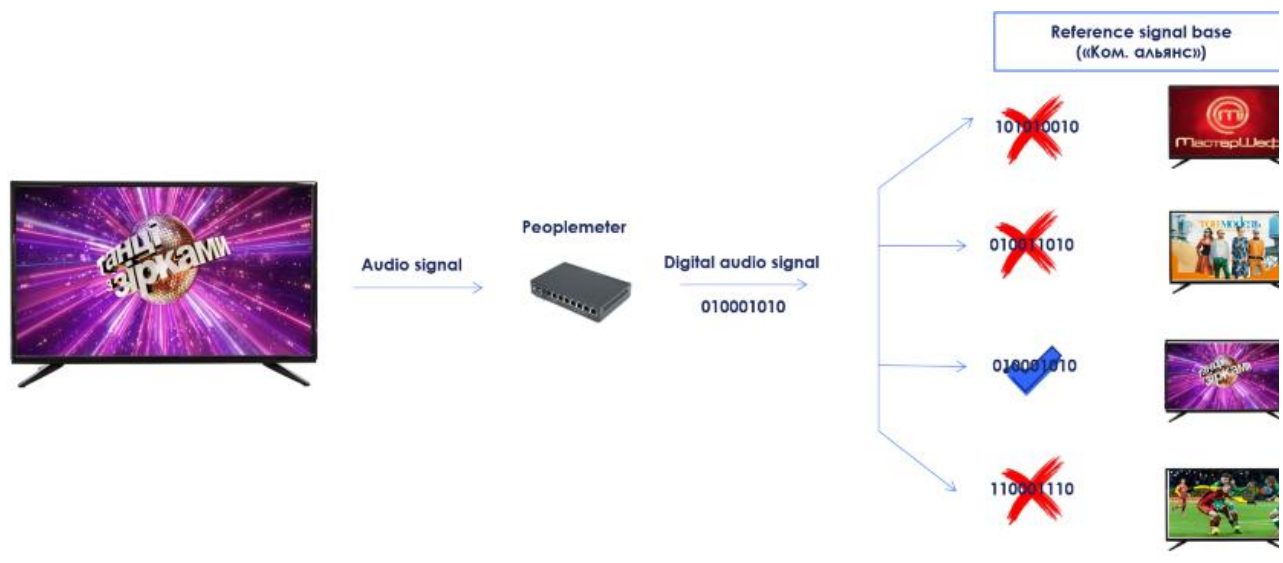


Рисунок 1.1— Методологія аудіометчингу

Аудіометчинг полягає в ідентифікації телевізійного каналу за унікальним цифровим ключем (цифровим відбитком аудіосигналу). Ця технологія забезпечує точну ідентифікацію контенту незалежно від платформи мовлення чи провайдера. Щосекунди звуковий ряд телевізійного сигналу перетворюється на унікальний цифровий ключ, який порівнюється з референтною базою сигналів, отриманих від каналів-учасників. Це дозволяє чітко встановити, який саме контент переглядався в конкретний момент часу.

Після формування ключів дані щоденно передаються із піпметрів до центральної станції обробки. Передача відбувається по мобільному зв'язку в зашифрованому вигляді, що забезпечує надійність і конфіденційність інформації.

Уся інформація обробляється централізовано – в Україні та більшості інших країн це відбувається на серверах компанії Nielsen у США. Дані проходять через наступні етапи:

- Передача і збирання інформації (овернайт і фінальні дані).
- Верифікація даних на відповідність та коректність.
- Валідація та редагування даних (контроль якості).
- Конвертація та поєднання з інформацією щодо контенту, отриманою від місцевих комунікаційних альянсів.

Панель зазвичай охоплює міста з населенням понад 50 тисяч осіб. Через обмеженість ресурсів, замість того, щоб вимірювати все населення, панель складається з вибірки близько 1000 домогосподарств, яка репрезентує більшу кількість людей (наприклад, близько 4,5 мільйонів осіб у містах з населенням понад 50 тисяч осіб).

Телевізійне вимірювання дає широкі можливості для детального аналізу аудиторії за різними характеристиками.

Таблиця 1.1. Характеристики аудиторій

| Індивідуальні характеристики             | Характеристики сім'ї       | Додаткові характеристики                     |
|--|----------------------------|--|
| Стать/Вік (4+)                           | Розмір міста (50К+, 500К+) | Тип прийому                                  |
| Сімейний стан                            | Регіон                     | Кількість ТБ в сім'ї                         |
| Освіта                                   | Область                    | Доступ до аналог./цифр. ефіру                |
| Зайнятість                               | Основна мова               | Наявність DVD                                |
| Зайнятість на роботі                     | Кількість членів сім'ї     | Наявність ігрової приставки                  |
| Непрацюючі                               |                            | Інтернет вдома                               |
| Приймає основні рішення у сім'ї          |                            | Наявність телефону                           |
| Індивідуальна основна мова (з 6.04.2015) |                            | Комп'ютер                                    |
|  |                            | Наявність дітей віком 0-3, 4-13, 14-17 років |

Отримані дані використовуються для:

- Аналізу глядацької поведінки та переміщення аудиторії;
- Оцінки охоплення телепередач та рекламних кампаній;
- Побудови та аналізу цільових аудиторій для рекламних кампаній;
- Оцінки ефективності телевізійного контенту та рекламних роликів;
- Вивчення тенденцій перегляду та планування програмування телеканалів.

### 1.1.3 Основні показники теледивлення

Перш ніж переходити до телевізійних показників, варто з'ясувати, що таке «аудиторія» та «цільова аудиторія». **Аудиторія** в контексті телемовлення – це сукупність людей, які переглядають конкретний телеканал або певний контент у

зазначений проміжок часу. При цьому кожен глядач належить до певної демографічної, соціальної, географічної чи іншої групи.

**Цільова аудиторія (ЦА)** – це підгрупа глядачів, що має конкретні характеристики (наприклад, вік, стать, регіон проживання, рівень доходу) і становить безпосередній інтерес для дослідника або рекламодавця. Усі телевізійні показники розраховуються саме щодо визначеної цільової аудиторії (наприклад, «усі глядачі 18–54», «жінки 25–45» тощо).

**Рейтинг** (позначають як RAT% або TV Rating, TVR) відображає відсоток людей із вибраної цільової аудиторії, які дивилися певний телеканал чи передачу в заданий період. Рейтинг дозволяє оцінити привабливість каналу або передачі серед заданої аудиторії. На основі рейтингу планують оптимальні часові слоти для розміщення реклами.

Формула рейтингу:

$$RAT\% = \frac{\text{Кількість глядачів каналу із ЦА}}{\text{Загальна кількість осіб у ЦА}} \times 100\% \quad (1.1)$$

Коли рахуємо сумарний результат усіх виходів ролика, отримуємо **GRP (Gross Rating Points)** — це арифметична сума окремих RAT за кожен вихід. Вона відповідає на запитання «скільки разів у середньому реклама могла бути побачена всією аудиторією» і не обмежується сотнею, бо один глядач може бачити ролик багато разів:

$$GRP = \sum_{i=1}^k RAT\%_i \quad (1.2)$$

Якщо ж нас цікавить саме вузька цільова аудиторія (наприклад, жінки 25–45 років), ми оперуємо **TRP (Target Rating Points)**. Формально формула та сама, але RAT береться по ЦА, тож:

$$TRP = \sum_{i=1}^k RAT \quad (1.3)$$

Через те, що на українському TV-ринку ціни виставляють за 30-секундний ролик, виникає потреба зрівняти рейтинги різних хронометражів. Так виникає **EqGRP**— «еквівалентний GRP». Ми коригуємо кожен рейтинг за коефіцієнтом

довжини ролика (з таблиці нижче) і вже потім підсумовуємо. Таким чином, у найкоротшому вигляді еквівалентний рейтинг можна показати як добуток загального GRP і коефіцієнта хронометражу:

$$EqGRP = GRP \times C_{length} \quad (1.4)$$

Де  $C_{length}$  – відповідний коефіцієнт із таблиці (наприклад, 0.70 для 15-секундного ролика).

Таблиця 1.2. Коефіцієнти переведення хронометражів

| Хронометра<br>ж | 5"   | 10"  | 15"  | 20"  | 25"  | 30"  | 35"  | 40"  | 45"  | 50"  | 55"  | 60"  |
|-----------------|------|------|------|------|------|------|------|------|------|------|------|------|
| Коефіцієнт      | 0.30 | 0.50 | 0.70 | 0.90 | 0.95 | 1.00 | 1.25 | 1.40 | 1.60 | 1.80 | 1.90 | 2.00 |

Нарешті, **Affinity Index (AFF%)** показує, наскільки для цільової аудиторії характерний контакт з певною подією, ніж для будь-якої іншої аудиторії (наприклад, закупівельної). Якщо індекс  $>100\%$ , трансляція «краще працює» саме на ЦА;  $<100\%$ — гірше. У простій формі це відношення рейтингів цільової й закупівельної аудиторії:

$$AFF\% = \frac{RAT\%_{ЦА}}{RAT\%_{БА}} \times 100\% \quad (1.5)$$

Для рекламної кампанії формула матиме вигляд:

$$AFF\% = \frac{TRP}{GRP} \times 100\% \quad (1.6)$$

Ці показники дозволяють описувати ефективність програмного контенту, порівнювати канали між собою, здійснювати сегментацію аудиторії та планувати рекламні кампанії.

## 1.2 Постановка задачі прогнозування показників теледивлення

### 1.2.1 Особливості часових рядів теледивлення

Показники теледивлення, зокрема рейтинг, частка, охоплення, Affinity та інші, зазвичай представлені у вигляді часових рядів — впорядкованих у часі послідовностей числових значень, які відображають динаміку медіаактивності глядачів. Такі ряди можуть мати різну частоту: від похвилинної до щотижневої або щомісячної. Їм притаманні певні характерні властивості, зокрема сезонність, трендові компоненти, наявність структурних зламів та випадкових коливань.

Сезонність проявляється у вигляді регулярних коливань, що повторюються упродовж доби, тижня або року — наприклад, підвищення глядацького інтересу у вечірній прайм-тайм, відмінності між буднями та вихідними, а також зміни, пов'язані з літніми відпустками, святами чи початком навчального року. Тренди, у свою чергу, відображають довгострокові тенденції зростання або падіння інтересу до певних каналів, жанрів або форматів програм.

У часових рядах теледивлення також часто фіксуються раптові зміни, викликані зовнішніми подіями — виходом нової програми, трансляцією резонансної події або політичними та соціальними кризами. Окрім того, навіть за відсутності явних змін, показники можуть суттєво коливатись через складну й часто непередбачувану поведінку глядачів. Така природа даних вимагає побудови гнучких моделей, які здатні адаптуватися до складної структури та нестабільності часових рядів.

### 1.2.2 Практичні задачі та кейси прогнозування: маркетинг, програмне планування, закупівлі реклами

Задача прогнозування телепоказників має прикладне значення для багатьох сфер діяльності медіаринку. У маркетингу телеканалів оцінка рейтингу та інших показників дозволяє планувати ефективні комунікаційні стратегії,

запускати або адаптувати медіапродукти, визначати ефективність промокампаній.

У медіаплануванні та рекламній аналітиці прогнози GRP, охоплення або індексу Affinity необхідні для побудови рекламних кампаній, визначення вартості розміщення і підбору найефективніших слотів трансляції.

У програмному плануванні телеканалів аналіз теледивлення дозволяє краще управляти сіткою мовлення: змінювати розклад передач, запускати нові формати в оптимальні години, уникати конкурентного перетину з сильнішими програмами на інших каналах.

1.2.3 Виклики у прогнозуванні: сезонність, тренди, особливі події, різкі зміни, розмір даних

Водночас побудова ефективних моделей прогнозування стикається з рядом викликів. Одним з них є складна **багаторівнева сезонність** — наприклад, одночасна дія добових, тижневих і річних циклів. Крім того, поведінка глядача нестабільна: вона може змінюватися під впливом нових цифрових платформ, змін способу життя чи навіть подій зовнішнього характеру.

Важливим фактором є також **непередбачуваність подій** — зокрема спортивних трансляцій, концертів, надзвичайних ситуацій або змін політичного порядку денного. Такі події мають значний вплив на структуру перегляду, але часто не можуть бути заздалегідь враховані у стандартних статистичних моделях.

Додатковою проблемою є те, що класичні моделі можуть виявляти високі результати на історичних даних, але не давати достовірного прогнозу на майбутнє через явище перенавчання. Це вимагає не лише правильного підбору методів, а й розробки гнучких адаптивних технологій, здатних оновлювати прогнози в реальному часі, використовувати дані з кількох джерел (наприклад, телепанелі та онлайн-аналітики), а також враховувати контекстні зміни.

Також значно ускладнює задачу те, що через розпочате 24 лютого 2022 року повномасштабне вторгнення російської федерації на територію України, вимірювання показників було тимчасово припинене і відновилося лише з 1 січня 2023 року. Для точнішого прогнозування показників важливо мати якомога довший часовий ряд, а внаслідок таких обставин дані за період більше ніж 9 місяців є недоступними. Для вирішення задач, які потребують оцінки щомісячних чи річних значень, інформації може бути недостатньо. Також через зміну глядацьких звичок та самої структури телевізійних домогосподарств, попередні дані втрачають актуальність та можуть лише вносити похибку.

Загалом, задача прогнозування показників теледивлення є складною, але надзвичайно актуальною в умовах цифрової трансформації медіа. Її вирішення потребує застосування сучасних методів науки про дані, поєднання статистичних та машинних підходів, а також створення інформаційної технології, що може інтегруватися у практичну діяльність гравців медіаринку.

### **1.3 Огляд сучасних методів прогнозування в телеметрії**

#### **1.3.1 Класичні статистичні методи: ARIMA**

Одним із найпоширеніших класичних підходів до прогнозування часових рядів є модель ARIMA (AutoRegressive Integrated Moving Average). Вона ґрунтується на поєднанні трьох компонентів: авторегресії (AR), яка враховує залежність поточного значення від попередніх спостережень; інтегрування (I), що відповідає за приведення ряду до стаціонарного вигляду шляхом диференціювання; та ковзного середнього (MA), яке моделює залежність від попередніх похибок. Модель може бути розширена до SARIMA для обліку сезонних компонентів, що особливо важливо в телеметрії, де сезонність має добову, тижневу та річну структуру.

ARIMA є легкою для розуміння і відносно простою в реалізації, що робить її зручною для закупівельного прогнозування. Вона добре працює при коротких горизонтах передбачення і на стаціонарних рядах, проте має обмеження при роботі зі складними, багатокomпонентними структурами або в умовах сильних зовнішніх впливів.

### 1.3.2 Експоненційні моделі згладжування: ETS

Іншим класичним підходом є моделі експоненційного згладжування, зокрема ETS (Error-Trend-Seasonality), які моделюють часовий ряд як комбінацію тренду, сезонності та випадкової компоненти. Модель може бути як адитивною, так і мультиплікативною, що дозволяє адаптувати її до різної природи варіацій. ETS автоматично обирає оптимальну комбінацію компонентів та метод оцінки (наприклад, з використанням інформаційного критерію AIC), що робить її гнучкою і придатною для автоматичного прогнозування у системах реального часу.

ETS відзначається високою точністю в умовах вираженої регулярної сезонності, таких як тижнева структура теледивлення. Вона менш чутлива до шуму, ніж ARIMA, і забезпечує гладкий прогноз без надмірної складності. Однак, на відміну від машинного навчання, вона не враховує зовнішні фактори чи додаткові ознаки.

### 1.3.3 Байєсівсько-евристична модель Prophet

Модель Prophet, розроблена Facebook, стала популярним інструментом для прогнозування бізнес-часових рядів завдяки своїй гнучкості, автоматичній обробці сезонності та зручності в застосуванні. Prophet моделює ряд як суму тренду, сезонності (добової, тижневої, річної) та впливу свят або спеціальних подій. Модель може враховувати зовнішні регресори, підтримує як адитивні, так і мультиплікативні сезонні компоненти, а її параметри легко інтерпретуються.

Особливо зручною є можливість введення “holiday effects” — календарних подій, які істотно впливають на поведінку глядачів.

У контексті телеметрії Prophet дає змогу врахувати періодичну структуру теледивлення та адаптувати прогноз до зміни розкладу мовлення, святкових ефірів або спортивних подій. Завдяки robust-фільтрації Prophet є стійким до аномалій і здатен забезпечувати стабільний прогноз навіть при нерівномірній вибірці. Основним обмеженням є припущення адитивної структури та складність у моделюванні взаємодій між компонентами, що іноді знижує точність при складній поведінці ряду.

#### 1.3.4 Глибинне навчання: LSTM

Для прогнозування медіапоказників у складних умовах, зокрема при наявності нелінійностей, відкладених ефектів і багатьох залежностей, зручними є моделі глибинного навчання. Однією з провідних архітектур є LSTM (Long Short-Term Memory) — різновид рекурентних нейронних мереж, який дозволяє зберігати довготривалі залежності в часових рядах завдяки спеціальним структурам — «коміркам пам’яті» та «гейтам» (воротам).

LSTM здатна враховувати як короткочасні, так і довготривалі закономірності поведінки даних, включаючи сезонні цикли, тренди, вплив попередніх ефірів та комбіновані залежності. Вона працює без необхідності попередньої трансформації ряду до стаціонарного вигляду, автоматично навчається на історичних даних і забезпечує високу точність за наявності великої кількості навчальних прикладів.

Застосування LSTM у телеметрії дозволяє моделювати складні динамічні залежності між показниками перегляду, особливо при аналізі поєднаних рядів (наприклад, перегляд по різних платформах, регіонах або демографічних групах). Недоліками є висока чутливість до налаштувань гіперпараметрів,

потреба у великій кількості даних, а також обмежена інтерпретованість результатів, що може ускладнювати їх використання в бізнес-практиках.

### 1.3.5 Порівняння ефективності підходів для даних теледивлення

Вибір методу прогнозування залежить від специфіки задачі, обсягу наявних даних, вимог до точності та часу обчислень. Класичні статистичні методи залишаються значущими у випадках обмежених ресурсів або при потребі швидкого зрозумілого результату. Вони мають прості параметри, зрозумілу логіку та низькі вимоги до обчислень, однак погано справляються з нестационарністю та багатовимірними структурами.

Машинне навчання є ефективним у задачах, де можна сформулювати вхідні ознаки, а також там, де важливі нетривіальні взаємозв'язки між факторами. Його недоліком є ризик перенавчання і необхідність постійної перевірки. Глибинні нейромережі, зокрема LSTM і TFT, демонструють найвищу точність у складних сценаріях — як при прогнозуванні поєднаних ефірних і цифрових даних, багатоканального мовлення або динаміки у спеціальних проектах. Натомість, їх впровадження потребує значних обчислювальних потужностей, ретельного налаштування архітектури та великої кількості даних.

У практиці прогнозування показників теледивлення найкращий результат часто досягається шляхом поєднання підходів — наприклад, використання статистичної моделі для виявлення закупівельних патернів, доповненого машинним або глибинним методом для уточнення складних сценаріїв. Це підкреслює важливість побудови гнучкої, модульної системи прогнозування, здатної адаптуватися до зміни умов та завдань.

## 1.4 Постановка задачі

**Метою даного дослідження є розробка моделей та інформаційної технології прогнозування показника *Affinity*, яка дозволить суттєво підвищити**

точність планування в межах діяльності медіа-агентства повного циклу. У зв'язку з постійними змінами в структурі телевізійної аудиторії, а також високою конкуренцією між телеканалами за увагу цільових груп, виникає гостра потреба в інструментах, здатних прогнозувати поведінку глядачів із достатньо високою точністю та адаптивністю. На основі проведеного аналізу можна виокремити низку ключових проблем, які ускладнюють процес побудови ефективного прогнозу Affinity.

**По-перше,** це відсутність сталості у поведінці аудиторії — вподобання глядачів можуть змінюватися під впливом зовнішніх факторів (сезон, новини, події, зміна програмної сітки тощо), що формує складну сезонність і нерівномірність у часових рядах показників. Це вимагає використання моделей, здатних враховувати довгострокові та короткострокові зміни в трендах.

**По-друге,** на практиці відсутній універсальний метод прогнозування: моделі типу ARIMA добре працюють для стаціонарних процесів, однак вимагають ретельного попереднього аналізу і трансформації даних; метод ETS забезпечує добру інтерпретацію зібраних даних, проте обмежений за гнучкістю; Prophet, хоча й адаптований до сезонності та свят, може давати некоректні оцінки при зміні трендів; нейромережеві підходи на основі LSTM дозволяють обробляти складні послідовності, однак вимагають великого обсягу даних, часу на налаштування та глибшого розуміння гіперпараметрів. Таким чином, постає потреба у порівняльному аналізі зазначених підходів саме на прикладі реальних даних медіа-агентства.

**Третьою проблемою** є обмежена інтеграція прогнозних моделей у бізнес-процеси агентства: результати прогнозів часто не доходять до осіб, що приймають рішення (медіабаєри, аналітики), або не представлені у зручній для них формі, що знижує цінність навіть точного прогнозу.

**Четверта складність** стосується відсутності єдиної автоматизованої системи обробки даних і формування прогнозів. На практиці обробка CSV-файлів, різноманітні трансформації, тренування моделей і виведення результатів

часто відбувається вручну, що знижує ефективність роботи аналітиків і підвищує ризик помилок.

Саме тому, з метою подолання вказаних проблем, у межах дослідження передбачається реалізувати наступні ключові етапи:

- Провести порівняльний аналіз ефективності різних моделей прогнозування (ARIMA, ETS, Prophet, LSTM) на основі реальних щоденних даних теледивлення. Оцінка здійснюватиметься за поширеними метриками точності: MAPE, MAE, RMSE, що дозволить об'єктивно порівняти результати та визначити найбільш доцільну модель для подальшого впровадження.
- Створити комплексну IT-систему для автоматизованого прогнозування телеперегляду, що охоплює імпорт даних із Yumianalytics, попередню обробку, налаштування моделей, формування прогнозів і збереження результатів, реалізовану в Python із бібліотеками pandas, statsmodels, Prophet, TensorFlow/Keras та scikit-learn.
- Інтегрувати згенеровані прогнозні дані у процес медіапланування рекламних кампаній, розробивши аналітичний інструментарій у Power BI для візуалізації результатів моделей.
- Автоматизувати цикл отримання, обробки й публікації даних *Affinity*, забезпечивши регламентне завантаження сирих даних, запуск скриптів прогнозування та передачу оновлених метрик у Power BI для кінцевих користувачів.

**Отже,** дослідження має як теоретичне значення — в аспекті аналізу ефективності моделей прогнозування часових рядів, так і прикладну спрямованість — у вигляді створення практичного інструменту для потреб медіа-індустрії.

## 1.5 Висновки до розділу 1

У цьому розділі було здійснено комплексний аналіз предметної області прогнозування теледивлення та проведено огляд сучасних підходів до вимірювання й моделювання відповідних показників. Телебачення, незважаючи на цифрову трансформацію медіапростору, залишається важливою складовою інформаційного ландшафту та ефективним каналом комунікації з широкими масами аудиторії. Зміна форм споживання контенту, поява нових платформ і зростання конкуренції між джерелами інформації зумовлюють потребу у точному вимірюванні та прогнозуванні поведінки глядачів. Основні медіапоказники, такі як **рейтинг, частка, охоплення, GRP та Affinity**, формують основу медіааналітики і використовуються для прийняття управлінських рішень на рівні телеканалів, рекламодавців і дослідницьких агентств. Їх вимірювання здійснюється за допомогою панельних телеметричних досліджень, що базуються на даних від репрезентативних домогосподарств і забезпечують регулярний, стандартизований збір інформації з високою деталізацією. Показники теледивлення мають форму часових рядів, яким властиві сезонні коливання, тренди, структурні зміни й випадкові відхилення. Це створює складні умови для їх прогнозування, оскільки модель має не лише відтворювати історичну динаміку, але й адаптуватися до контексту — змін у контенті й поведінці аудиторії, а також зовнішніх подій. Прикладні задачі прогнозування охоплюють маркетинг, програмне планування, медіапланування та соціально-політичні проєкти, і мають практичне значення для стратегічного управління медіаресурсами. Огляд методів прогнозування підтвердив, що класичні статистичні моделі, такі як ARIMA та експоненційне згладжування, залишаються актуальними для закупівельного моделювання, тоді як сучасні задачі вимагають використання машинного навчання (Random Forest, XGBoost) та глибоких нейронних мереж (LSTM, Temporal Fusion Transformer). Комбінування різних підходів часто дає найкращі результати, оскільки дозволяє

врахувати як глобальні риси, так і локальні аномалії в даних. Для практичного впровадження моделей прогнозування необхідно створити ефективну інформаційну технологію, що включає засоби збору, обробки та візуалізації даних, інтеграцію моделей, автоматичне оновлення результатів і зручний інтерфейс у користувача. Такі технології базуються на інструментах Python, R, SQL та спеціалізованих бібліотеках для роботи з часовими рядами та машинним навчанням. **Таким чином**, у результаті проведеного аналізу було обґрунтовано доцільність застосування методів науки про дані для прогнозування показників теледивлення, визначено основні виклики та потреби, що ляжуть в основу подальшої розробки моделей та інформаційних технологій в наступних розділах дипломної роботи.

## РОЗДІЛ 2. МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИКА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

### 2.1 ARIMA-модель для прогнозування часових рядів

У розбудові сучасних методів прогнозування часових рядів модель ARIMA (Autoregressive Integrated Moving Average) посідає провідне місце вже понад півстоліття. По суті, ARIMA є кульмінацією багатьох історичних напрацювань у сфері статистичного аналізу: від перших лінійних автогресійних моделей кінця 1920-х років до комплексного підходу Бокса–Дженкінса, який був узагальнений у їхній класичній праці «Time Series Analysis: Forecasting and Control». Саме в цій роботі Джордж Бокс і Гвілім Дженкінс систематизували процедуру побудови моделей, які могли б адекватно описувати як стаціонарні, так і нестаціонарні часові ряди; а введення диференціювання (інтеграції) як засобу перетворення нестаціонарних рядів на стаціонарні стало ключовим нововведенням книги.

Грунтовна методологія Бокса–Дженкінса включає три етапи: ідентифікацію (визначення порядків  $p$ ,  $d$ ,  $q$  за аналізом автокореляційної і часткової автокореляційної функцій), оцінювання параметрів (найчастіше за методом максимальної правдоподібності) та валідацію моделі (перевірка залишків на «білий шум» з використанням критерію Л'юнга–Бокса та візуальним аналізом автокореляції залишків).

Авторегресійну частину моделі ARIMA описують рівнянням:

$$y_t = c + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (2.1)$$

Замість того, щоб використовувати попередні значення прогнозованої змінної, модель ковзного середнього використовує минулі помилки прогнозу:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} \quad (2.2)$$

Комбінація цих двох підходів дала змогу сформулювати модель ARMA, яка застосовується до стаціонарних часових рядів. Однак реальні часові ряди рідко бувають стаціонарними. Для розв'язання цієї проблеми Бокс і Дженкінс

запропонували використовувати оператор диференціювання, що перетворює нестационарний ряд на стаціонарний. У результаті отримують узагальнену модель ARIMA(p,d,q):

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2.3)$$

Для сезонних даних модель розширюють до SARIMA(p,d,q)(P,D,Q)<sub>s</sub>, додаючи сезонні оператори і сезонне диференціювання з періодом s:

$$(1 - \phi_1 B) (1 - \Phi_1 B^s) (1 - B) (1 - B^s) y_t = (1 + \theta_1 B) (1 + \Theta_1 B^s) \varepsilon_t \quad (2.4)$$

Діагностика моделі включає також перевірку залишків на незалежність та відсутність автокореляції. Для цього використовують модифікацію Q-тесту Бокса-Пірса — критерій Л'юнга-Бокса:

$$Q' = n(n+2) \sum_{j=1}^p \frac{r_j^2}{n-j} \quad (2.5)$$

Інформаційний критерій Акаїке (AIC) допомагає вибрати порядок моделі:

$$AIC = -2 \log(L) + 2(p + q + k + 1) \quad (2.6)$$

Для малих вибірок використовують скоригований критерій AIC<sub>c</sub>:

$$AIC_c = AIC + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2} \quad (2.7)$$

Баєсівський інформаційний критерій (BIC або SBC) має вигляд:

$$BIC = AIC + [\log(T) - 2] (p + q + k + 1) \quad (2.8)$$

ARIMA залишається одним із найнадійніших інструментів короткострокового прогнозування, зокрема для таких показників, як Affinity, де важливі легке пояснення та обґрунтованість висновків.

## 2.2 Метод експоненційного згладжування (ETS)

У класичному підході до прогнозування часових рядів експоненційне згладжування виникло як природне продовження методів згладжування, що спочатку пропонувалися Брауном і Голтом у середині ХХ століття. Ідея полягала в тому, щоб простим експоненціальним способом надавати більшої ваги

останнім спостереженням, водночас поступово «забуваючи» старі значення. Проте ранні форми експоненційного згладжування враховували лише рівень ряду (single smoothing) або рівень і лінійний тренд (Holt's linear method), не забезпечуючи адекватного моделювання сезонних коливань. У 2002 році науковці узагальнили ці підходи в єдиному просторі, що став називатися ETS – Exponential smoothing State space model [11].

У загальному вигляді сімейство ETS-моделей складається з трьох компонентів, кожна з яких може бути реалізована в адитивному (A) чи мультиплікативному (M) варіанті (або відсутня, N). Позначення ETS(E,T,S) означає:

- E (Error) — тип похибки (A або M)
- T (Trend) — наявність і тип тренду (N, A, Ad, M)
- S (Seasonal) — наявність і тип сезонності (N, A, M)

Компоненти моделі:

- $l_t$  — рівень (середнє значення в момент часу  $t$ )
- $b_t$  — тренд (зміна рівня)
- $s_t$  — сезонність
- $y_t$  — фактичне значення

У мультиплікативних моделях ETS(M,A,M) компоненти поєднуються через добуток:

$$y_t = (l_{t-1} + b_{t-1}) s_{t-m} (1 + \varepsilon_t), l_t = (l_{t-1} + b_{t-1}) (1 + \alpha \varepsilon_t), b_t = b_{t-1} + \beta (l_{t-1} + b_{t-1}) \varepsilon_t, s_t = s_{t-m} (1 + \gamma \varepsilon_t). \quad (2.9)$$

Розрізнення адитивності та мультиплікативності полягає в тому, що в адитивних моделях зміни тренду або сезонності не залежать від рівня ряду, тоді як у мультиплікативних вони масштабуються рівнем [9].

Процедура побудови ETS-моделі:

1. **Ініціалізація** — параметри оцінюють на початковому підборі даних або задають за допомогою методу моментів.

2. **Оцінка параметрів** — максимізація правдоподібності (MLE) з AICc як критерієм відбору:

$$AICc = -2\log(L) + 2(p + k) + \frac{2(p+k)(p+k+1)}{T-p-k-1} \quad (2.10)$$

3. **Авто-вибір структури** — порівняння всіх комбінацій ETS-моделей за AICc [1].
4. **Прогнозування** — багатокроковий прогноз:

$$\hat{y}_{t+h|t} = f(l_t, b_t, s_{t-m+((h-1) \bmod m)}, \dots) \quad (2.11)$$

ETS-моделі широко застосовуються в економічному прогнозуванні, роздрібних продажах, логістиці та медіаплануванні (наприклад, для моделювання циклічних коливань Affinity). Порівняно з ARIMA, ETS не вимагає диференціювання та дає чітку інтерпретацію кожного компонента. Для складних автокореляційних структур часто комбінують ETS з ARIMA або нейромережами (LSTM), але ETS залишається ефективним закупівельним рішенням із високою інтерпретованістю [15].

### 2.3 Модель LSTM

У 1997 році Шмідхубер та Гохрайтер вперше запропонували модель Long Short-Term Memory (LSTM). Раніше в нейронних мережах для обробки послідовностей (рекурентних нейронних мережах, або RNN) виникала серйозна проблема: коли модель намагалась «пам'ятати» інформацію з минулих кроків (далеко в часі), то під час навчання градієнти або зникали, або ставали надто великими, що призводило до втрати пам'яті довгих залежностей[8].

В основі LSTM лежить ідея вбудованої «пам'яті», здатної зберігати інформацію протягом довгих інтервалів, та набір керуючих механізмів — вхідний клапан  $i_t$ , клапан забуття  $f_t$  і вихідний клапан  $o_t$ . На кожному кроці мережа отримує вхід  $x_t$  і попередній вихід  $h_{t-1}$ , після чого обчислює:

$$\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), o_t \\
&= \sigma(W_o x_t + U_o h_{t-1} + b_o), \tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c), c_t \\
&= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, h_t = o_t \odot \tanh(c_t). \quad (2.12)
\end{aligned}$$

Вхідний вентиль  $i_t$  визначає, яка частина нового контенту  $\tilde{c}_t$  потрапить у внутрішній стан, вентиль забуття  $f_t$  — яку частину  $c_{t-1}$  слід зберегти, а вихідний вентиль  $o_t$  — яка частина оновленого стану  $c_t$  піде на вихід  $h_t$ . Цей механізм долає проблему зникнення градієнта через Constant Error Carousel[4].

Пізніші модифікації розширили закупівельну архітектуру:

- Gated Recurrent Unit (GRU) з об'єднаним вентилям оновлення[3].
- Двонаправлені LSTM (BiLSTM) для кращого контексту[
- Паралельні блоки в xLSTM для масштабованості[2].

LSTM довела свою ефективність у розпізнаванні мовлення[7], машинному перекладі[17], а також у прогнозуванні часових рядів[13]. Головні переваги LSTM над класичними ARIMA/ETS:

1. Обробка нелінійних і довготривалих залежностей;
2. Адаптивні механізми вентилів для динамічного виділення релевантного контексту;
3. Менше ручного налаштування порівняно з Prophet.

Недоліки:

1. Велика обчислювальна вартість і потреба в ВРПТ.
2. Схильність до перенавчання на маленьких наборах (вирішується дропаутами, L2-регуляризацією, ранньою зупинкою)[6].

Для оцінки якості прогнозу використовуються ті ж метрики (MAE, RMSE, MAPE, MASE), що й для ARIMA/ETS. Порівняння показує, що LSTM краще захоплює короткострокові коливання та пом'якшує ефект різких змін[5].

## 2.4 Prophet

У 2015–2017 роках інженери та дослідники з команди Core Data Science компанії Facebook (нині Meta) під керівництвом Шона Тейлора та Бенджаміна Летема розробили процедуру прогнозування під назвою Prophet, покликану спростити завдання часових рядів «на масштабах» — коли потрібно генерувати й контролювати тисячі прогнозів для різних бізнес-процесів[18]. Основна ідея полягала в поєднанні статистичних підходів з інтуїтивно зрозумілими параметрами, що дозволило аналітикам легко налаштувати «скелет» моделі без глибокого занурення в математичну специфіку[21].

Принципова архітектура Prophet базується на адитивній декомпозиції часової серії:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (2.13)$$

- $g(t)$  — нелінійний тренд з виявленням точок переходу (changepoints) або логістичним обмеженням місткості;
- $s(t)$  — сезонні компоненти (річна, тижнева, добова) через ряди Фур'є;
- $h(t)$  — ефекти свят та подій;
- $\varepsilon_t$  — випадкова похибка, яка моделюється як нормальний шум[19].

**Тренд  $g(t)$**  у вигляді логістичної кривої:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))} \quad (2.14)$$

де  $C$  — місткість,  $k$  — швидкість росту,  $m$  — точка середини логістичної кривої[19]. При кусково-лінійній формі модель автоматично обирає потенційні точки переходу та регулює їх вплив.

**Сезонність** через ряди Фур'є:

$$s(t) = \sum_{n=1}^N (a_n \cos(2\pi nt/P) + b_n \sin(2\pi nt/P)) \quad (2.15)$$

де  $P$  — період,  $N$  — число гармонік,  $a_n, b_n$  — оцінювані коефіцієнти[19].

**Святкові ефекти  $h(t)$**  описуються індикаторними змінними з можливістю продовження впливу на сусідні дні[19].

### **Workflow** у Python чи R:

1. Імпорт даних у DataFrame з колонками ds та y.
2. Ініціалізація моделі:

```
from prophet import Prophet
model = Prophet(
    growth='linear',
    yearly_seasonality=True,
    weekly_seasonality=True,
    daily_seasonality=False,
    holidays=holidays_df
)
```

3. Навчання: `model.fit(df)`
4. Генерація майбутніх дат: `future = model.make_future_dataframe(periods=30)`
5. Прогноз: `forecast = model.predict(future)`
6. Візуалізація: `model.plot_components(forecast)`[21].

У дослідженні було виявлено переваги Prophet над ARIMA та ETS у випадках з мультисезонністю, нерівномірними інтервалами та викидами[20], а також у можливості автоматичного обліку спадів активності під час свят[19].

Протилежно до ARIMA, Prophet не потребує попереднього диференціювання і автоматично виокремлює тренд та сезонність. У порівнянні з ETS він краще справляється з мультисезонними рядами та нерівномірною частотою спостережень. Хоча глибинні моделі (LSTM) можуть досягати вищої точності при достатньому обсязі даних, Prophet забезпечує прозорі параметри та швидке налаштування без складних скриптів.

Особливість Prophet — аналіз точок переходу (changepoints): за замовчуванням 25 точок на перших 80 % даних з  $L_1$ -регуляцією, що дозволяє балансувати між надмірним згладжуванням і перенавчанням[19].

**Обмеженням** архітектури Prophet є те, що нормально розподілені помилки та адитивність складових не завжди відповідають реальним даним з сильними нелінійностями чи мультиплікативними ефектами. Складні події можуть вимагати ручного додавання точок змін[19].

Незважаючи на це, простота, гнучкість і прозорість зробили Prophet одним із найпопулярніших інструментів для прогнозування в бізнесі та науці[18].

## 2.5 Показники якості прогнозу

Вибір адекватних метрик для оцінки точності моделей має вирішальне значення, адже від точності цих показників залежить наукова цінність висновків і практична користь результатів для планування. Для прогнозування Affinity традиційно використовують такі закупівельні метрики:

- **MAE** — середня абсолютна похибка
- **MAPE** — середня абсолютна відсоткова похибка
- **RMSE** — корінь середньоквадратної похибки
- **R<sup>2</sup>** — коефіцієнт детермінації
- **MASE** — масштабована абсолютна помилка

### 2.5.1 MAE

MAE визначає середню величину абсолютних відхилень прогнозу від фактичних значень:

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (2.16)$$

Ця метрика відповідає мінімізації L<sub>1</sub>-норми і менш чутлива до великих відхилень [9]. MAE вимірюється в тих самих одиницях, що й дані, наприклад, в одиницях індексу Affinity.

### 2.5.2 MAPE

MAPE рахується у відсотках, що корисно при порівнянні різних рядів:

$$MAPE = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (2.17)$$

Ця метрика зручна для комунікації результатів [9], але вимагає, щоб всі  $y_t$ , тому для рядів з нульовими значеннями використовують корекцію.

### 2.5.3 RMSE

RMSE мінімізує  $L_2$ -норму похибок:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (2.18)$$

Великий вплив великих похибок робить RMSE доречним, коли помилки мають непропорційно високі витрати [9].

Класичний розклад:

$$RMSE^2 = Bias^2 + Var(\varepsilon) \quad (2.19)$$

### 2.5.4 Коефіцієнт детермінації ( $R^2$ )

$R^2$  показує, яка частка дисперсії пояснюється моделлю:

$$R^2 = 1 - \frac{\sum_t (y_t - \hat{y}_t)^2}{\sum_t (y_t - y^-)^2}, \quad y^- = \frac{1}{n} \sum_t y_t \quad (2.20)$$

Для часових рядів його слід інтерпретувати обережно через автокореляцію та тренд [12].

### 2.5.5 MASE

MASE порівнює MAE з «наївною» моделлю:

$$MASE = \frac{MAE}{\frac{1}{n-m} \sum_{t=m+1}^n |y_t - y_{t-m}|} \quad (2.21)$$

де  $m$  — сезонний лаг (наприклад,  $m=7$  для щоденних даних) [9]. MASE незалежна від одиниць виміру та дозволяє порівнювати ряди.

Для комплексної оцінки прогнозу варто використовувати декілька метрик одночасно:

- MAE і RMSE — абсолютна точність
- MAPE — відносна оцінка
- $R^2$  — пояснювальна здатність
- MASE — порівняння з наївною моделлю

Окремо оцінюють результати на навчальній та тестовій вибірках (зазвичай 70/30 або 80/20) і аналізують метрики на різних горизонтах  $h$  ( $MAE_h$ ,  $RMSE_h$ ) для розуміння динаміки точності на коротких і довгих інтервалах.

## 2.6 Процес охайного (tidy) прогнозування (tidy forecasting workflow)

Поняття *tidy forecasting workflow* уперше було системно сформульовано Робертом Гайндманом та його колегами у третьому виданні підручника **Forecasting: Principles and Practice** [10]. Автори перенесли ідеї Хедлі Вікема щодо *tidy data* [23] у сферу часових рядів, запропонували єдину граматику роботи з даними, моделями та прогнозами на основі екосистеми **tidyverts** [22]. Такий підхід дав змогу описати весь життєвий цикл прогнозування — від імпорту сирих даних до інтеграції готових прогнозів — у вигляді послідовності однорідних, «чистих» операцій, кожна з яких повертає зрозумілу *tidy-таблицю* з одним спостереженням у рядку та одним атрибутом у стовпці.

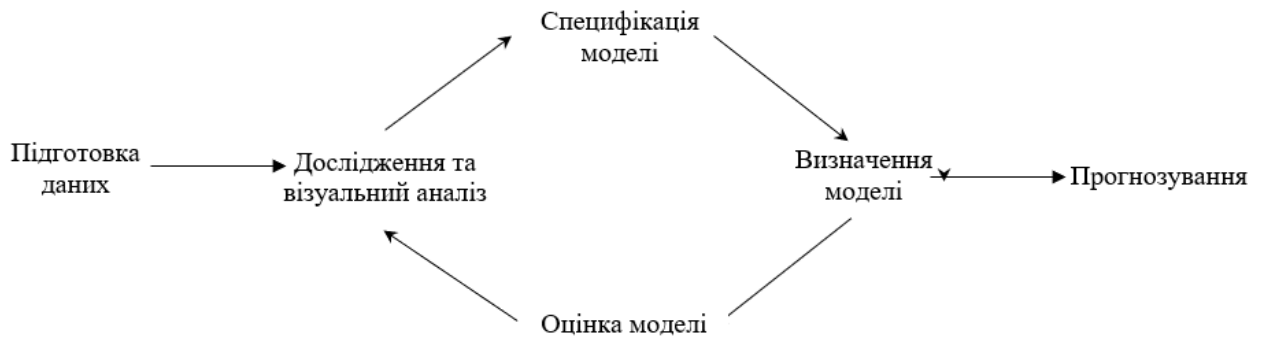


Рисунок 2.1— Tidy forecasting workflow

У Python-середовищі, попри відсутність прямої аналогії до *tsibble*, концепцію *tidy* легко реалізувати з використанням *pandas*, *statsmodels*, *scikit-learn* і сучасних пайплайнів [14]; [16]. Отже, подальший опис спиратиметься на *tidy*-ідеологію, але проходитьиме крізь призму Python-екосистеми, що забезпечує інтеграцію з інструментами машинного навчання.

### 2.6.1 Загальні принципи *tidy*-підходу

1. **Єдиний формат даних:** часові ряди подаються у довгому вигляді (*long format*), де ключем є часовий індекс, а значення — вимірюваний показник.
2. **Функціональний ланцюжок:** кожний крок обробки або моделювання — це функція, що приймає «чисту» таблицю та повертає таку саму структуру.
3. **Відокремлення даних і моделей:** моделі зберігаються у стовпці типу *model* поряд із метаданими, що полегшує об'єднання та порівняння.
4. **Явні мета-дані:** інформація про частоту, часову зону, відсутні значення або трансформації зберігається як атрибути часових рамок.
5. **Кодування в стилі pipeline:** ланцюжки `df.pipe()` або `sklearn.pipeline.Pipeline` документують процес та роблять його відтворюваним.

- б. «**Все є таблицею**»: навіть прогнози повертаються у tidy-форматі, тобто кожен горизонт у своєму рядку, а стовпці — це середнє, довірчі інтервали, моделі, тощо.

### 2.6.2 Етап 1. Підготовка та структуризація даних

Перший крок — перевести різномірні джерела (SQL-запити, CSV, API) у єдиний датафрейм зі змінною *timestamp* як індексом. Для цього варто:

- конвертувати часову зону до *Europe/Kyiv* і задати потрібну частоту (*asfreq*) для уніфікації;
- перевірити пропуски (*isna()*) та або додати їх (лінійно, *locf*, регресією), або помітити їх прапорцем;

У контексті Python рекомендується створити окремий набір тестових функцій (*pytest*) для перевірки коректності часових відміток і цілісності сумарних показників, щоб уникнути проблем у пайплайні на пізніших етапах.

### 2.6.3 Етап 2. Дослідження та візуальний аналіз

Візуалізація — ключ до розуміння сезонностей, трендів і аномалій. У tidy-підході функції візуалізації (*plotly.express*, *matplotlib*) сприймають ту ж таблицю, тож:

- спершу будують лінійні графіки кожного *series\_id*;
- далі — фасетні діаграми за днями тижня (*weekday seasonality*), місяцями тощо;
- порівнюють закупівельні метрики (середнє, дисперсія, автокореляція) між аудиторіями.

Якщо довгі «хвости» чи пропуски спотворюють масштаби, застосовують функцію `transform("log1p")`, а після аналізу фільтр `groupby().agg()` повертає tidy-дані для подальших етапів.

#### 2.6.4 Етап 3. Попередня обробка та інженерія ознак

Перед створенням самої моделі дані часто потрібно трансформувати:

- **Календарні ознаки:** `dayofweek`, `weekofyear`, `is_holiday` — зберігаються у стовпцях для можливих регресорів;
- **Сплайн-тенденції:** ортогональні В-сплайни чи натуральні кубічні сплайни для гнучкого тренду;
- **Вох-Соx** або `np.log1p` для стабілізації дисперсії;
- **Диференціювання** (перший/сезонний порядок) для усунення нестационарності, проте зберігається варіант «`inverse_transforms`» у метаданих, аби повернути прогноз у вихідний масштаб.

Важливо, що `sklearn.compose.ColumnTransformer` дозволяє «загорнути» всі перетворення в об'єкт, а `Pipeline` записує порядок кроків. Це втілює правило *tidy-workflow*: *дані залишаються в єдиному форматі до й після трансформацій*.

#### 2.6.5 Етап 4. Визначення та навчання моделей

У *tidy*-підході кожна модель — це об'єкт, що лежить у стовпці `model` (або окремому списку) й «знає» методи `fit()` та `predict()`.

Усіх їх можна об'єднати у список і подати широким форматом (`pd.concat`), після чого легко порівнювати MAPE, RMSE та MAE. Ключове: **модель не змінює формат даних**, а результати повертаються тією ж *tidy*-таблицею.

#### 2.6.6 Етап 5. Оцінювання точності та порівняння моделей

Правильна оцінка моделі — це запорука достовірності прогнозу. У *tidy-framework* використовується *rolling-origin* або *sliding-window* перехресне оцінювання, коли:

1. задається початкове тренувальне вікно (*initial*), крок (*step*) й горизонт (*h*);

2. генерується багато підмножин даних (`sklearn.model_selection.TimeSeriesSplit`);
3. для кожної підмножини зберігається значення фактичних і прогнозованих значень  $y$ ;
4. обчислюється MAPE, RMSE, MAE тощо через `sklearn.metrics`.

Результатом є нова `tidy`-таблиця, де кожен рядок — це пара *модель–фолд*. Така структура дає змогу будувати `boxplot` помилок і одразу бачити, чи ARIMA суттєво поступається градієнтному бустингу.

### 2.6.7 Етап 6. Перенавчання, прогноз та інтеграція у бізнес-процеси

Після вибору найкращої моделі та перенавчання на повному тренувальному наборі забезпечується її максимальне використання:

- прогноз вивантажується у `DataFrame` з колонами `timestamp`, `point_forecast`, `lo80`, `hi80`, `lo95`, `hi95`, `model_id`;
- результат пишеться у базу (`PostgreSQL`, `BigQuery`) або передається в API для медіапланера;
- усі кроки збираються у `makefile` чи `prefect.flow`, що гарантує відтворюваність і версіювання;
- `Dash/Streamlit`-дашборд споживає `tidy`-таблицю й відображає сценарії з довірчими інтервалами.

Таким чином, принцип «все є таблицею» завершує коло: від сирих спостережень через моделі до прогнозу — кожен крок зберігається у єдиному форматі.

## 2.7 Висновки до розділу

У другому розділі було детально розглянуто математичні моделі та методологію прогнозування часових рядів, що використовуються для

передбачення показників теледивлення. Зокрема, проведено огляд та аналіз ключових підходів: ARIMA, ETS, LSTM та Prophet.

Модель ARIMA (AutoRegressive Integrated Moving Average) було описано як потужний класичний метод прогнозування, що дозволяє ефективно працювати зі стаціонарними та нестаціонарними часовими рядами шляхом їх диференціювання. Вона забезпечує просту інтерпретацію та прозорий алгоритм підбору параметрів за методологією Бокса–Дженкінса.

Метод експоненційного згладжування (ETS) було охарактеризовано як ефективну модель для часових рядів з вираженою сезонністю та трендом, що не вимагає додаткових перетворень ряду до стаціонарного вигляду. ETS надає високий ступінь автоматизації процесу моделювання та є зручним інструментом для швидкого та зрозумілого прогнозування.

У рамках роботи також було описано нейромережеву модель LSTM, яка належить до глибинного навчання та здатна враховувати складні нелінійні залежності у даних завдяки своїй архітектурі, що містить спеціальні механізми пам'яті та управління інформацією. Ця модель демонструє високу точність у складних ситуаціях, проте потребує великого обсягу даних та ретельного налаштування.

Окремо було розглянуто модель Prophet, яка є сучасним інструментом прогнозування, заснованим на адитивній моделі тренду та сезонності з можливістю врахування календарних подій. Prophet характеризується простотою використання, стійкістю до аномальних спостережень та можливістю швидко адаптуватися до змін трендів та сезонності.

Особливу увагу було приділено концепції tidy forecasting workflow — охайного процесу прогнозування, що дозволяє систематизувати етапи аналізу та моделювання даних. Було показано, що такий підхід сприяє ефективності та прозорості прогнозування, покращує якість отриманих результатів завдяки чіткій структуризації даних, візуальному аналізу, інженерії ознак,

обґрунтованому вибору моделей, а також їх оцінюванню та інтеграції в практичні бізнес-процеси.

Таким чином, у розділі було комплексно охарактеризовано основні моделі та методологічні засади, які є фундаментальними для подальшої розробки ефективних прогнозних рішень та інформаційних технологій у сфері прогнозування показників теледивлення.

## РОЗДІЛ 3. РОЗРОБКА МОДЕЛЕЙ ПРОГНОЗУВАННЯ ПОКАЗНИКА AFFINITY

### 3.1 Отримання даних теледивлення

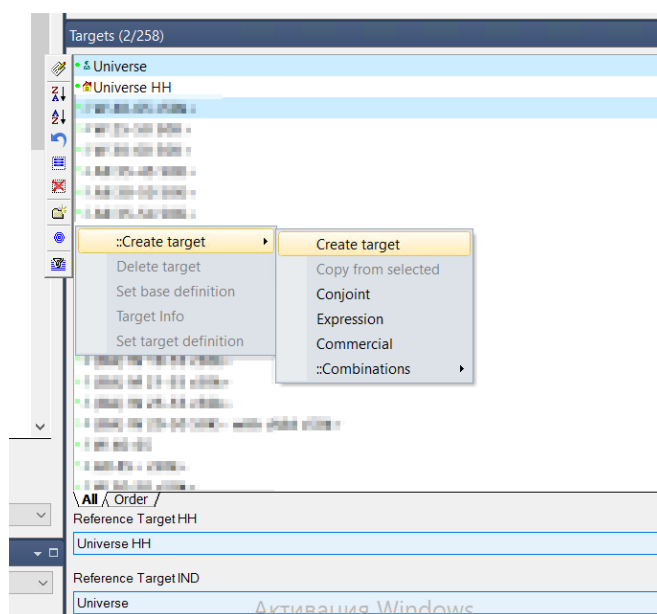
Отримання даних теледивлення у Yumianalytics відбувається у два ключові етапи: спочатку створюємо аудиторії, а потім генеруємо звіт Report Advertising з потрібними метриками.

#### 3.1.1 Створення аудиторії

Для кожної аудиторії (цільової або закупівельної) потрібно визначити набір характеристик, за якими система відбиратиме домогосподарства чи індивідів:

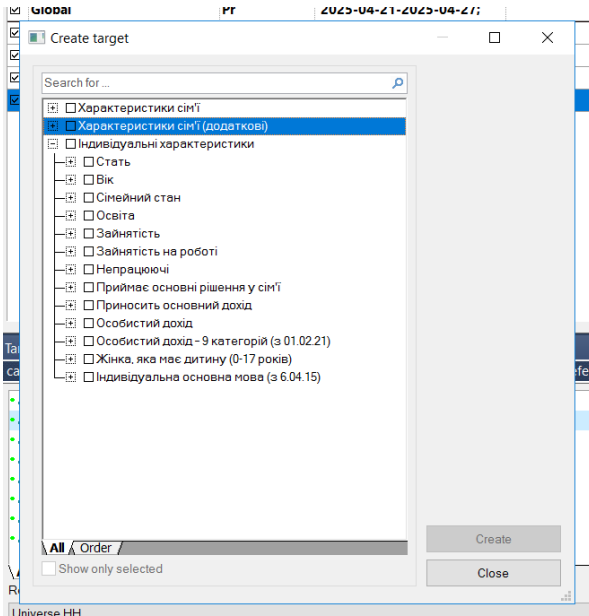
- **Цільові аудиторії** формується на основі характеристик, які має певна аудиторія, на яку спрямована рекламна кампанія.
- **Закупівельна аудиторія** будується відповідно до тих описів, які передали телеканали, та на які вони орієнтуються при розміщенні рекламних блоків.

У Yumianalytics це реалізовано через інтерфейс Targets → New Targets, як зображено на рис 3.1.



### Рисунок 3.1— Процес створення нової аудиторії

У параметрах ми можемо задати всі необхідні характеристики домогосподарства чи окремого індивіду, а також інші додаткові характеристики



### Рисунок 3.2— Приклад можливостей для обрання параметрів аудиторії

#### 3.1.2 Створення звіту (Report Advertising)

Коли маємо обрану одну чи декілька аудиторій, створюємо звіт. Цей етап також складається з декількох підетапів

##### 3.1.2.1 Задання шаблону звіту (Layout Pivot)

В Yumianalytics передбачено кілька типів звіту, в залежності від того, що саме потрібно аналізувати. Нам потрібен саме Advertising Report. В інтерфейсі програми обираємо його, а далі формуємо шаблон звіту (Layout Pivot) — зведену таблицю, що визначає структуру даних, яка зображена на рис 3.3:

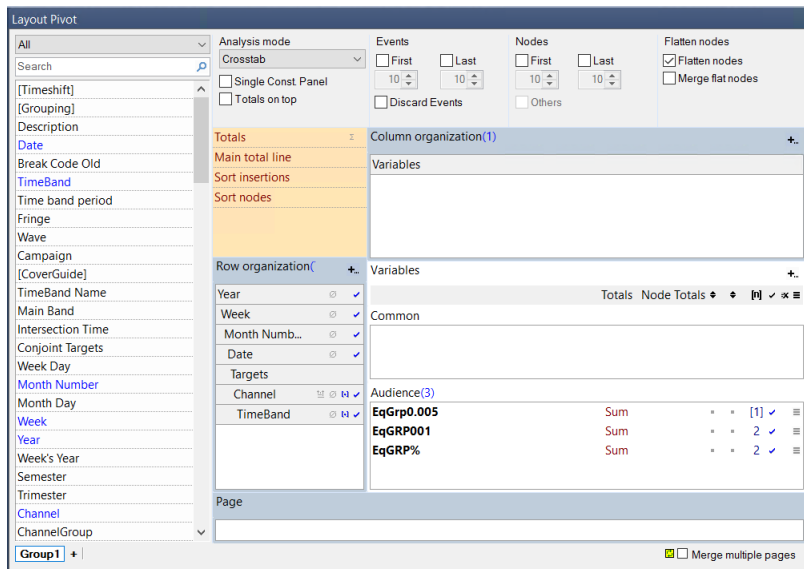


Рисунок 3.3— Вигляд шаблону звіту (Layout Pivot)

### 1. Рядки (Rows):

- Year
- Week
- Month Number
- Date
- Targets
- Channel
- TimeBand

### 2. Поля значень (Values):

- EqGRP%
- EqGrp0.005
- EqGRP001

**TimeBand** — це добові часові смуги, які допомагають аналізувати пікові періоди перегляду:

- *Early Morning* (04:00–08:59)
- *Daytime* (09:00–16:59)
- *Prime Time* (17:00–23:59)

Завдяки розподілу за TimeBand можна виявити, у які години аудиторія найбільш активна, й оптимізувати медіаплан під ці інтервали.

Параметри EqGrp0.005 і EqGRP001 використовуються для того, щоб якщо певний ролик не набрав жодного значення EqGRP, тобто його ніхто не побачив з домогосподарств, які беруть участь у панельному дослідженні (наприклад, о першій годині ночі за місцевим часом на телеканалі з невеликою аудиторією), у значенні EqGRP не вказувати 0, що було б помилкою, а певним орієнтовним значенням. У залежності від каналу, використовуються значення EqGrp0.005 або EqGRP001. Детальніше побачимо наочно в наступному підрозділі.

### 3.1.2.2 Параметри звіту

У формі створення звіту ми маємо також задати різноманітні фільтри й додаткові умови, що вплинуть на кінцевий файл, такі як дані яких каналів отримувати, які аудиторії (може бути не одна), часові проміжки, особливості аудиторій і так далі

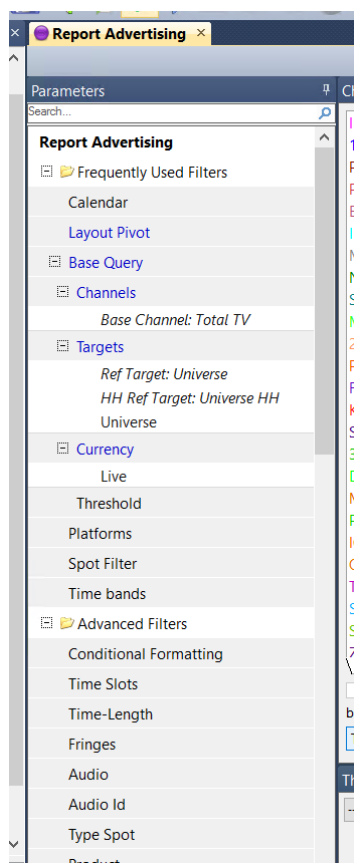


Рисунок 3.4— Можливості фільтрування звіту

Після підтвердження всіх параметрів система згенерує Excel-файл відповідно до структури, яка вказана у шаблоні звіту.

|    | A    | B    | C            | D    | E          | F         | G        | H          | I        | J      | K      |
|----|------|------|--------------|------|------------|-----------|----------|------------|----------|--------|--------|
|    | Year | Week | Month Number | Date | Targets    | Channel   | Timeband | EqGRP0 005 | EqGRP001 | EqGRP% |        |
| 2  | 1    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,0      | 0,03   | 0,00   |
| 3  | 2    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,1      | 0,10   | 0,09   |
| 4  | 3    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,4      | 0,42   | 0,41   |
| 5  | 4    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 4,4      | 4,43   | 4,32   |
| 6  | 5    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 22,9     | 23,77  | 22,05  |
| 7  | 6    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,2      | 0,25   | 0,00   |
| 8  | 7    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 4,0      | 5,22   | 2,82   |
| 9  | 8    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,2      | 0,33   | 0,00   |
| 10 | 9    | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 70,9     | 71,10  | 70,64  |
| 11 | 10   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 37,7     | 37,69  | 37,69  |
| 12 | 11   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 244,9    | 244,88 | 244,88 |
| 13 | 12   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 42,4     | 42,41  | 42,39  |
| 14 | 13   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 375,3    | 375,59 | 375,05 |
| 15 | 14   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 9,3      | 9,45   | 9,23   |
| 16 | 15   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 26,4     | 27,21  | 25,61  |
| 17 | 16   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,1      | 0,16   | 0,00   |
| 18 | 17   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 1,1      | 2,27   | 0,00   |
| 19 | 18   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,2      | 0,31   | 0,00   |
| 20 | 19   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 1,4      | 2,74   | 0,00   |
| 21 | 20   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,1      | 0,27   | 0,00   |
| 22 | 21   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 1,1      | 2,11   | 0,00   |
| 23 | 22   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,1      | 0,23   | 0,00   |
| 24 | 23   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 1,2      | 2,19   | 0,24   |
| 25 | 24   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,1      | 0,24   | 0,00   |
| 26 | 25   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 1,0      | 1,97   | 0,00   |
| 27 | 26   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,1      | 0,27   | 0,00   |
| 28 | 27   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 1,1      | 2,28   | 0,00   |
| 29 | 28   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 0,2      | 0,31   | 0,00   |
| 30 | 29   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 14,1     | 14,99  | 13,29  |
| 31 | 30   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 12,0     | 12,04  | 12,04  |
| 32 | 31   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 35,3     | 35,80  | 34,90  |
| 33 | 32   | 2025 | 17           | 4    | 26 04 2025 | Семантика | Search   | 17:30      | 64,1     | 64,11  | 64,11  |

Рисунок 3.5— Вигляд готового звіту Excel

Таким чином, ми завантажуюмо звіт з даними рейтингів за період з 1 січня 2023 року по 31 грудня 2024 року для цільової аудиторії (далі будемо використовувати назву Цільова\_Аудиторія) та для закупівельної аудиторії (далі будемо використовувати назву Закупівельна\_Аудиторія). У ході роботи в програмі Yumianalytics виявилось, що дані, розбиті на файли з щоденними даними, вивантажуються значно швидше, ніж дані, розбиті потижнево чи помісячно. Тому отримуємо дані у вигляді звітів з щоденними даними, які далі оброблюємо.

### 3.2 Обробка даних

Після отримання даних про рейтинги цільових та закупівельних аудиторій з Yumianalytics, ми переходимо до процесу підготовки даних до моделювання. На цьому етапі ми не виконуємо жодних фільтрацій чи об'єднань, а лише готуємо актуальний масив даних для подальшої роботи.

На рис 3.6 зображений процес обробки даних.

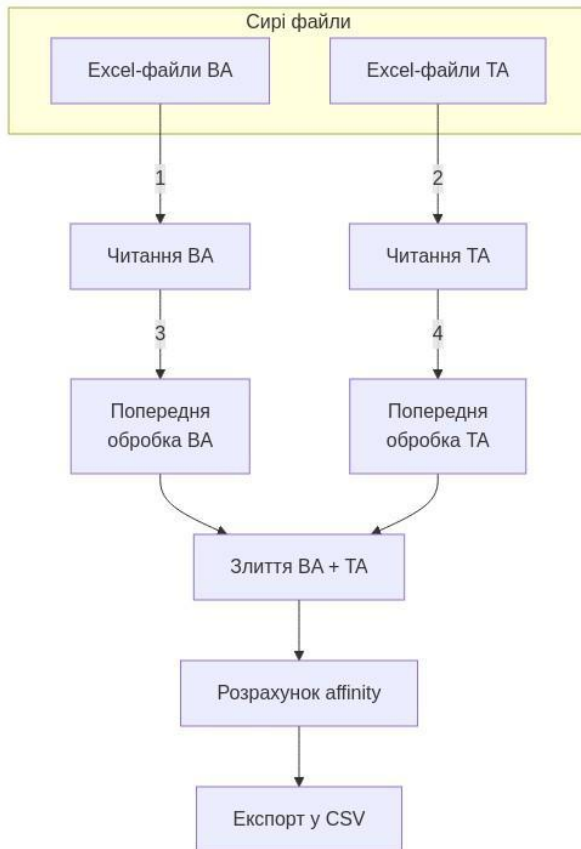


Рисунок 3.6— ETL схема обробки даних

Розглянемо його детальніше

### 3.2.1 Завантаження вхідних даних

На цьому кроці ми об'єднуємо звіти з щоденними даними, які отримали у попередньому підрозділі. Для зручності аналізу ми використовуємо структуру типу DataFrame з бібліотеки pandas, що забезпечує індексацію та роботу з колонками. Так ми отримуємо дві таблиці даних, з даними рейтингів для цільової та закупівельної аудиторії

### 3.2.2 Обробка даних

Об'єднані файли ми доповнюємо даними зі словника, який зберігається окремим Excel файлом. З нього ми отримуємо офіційні назви каналів, оскільки в Yutianalytics збережені лише назви англійською та назву групи, в яку входить обраний телеканал, що потрібно для зручності користувачів. У моделюванні ці дані використовуватись не будуть.

Для таблиці з даними закупівельних аудиторій ми додатково створюємо нову колонку eqGRP\_round\_BA. Вона містить значення рейтингу, вибрані з двох альтернативних колонок (EqGrp0.005 або EqGRP001) залежно від значення параметра, який вказаний у словнику. Якщо його значення становить 0.005, ми використовуємо колонку EqGrp0.005, якщо 0.01 — колонку EqGRP001. Таким чином забезпечується правильне округлення рейтингів згідно з методологією дослідження.

### 3.2.3 Об'єднання даних цільової та закупівельної аудиторій

Після створення нової колонки ми перейменовуємо поле Targets на Target\_BA, щоб чітко розмежувати цільову і закупівельні аудиторії. На завершення об'єднуємо обидві таблиці, у фінальну структуру final\_df, яка включає дані за всіма потрібними параметрами — рік, тиждень, дата, канал, слот, рейтинг, аудиторія. Її зображення наведено на рис 3.7.

|          | Year | Week | Month | Date       | Targets   | Channel   | TimeBand  | EqGrp0.005 | EqGRP001 | EqGRP%  | flow      | SH        | eqGRP_round_BA | Target_BA |
|----------|------|------|-------|------------|-----------|-----------|-----------|------------|----------|---------|-----------|-----------|----------------|-----------|
| 64606200 | 2025 | 9    | 3     | 2025-03-01 | [Complex] | [Complex] | [Complex] | 0.0000     | 0.0100   | 0.0000  | [Complex] | [Complex] | NaN            | [Complex] |
| 64606201 | 2025 | 9    | 3     | 2025-03-01 | [Complex] | [Complex] | [Complex] | 0.0000     | 0.0100   | 0.0000  | [Complex] | [Complex] | NaN            | [Complex] |
| 64606202 | 2025 | 9    | 3     | 2025-03-01 | [Complex] | [Complex] | [Complex] | 0.0000     | 0.0100   | 0.0000  | [Complex] | [Complex] | NaN            | [Complex] |
| 64606203 | 2025 | 9    | 3     | 2025-03-01 | [Complex] | [Complex] | [Complex] | 0.0000     | 0.0100   | 0.0000  | [Complex] | [Complex] | NaN            | [Complex] |
| 64606204 | 2025 | 9    | 3     | 2025-03-01 | [Complex] | [Complex] | [Complex] | 0.0000     | 0.0100   | 0.0000  | [Complex] | [Complex] | NaN            | [Complex] |
| ...      | ...  | ...  | ...   | ...        | ...       | ...       | ...       | ...        | ...      | ...     | ...       | ...       | ...            | ...       |
| 65729431 | 2025 | 11   | 3     | 2025-03-16 | [Complex] | [Complex] | [Complex] | 93.0000    | 93.1900  | 92.8200 | [Complex] | [Complex] | 51.6000        | [Complex] |
| 65729432 | 2025 | 11   | 3     | 2025-03-16 | [Complex] | [Complex] | [Complex] | 93.0000    | 93.1900  | 92.8200 | [Complex] | [Complex] | 66.6000        | [Complex] |
| 65729433 | 2025 | 11   | 3     | 2025-03-16 | [Complex] | [Complex] | [Complex] | 93.0000    | 93.1900  | 92.8200 | [Complex] | [Complex] | 62.6000        | [Complex] |
| 65729434 | 2025 | 11   | 3     | 2025-03-16 | [Complex] | [Complex] | [Complex] | 93.0000    | 93.1900  | 92.8200 | [Complex] | [Complex] | 35.4000        | [Complex] |
| 65729435 | 2025 | 11   | 3     | 2025-03-16 | [Complex] | [Complex] | [Complex] | 93.0000    | 93.1900  | 92.8200 | [Complex] | [Complex] | 53.1000        | [Complex] |

Рисунок 3.7— Вихідний масив даних

Для забезпечення коректної роботи на наступних етапах, ми очищуємо фінальний датафрейм: видаляємо службову першу колонку (що може з'явитись під час злиття), а також замінюємо всі нескінченні значення ( $inf$ ,  $-inf$ ) на нулі. Це дозволяє уникнути збоїв у подальшій обробці даних.

### 3.3 Візуалізація часових рядів

Для візуалізації даних, нам потрібно привести масив даних до tidy (охайного) вигляду. Для цього ми приберемо зайві колонки, та колонку “Date” перетворимо в індекс. В результаті, отримуємо масив даних як на малюнку.

| Date       | EqGRP_TA | EqGRP_BA |
|------------|----------|----------|
| 2023-01-01 | 450.06   | 649.66   |
| 2023-01-02 | 459.65   | 556.36   |
| 2023-01-03 | 275.82   | 362.91   |
| 2023-01-04 | 458.18   | 512.93   |
| 2023-01-05 | 445.14   | 477.92   |
| 2023-01-06 | 868.34   | 827.2    |
| 2023-01-07 | 500.89   | 657.28   |
| 2023-01-08 | 499.61   | 451.6    |
| 2023-01-09 | 348.28   | 458.07   |
| 2023-01-10 | 490.58   | 507.63   |

Рисунок 3.8 — Вихідний масив даних

Також створимо окремий набір даних, в якому також додамо колонку з обрахованим значенням Affinity.

| Date       | EqGRP_TA | EqGRP_BA | affinity           |
|------------|----------|----------|--------------------|
| 2023-01-01 | 450.06   | 649.66   | 0.692762368007881  |
| 2023-01-02 | 459.65   | 556.36   | 0.8261737004817025 |
| 2023-01-03 | 275.82   | 362.91   | 0.7600231462346035 |
| 2023-01-04 | 458.18   | 512.93   | 0.8932602889283139 |
| 2023-01-05 | 445.14   | 477.92   | 0.931411114830934  |
| 2023-01-06 | 868.34   | 827.2    | 1.0497340425531914 |
| 2023-01-07 | 500.89   | 657.28   | 0.7620648734177216 |
| 2023-01-08 | 499.61   | 451.6    | 1.1063108945969884 |
| 2023-01-09 | 348.28   | 458.07   | 0.7603204750365664 |
| 2023-01-10 | 490.58   | 507.63   | 0.9664125445698638 |

Рисунок 3.9 — Вихідний масив даних з обрахованим Affinity

Побудуємо зображення часових рядів:

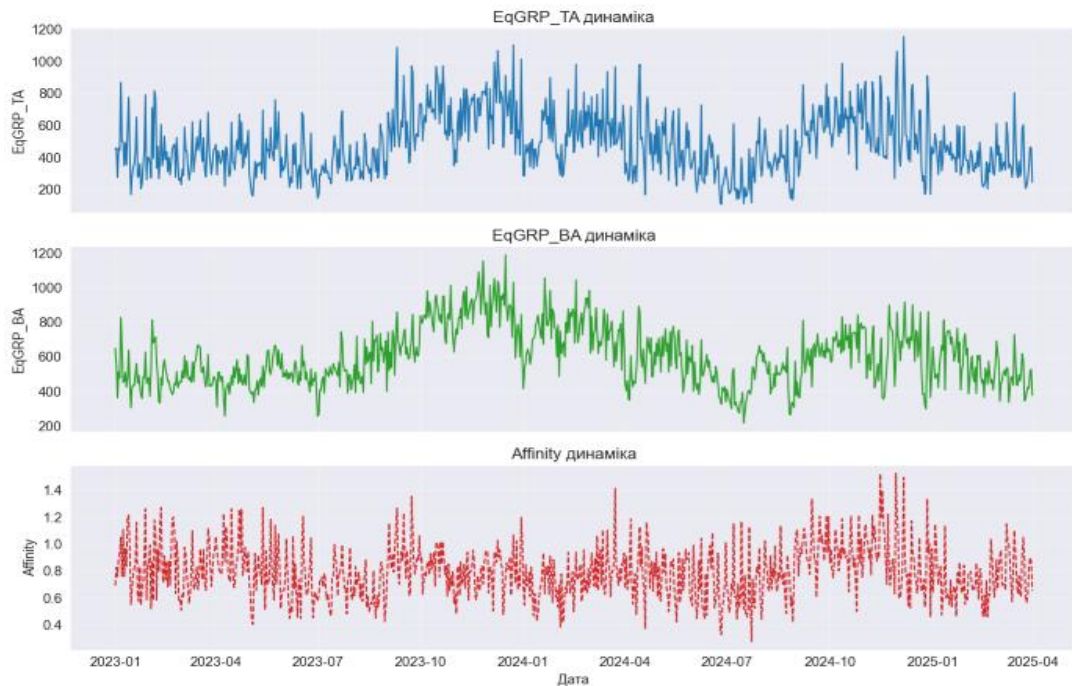


Рисунок 3.10— Часові ряди обрахованих EqGRP та Affinity

Розглянемо уважно отримані часові ряди та опишемо їх. Криві еквівалентних GRP для цільової та закупівельної аудиторій, попри належність до одного ринку й спільну календарну сітку, демонструють відчутно різну поведінку. Динаміка цільової аудиторії характеризується високочастотними імпульсами з різкою зміною амплітуди, що свідчить про сильний вплив короткострокових тригерів: релізів прем'єр, ситуативних рекламних кампаній, подій одноденного формату. Натомість закупівельна аудиторія рухається хвилеподібно: упродовж 2023 року спостерігається поступове зростання до локального максимуму на початку 2024-го, далі—помірне зниження, і все це відбувається з відносно стабільною тижневою пульсацією. Накладання таких різнорідних траєкторій у показнику Affinity породжує ефект «рухомого масштабу»: у фазах, коли значення рейтингу для закупівельної аудиторії високий, відношення стискається майже до плоскої лінії, а коли просідає, навіть

помірні сплески значень рейтингу цільової аудиторії багаторазово роздмухують амплітуду. Унаслідок цього частка набуває явної гетероскедастичності й втрачає однорідну сезонну структуру, що принципово ускладнює побудову для неї самостійної стаціонарної моделі. Тому для подальшого обрахування ми будемо будувати окремо моделі прогнозу для цільової та закупівельної аудиторії, з яких отримуватимемо прогноз Affinity. Таким чином, задача прогнозування Affinity зводиться до задачі прогнозування рейтингу теледивлення цільової аудиторії

Виконаємо для часових рядів рейтингів цільових аудиторій STL-декомпозицію.

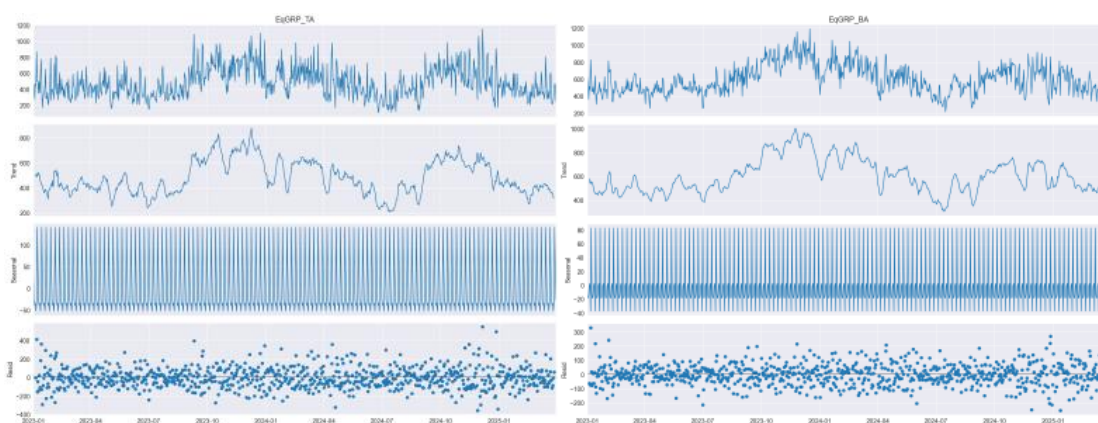


Рисунок 3.11— STL-декомпозиція даних часових рядів

На обох рисунках подано результати STL-декомпозиції щоденних рядів EqGRP для цільової (ТА) та закупівельної (ВА) аудиторій. Почнемо з компоненти тренду. У випадку ТА ми спостерігаємо відносно рівний старт на початку 2023 року із значеннями близько 400 пунктів, після чого у другому та третьому кварталах 2023 року рівень поступово знижується, досягає локального мінімуму у липні-серпні, а надалі швидко зростає й підтримується в діапазоні 600–700 пунктів. На початку 2024 року тренд знову знижується, проте протягом літа-осені відновлює схильність до підйому і завершує період на рівні, що все ж нижчий, ніж максимум попередньої хвилі. Для ВА загальна форма хвилі схожа, однак амплітуда коливань помітно більша: висхідна фаза 2023 року приводить

показник майже до 1000 пунктів, а спадання у 2024 році відбувається до вищого базового рівня. Тобто обидва ряди мають схожі риси динаміки «підйом — спад — відскок», але масштаб і швидкість реагування різняться.

Сезонні складові відтворюють чітку семиденну періодичність, що характерна для телеперегляду. Амплітуда у ТА сягає  $\pm 120$  пунктів, тоді як у ВА вона обмежується приблизно  $\pm 70$  пунктами, — ще один індикатор того, що баїнгова аудиторія менш чутлива до короткотермінових контентних імпульсів. При цьому форма хвилі практично синфазна: максимуми припадають на вихідні, мінімуми — на будні, що підтверджує спільну поведінку споживання медіаконтенту обома сегментами.

Залишкова компонента в обох випадках коливається навколо нуля й демонструє приблизно сталу дисперсію, що свідчить про адекватне виділення тренду й сезонності. У ТА ми бачимо окремі викиди до  $\pm 400$  пунктів, тоді як у ВА розкид рідко перевищує  $\pm 250$  пунктів, отже цільовий ряд волатильніший навіть після вилучення регулярних ефектів. Графічний огляд резидуалів не виявляє явної автокореляції, хоча точна оцінка цього факту потребуватиме формальних тестів (Ljung-Box, Durbin-Watson).

Сумарно декомпозиція підтверджує три ключові висновки. По-перше, трендові траєкторії рядів ТА і ВА збігаються за фазою, але суттєво відрізняються за амплітудою, що обумовлює змінну різницю між ними упродовж періоду. По-друге, тижнева сезонність є спільним драйвером обох процесів, але її сила різна, тож на рівні частки Affinity цей фактор не взаємно компенсується. По-третє, після усунення тренду й сезонності залишки набувають доволі стабільної природи, що створює передумови для побудови стаціонарних моделей.

Отже, декомпозиція продемонструвала, що обидва ряди мають синхронний, але різноамплітудний тренд, спільну тижневу сезонність із неоднаковою силою та відносно стабільні, майже некорельовані залишки. Така структура цілком відповідає припущенням класичної лінійної модельної рамки: тренд можна врахувати різницюванням, сезонність — у вигляді сезонних

різниць, а залишки дають надію на білий шум після належної трансформації. Саме тому на наступному етапі ми перейдемо до побудови прогнозів із використанням сімейства ARIMA-моделей, зокрема сезонної SARIMA, аби формально описати динаміку кожного ряду та отримати точкові й інтервальні оцінки майбутніх значень.

### 3.4 Модель ARIMA

Перш ніж налаштовувати саму ARIMA-структуру, ми маємо формально переконатися, що після можливого диференціювання ряд дійсно задовольняє вимогу стаціонарності. Тому наступним кроком ми переходимо до статистичної перевірки цієї властивості. Спершу виконуємо візуальний огляд автокореляційних та часткових автокореляційних функцій: якщо ACF демонструє повільний експоненційний спад або зріз після застосування першої різниці, це вже побічно підтверджує, що трендову компоненту ми усунули правильно. Далі застосовуємо формальний тест Дікі-Фуллера: нульова гіпотеза полягає у наявності одиничного кореня, отже р-значення нижче 0.05 дозволить нам її відхилити і прийняти стаціонарність. Для надійності доповнюємо діагностику критерієм KPSS, де, навпаки, нульова гіпотеза стверджує стаціонарність; у такому разі ми очікуємо р-значення вище 0.05. Погоджені висновки двох тестів дадуть підстави вважати, що комбіноване застосування звичайного і сезонного різницювання адекватно прибрато нестаціонарність.

У разі, якщо хоча б один з рядів залишиться нестаціонарним (наприклад, через змінну дисперсію у верхах вибірки), ми протестуємо альтернативні трансформації — логарифм чи тест Бокса-Кокса з оптимальним  $\lambda$ , — а потім повторимо перевірки, поки результати двох тестів не стануть подібними.

```

def print_stationarity_tests(series, name):
    print(f"\n{name} - Augmented Dickey-Fuller (ADF):")
    adf = adfuller(series)
    print(f"ADF Statistic: {adf[0]:.3f}, p-value: {adf[1]:.4f}")
    print("Critical Values:", adf[4])
    print(f"{name} - KPSS:")
    kpss_res = kpss(series, regression='c', nlags='auto')
    print(f"KPSS Statistic: {kpss_res[0]:.3f}, p-value: {kpss_res[1]:.4f}")
    print("Critical Values:", kpss_res[3])
    print("-"*40)

print_stationarity_tests(train['EqGRP_TA'], 'EqGRP_TA (original)')
print_stationarity_tests(train['EqGRP_BA'], 'EqGRP_BA (original)')
[18]

EqGRP_TA (original) - Augmented Dickey-Fuller (ADF):
ADF Statistic: -2.225, p-value: 0.1974
Critical Values: {'1%': -3.439593802573824, '5%': -2.865619356068967, '10%': -2.568942332870462}
EqGRP_TA (original) - KPSS:
KPSS Statistic: 0.513, p-value: 0.0387
Critical Values: {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739}
-----

EqGRP_BA (original) - Augmented Dickey-Fuller (ADF):
ADF Statistic: -2.396, p-value: 0.1429
Critical Values: {'1%': -3.439593802573824, '5%': -2.865619356068967, '10%': -2.568942332870462}
EqGRP_BA (original) - KPSS:
KPSS Statistic: 0.637, p-value: 0.0192
Critical Values: {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739}
-----

```

Рисунок 3.12 — Скрипт для обрахунку тесту Діккі-Фуллера та KPSS, і їх результати

Виходячи з отриманих результатів тестування стаціонарності для обох часових рядів, бачимо цілком типовий для медійних даних випадок. За тестом Дікі-Фуллера початкові ряди EqGRP\_TA та EqGRP\_BA (оригінальні значення) мають р-значення, що суттєво перевищують пороговий рівень 0.05: 0.1974 для ТА і 0.1429 для ВА відповідно. Це означає, що нульову гіпотезу про наявність одиничного кореня не можна відхилити, тобто обидва ряди на цій стадії не є стаціонарними, і присутній тренд або інші форми нестаціонарності. Тест KPSS, що має протилежну нульову гіпотезу (про стаціонарність), також сигналізує про наявність нестаціонарності: для ТА статистика 0.513 із р-значенням 0.0387 вже перевищує критичне значення на рівні 5 %, а для ВА ситуація ще однозначніша — статистика 0.637, р-значення 0.0192. Тобто тут уже нульова гіпотеза про стаціонарність відкидається в обох випадках. Такі результати, коли обидва

незалежних критерії “говорять в унісон” щодо нестационарності, означають, що подальше моделювання потребує відповідної трансформації даних.

```
ta_diff_seas = train['EqGRP_TA'].diff(1).diff(7).dropna()
ba_diff_seas = train['EqGRP_BA'].diff(1).diff(7).dropna()
print_stationarity_tests(ta_diff_seas, 'EqGRP_TA (diff1, diff7)')
print_stationarity_tests(ba_diff_seas, 'EqGRP_BA (diff1, diff7)')
[18]
```

```
EqGRP_TA (diff1, diff7) - Augmented Dickey-Fuller (ADF):
ADF Statistic: -9.972, p-value: 0.0000
Critical Values: {'1%': -3.4396995339981444, '5%': -2.8656659438580796, '10%': -2.5689671530263554}
EqGRP_TA (diff1, diff7) - KPSS:
KPSS Statistic: 0.073, p-value: 0.1000
Critical Values: {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739}
-----

EqGRP_BA (diff1, diff7) - Augmented Dickey-Fuller (ADF):
ADF Statistic: -9.886, p-value: 0.0000
Critical Values: {'1%': -3.4396995339981444, '5%': -2.8656659438580796, '10%': -2.5689671530263554}
EqGRP_BA (diff1, diff7) - KPSS:
KPSS Statistic: 0.029, p-value: 0.1000
Critical Values: {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739}
-----
```

Рисунок 3.13— Результати статистичних тестів для рядів після подвійного диференціювання

Для вирішення цієї задачі до обох рядів було застосовано послідовне подвійне диференціювання — спочатку звичайне ( $d=1$ ), а потім сезонне з періодом 7 днів ( $D=1$ ), що відповідає типовій тижневій сезонності теледивлення. Після цих перетворень повторний аналіз тестами ADF і KPSS дає однозначний сигнал: значення ADF стали надзвичайно низькими (статистика близько -10 для обох рядів) з р-значенням 0.0000, що вказує на впевнене відхилення гіпотези про наявність одиничного кореня — тобто стаціонарність досягнута. KPSS для трансформованих рядів, навпаки, демонструє статистики 0.073 і 0.029 відповідно, а р-значення максимальні для тесту (0.1), що дозволяє залишити гіпотезу про стаціонарність без відхилення. По суті, обидва критерії тепер "погоджуються" між собою щодо стаціонарності оброблених рядів.

Така діагностика однозначно підказує нам подальшу стратегію: моделі SARIMA потрібно будувати вже не на початкових, а на двічі диференційованих часових рядах. Саме ці трансформовані дані будуть базою для ідентифікації

оптимальних параметрів  $(p, q)$  та  $(P, Q, s)$ , що дозволить отримати найбільш коректну та прогностично ефективну модель. Такий підхід забезпечує адекватність моделі як для короткострокового прогнозу, так і для відлову потенційної сезонної динаміки, властивої ринкам медіа.

Отже, для подальшого моделювання фіксуємо порядки диференціювання  $d=1$  та  $D=1$  у сезонній SARIMA-рамці з періодом  $s=7$ . Наступним кроком переходимо до ідентифікації параметрів  $p$  і  $q$  для нелагового компоненту, а також  $P$  і  $Q$  для сезонної частини. Це можна зробити двома шляхами: або «вручну» проаналізувати ACF та PACF очищених рядів і випробувати найбільш імовірні комбінації, або скористатися автоматизованим підбором (наприклад, ``pmdarima.auto_arima``) з обмеженням  $p, q, P, Q \leq 3$ , після чого перевірити адекватність моделей за AIC/BIC та діагностикою залишків (Ljung-Box, тест на гетероскедастичність). У разі задовільних діагностичних показників саме ці SARIMA-моделі стануть основою наших точкових і ймовірнісних прогнозів EqGRP для цільової та закупівельної аудиторій, які згодом поділимо для отримання прогнозу Affinity.

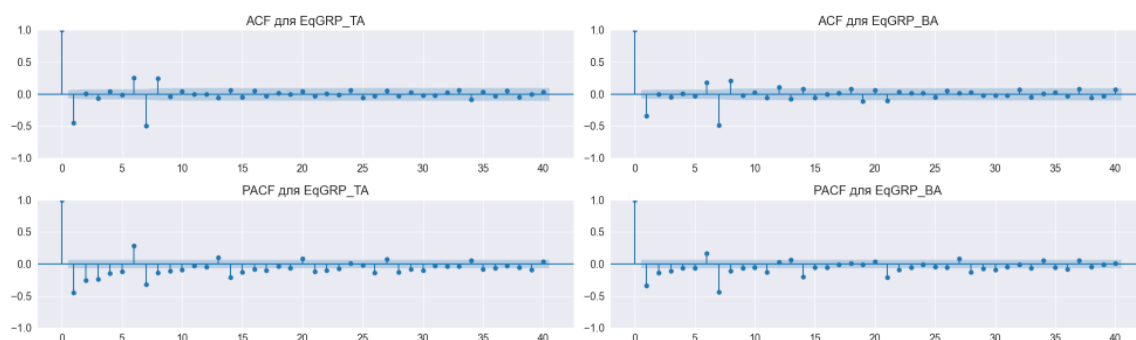


Рисунок 3.14 — Графіки автокореляційних і частково автокореляційних функцій

На обох очищених рядах — і для цільової, і для закупівельної аудиторії — подвійне різницювання (одна звичайна та одна сезонна різниця з періодом 7 днів) спрацювало майже однаково: перший стовпчик ACF залишається істотно відмінним від нуля, далі автокореляція згасає до шумового рівня, сезонних

«зубців» через кожні сім лагів не спостерігається. Часткова автокореляція показує ту саму картину — виразний пік лише на лагу 1, у наступних лагах значущих імпульсів немає. Такий силует ACF / PACF типово інтерпретується як сигнал ARMA(1, 1) у несезонній частині та відсутності потреби в окремих сезонних AR чи MA параметрах.

Отже, після аналізу часових рядів, побудуємо декілька моделей ARIMA, в яких за допомогою RMSE виберемо кращу. Для цього ми напишемо наступний код:

```
def best_auto_arima_stationary(series, name, seasonal_m=[7,14]):
    s, d, D = get_stationary_series(series, name)
    best_model = None
    best_rmse = np.inf
    best_m = None
    best_order = None
    best_seasonal_order = None
    for m in seasonal_m:
        try:
            model = pm.auto_arima(series, seasonal=True, m=m,
                                  d=d, D=D,
                                  stepwise=True, trace=False,
                                  suppress_warnings=True, error_action='ignore')
            rmse = np.sqrt(model.resid().var())
            if rmse < best_rmse:
                best_rmse = rmse
                best_model = model
                best_m = m
                best_order = model.order
                best_seasonal_order = model.seasonal_order
        except Exception as e:
            continue
    print(f"{name}: Best m={best_m}, order={best_order}, seas_order={best_seasonal_order}")
    return best_model, best_m, d, D
```

Рисунок 3.15 — Функція обрахунку кращої ARIMA-моделі

Код автоматично підбирає найкращу сезонну конфігурацію ARIMA з кількох можливих періодів сезонності, орієнтуючись на стаціонарність ряду та мінімум RMSE залишків.

Алгоритм, представлений у цьому фрагменті коду, реалізує детальний і водночас гнучкий підхід до підбору оптимальної сезонної моделі ARIMA для часових рядів, які вже було перевірено й перетворено на стаціонарні. На першому кроці для заданого ряду викликається спеціалізована функція

`get_stationary_series`, яка визначає оптимальне поєднання звичайного ( $d$ ) та сезонного ( $D$ ) порядків диференціювання. Далі модель шукає найкраще сезонне налаштування (параметр  $m$  — довжина сезонності) з-поміж кількох варіантів, що задаються у списку, наприклад, тижнева або двотижнева сезонність ( $m = 7$  або  $14$ ).

Для кожного можливого значення  $m$  запускається автоматична процедура побудови ARIMA через пакет `pmдаріма` (`pm.auto_arima`). Ця функція не тільки підбирає всі структурні параметри моделі ( $p, d, q, P, D, Q, s$ ) у відповідності до рівня стаціонарності, а й відразу враховує обраний період сезонності. Для оцінки якості кожної побудованої моделі обчислюється корінь з дисперсії залишків (RMSE), який слугує універсальним критерієм для порівняння різних конфігурацій. Серед усіх протестованих моделей з різними значеннями  $m$  вибирається та, для якої RMSE мінімальне, тобто модель, залишки якої мають найменший рівень “невловленого шуму”.

У підсумку, на виході отримуємо не лише саму модель, але й оптимальні значення сезонного періоду, порядків диференціювання та структурних параметрів ARIMA/SARIMA, що найкраще пасують до властивостей конкретного часового ряду. Це дозволяє побудувати максимально адаптивну модель, яка не лише враховує складну сезонність, а й забезпечує найкращу прогностну якість для обраної часової серії.

В результаті, ми маємо наступні коефіцієнти:

```
--- ARIMA TA ---  
EqGRP_TA: stationary after 1st diff  
EqGRP_TA: Best m=7, order=(3, 1, 1), seas_order=(2, 0, 1, 7)  
--- ARIMA BA ---  
EqGRP_BA: stationary after 1st diff  
EqGRP_BA: Best m=7, order=(2, 1, 2), seas_order=(2, 0, 2, 7)
```

Рисунок 3.16 — Результати роботи нашої отриманої функції

Результати автоматизованого підбору параметрів ARIMA свідчать про високу схожість у поведінці обох часових рядів, що є типовим для структурно пов'язаних медіапоказників. Для EqGRP\_TA модель досягає стаціонарності вже після першого диференціювання, що підтверджує незначну трендову складову та наявність сезонного патерну з періодом 7 днів. Найкраща конфігурація SARIMA для цього ряду визначена як (3, 1, 1) для несезонних параметрів та (2, 0, 1, 7) для сезонних, тобто три авторегресивні, одна диференціююча, один ковзний середній параметр плюс два сезонних авторегресивних, нуль сезонних диференціювань, один сезонний ковзний середній з тижневим періодом. Така комбінація свідчить про складну залежність поточних значень від попередніх, як на короткому відрізку, так і на рівні тижневої циклічності, що ідеально вписується у медійну логіку переглядів.

Для EqGRP\_BA ситуація схожа: також досягнуто стаціонарності після першого диференціювання, оптимальна сезонність становить 7 днів. Найкраща модель має параметри (2, 1, 2) для основної частини та (2, 0, 2, 7) для сезонної, тобто по два авторегресивних і ковзних середніх компоненти, знову ж таки з тижневим циклом. Це свідчить про ще більш виражену автокореляцію на коротких та сезонних лагах, що відображає стабільну ритміку поведінки закупівельної аудиторії.

Таким чином, обидва ряди добре піддаються моделюванню SARIMA з урахуванням тижневої сезонності, а автоматичний підбір параметрів дозволив максимально точно врахувати специфіку внутрішньої структури даних. Отримані параметри ляжуть в основу побудови фінальної прогностичної моделі та дозволять отримати якісні оцінки для подальшого аналізу та планування.

Виведемо графіки прогнозів:

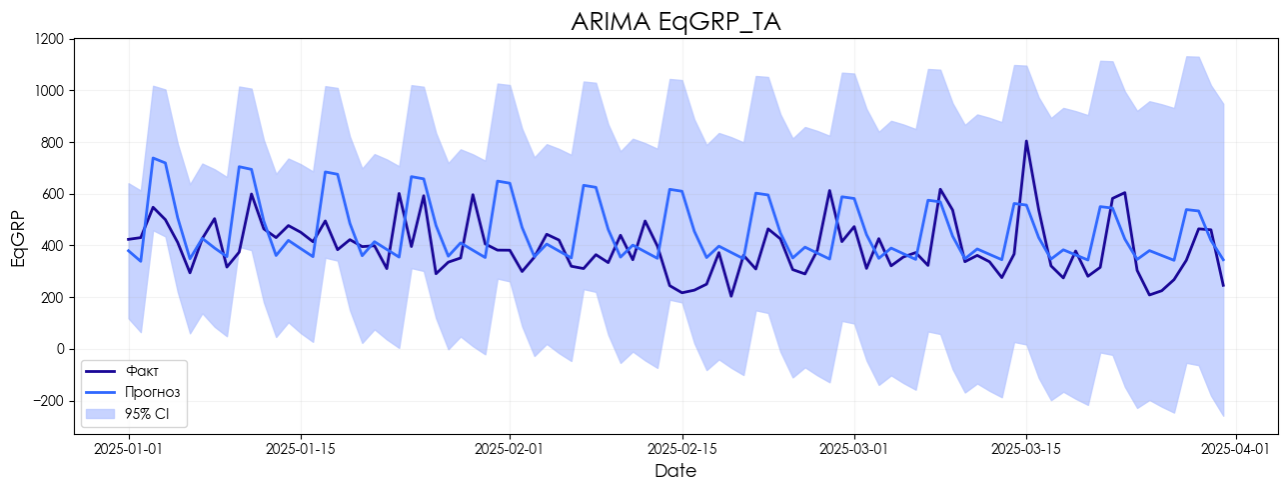


Рисунок 3.17 — Графік прогнозування ряду EeqGRP для цільової аудиторії

На першому графіку зображено результати прогнозування ряду EeqGRP\_TA за допомогою ARIMA-моделі, що враховує тижневу сезонність. Темно-синя лінія відображає реальні значення (факт), а більш світла — прогнозні значення моделі. Навколо прогнозної лінії нанесено заповнений світло-блакитний коридор, який ілюструє 95% довірчий інтервал (confidence interval) для майбутніх значень EeqGRP. На візуалізації чітко простежується ритміка змін: і реальний ряд, і прогноз підкоряються тижневим коливанням — цей патерн успішно вловлений моделлю. При цьому частина піків і спадів, притаманних реальній динаміці, також відтворюються у прогнозі, хоча локальні відхилення зберігаються — це очікувано для медіарядів з підвищеною варіативністю. Інтервали невизначеності мають хвилеподібний характер, що відображає сезонну невизначеність та поступове розширення прогнозної “довіри” із віддаленням у майбутнє.

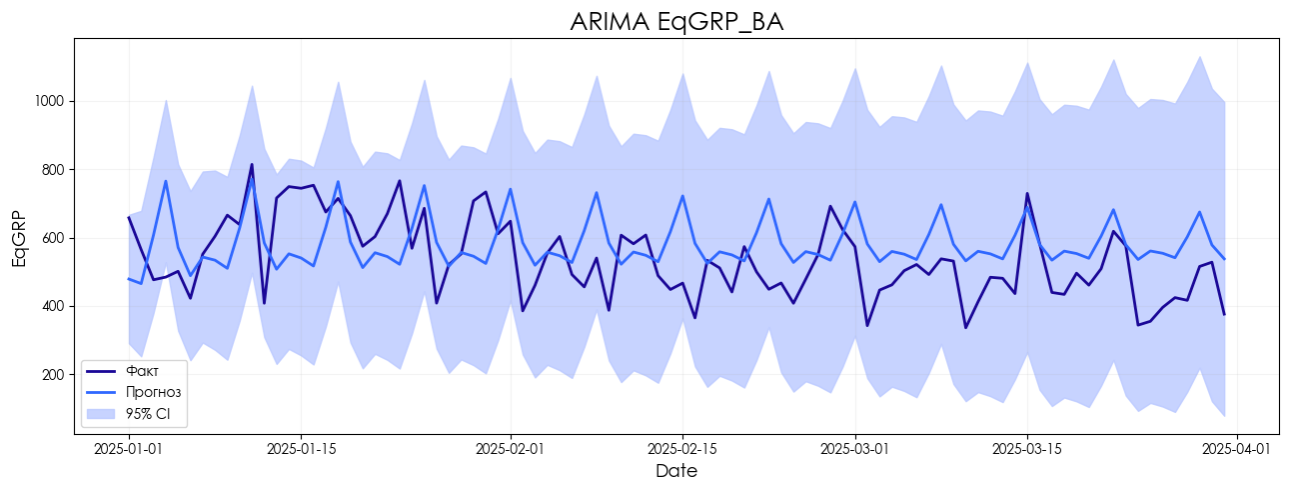


Рисунок 3.18 — Графік прогнозування ряду EeqGRP для закупівельної аудиторії

Наш графік аналогічно ілюструє результати і для закупівельної аудиторії (EeqGRP\_BA). Спостерігається подібна картина: фактичні дані та прогнозна лінія ідуть майже паралельно, зберігаючи основну сезонну хвилю. Прогноз моделі також добре відображає середній рівень і тижневу структуру, однак, як і в попередньому випадку, помітні епізодичні розбіжності між реальними і прогнозованими піками — особливо там, де реальні значення мають різкі сплески. Діапазон довірчого інтервалу досить широкий, що типово для коротких, “шумних” часових рядів у медіасфері, й свідчить про високий рівень варіативності, яку модель враховує у власній невизначеності.

У цілому, обидві візуалізації демонструють, що обрані ARIMA/SARIMA-моделі успішно захоплюють ключові динаміки реальних даних, адекватно прогнозують тижневі патерни і, разом із тим, чесно сигналізують про невизначеність при віддаленні у майбутнє — що є ознакою грамотної статистичної постановки моделі та обґрунтованих результатів для застосування у медіаплануванні.

### 3.5 Модель ETS

До побудови моделі ETS ми підійдемо схожим шляхом та створимо код, який зможе автоматично обрати кращу з моделей.

```
def fit_ets(series, seasonal_periods):
    best_model, best_type, best_rmse, best_preds = None, None, np.inf, None
    for t in ['add', 'mul']:
        for s in ['add', 'mul']:
            try:
                model = ExponentialSmoothing(series, trend=t, seasonal=s, seasonal_periods=seasonal_periods)
                fit = model.fit(optimized=True)
                preds = fit.forecast(len(test))
                rmse = np.sqrt(((preds.values - test[series.name])**2).mean())
                if rmse < best_rmse:
                    best_rmse, best_model, best_type, best_preds = rmse, fit, (t, s), preds
            except:
                continue
    print(f"ETS for {series.name}: {best_type}, RMSE={best_rmse:.2f}")
    return best_model, best_preds, best_type

ets_TA, pred_TA_ets, ets_type_TA = fit_ets(train['EqGRP_TA'], m_TA)
ets_BA, pred_BA_ets, ets_type_BA = fit_ets(train['EqGRP_BA'], m_BA)
```

Рисунок 3.19 — Функція для обрахунку кращої ETS-моделі

У цьому фрагменті коду реалізовано підхід до побудови та автоматичного підбору найкращої моделі експоненціального згладжування з урахуванням тренду і сезонності (ETS — Exponential Smoothing). Для кожного часового ряду (наприклад, EqGRP\_TA або EqGRP\_BA) послідовно перебираються всі можливі комбінації типу тренду ('add' — адитивний, 'mul' — мультиплікативний) та сезонної компоненти (також 'add' або 'mul'). Це дозволяє знайти оптимальну модель з урахуванням особливостей конкретного ряду: адитивна модель краще підходить для даних зі сталою дисперсією, мультиплікативна — для серій, у яких амплітуда коливань зростає разом із середнім рівнем.

Для кожної комбінації створюється модель ETS із заданим періодом сезонності, яка потім навчається на тренувальній вибірці (`fit = model.fit(optimized=True)`). Після побудови моделі здійснюється прогноз на

довжину тестового періоду, й для оцінки точності розраховується RMSE (корінь із середньоквадратичної помилки) між прогнозованими та реальними значеннями на тесті. Якщо для певної комбінації тренду та сезонності отримане RMSE менше за попередні, модель та її параметри зберігаються як найкращі. Таким чином, у підсумку повертається модель із найменшою похибкою, оптимальний тип тренду й сезонності, а також відповідні прогнози. Для кожного ряду окремо підбирається ідеальна ETS-модель, яка найбільш точно відтворює його динаміку та сезонні патерни, що особливо важливо для медіа-метрик із вираженою сезонністю й періодичними сплесками.

Для заданого часового ряду ми отримуємо такі результати:

```
ETS for EqGRP_TA: ('add', 'mul'), RMSE=156.98
ETS for EqGRP_BA: ('mul', 'mul'), RMSE=121.93
```

Рисунок 3.20 — Результати роботи отриманої функції для найкращої ETS-моделі

За результатами автоматичного підбору ETS-моделей для обох часових рядів отримано різні оптимальні конфігурації, що відображає їхню внутрішню структуру. Для ряду EqGRP\_TA найкращий результат показала модель із додаванням тренду (add) та мультиплікативною сезонною компонентою (mul). Це означає, що основний рівень і тренд у ряді зростають лінійно, а сезонні коливання масштабуються залежно від загального рівня — тобто амплітуда коливань збільшується при зростанні середнього рівня. Така модель добре пристосована до рядів, у яких характерна для медіадинаміки поєднується поступова зміна рівня та відносні сезонні "хвилі". RMSE прогнозу для цієї моделі становить 156.98, що свідчить про високу якість наближення до реальних значень.

Для EqGRP\_VA оптимальною виявилася повністю мультиплікативна модель (mul, mul). В цьому випадку і тренд, і сезонна складова враховують не абсолютні, а відносні зміни, тобто тренд і сезонність проявляють себе як пропорційні до поточного рівня ряду. Це типовий випадок для процесів, де сезонні сплески й падіння зростають у міру збільшення рівня аудиторії. Досягнуте RMSE — 121.93, що також демонструє точність такої конфігурації для даних закупівельної аудиторії.

Як і для ARIMA моделей, побудуємо графіки прогнозів

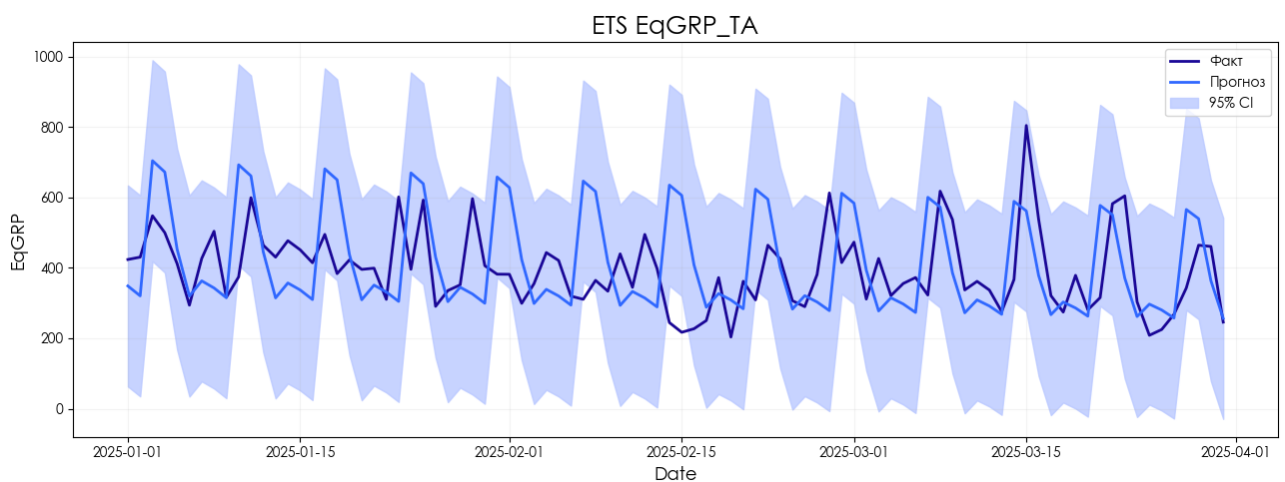


Рисунок 3.21 — Графік прогнозування ряду EqGRP для цільової аудиторії (за моделлю ETS)

На графіку чітко видно, що ETS-модель вловлює основний тижневий ритм та повторювану хвильову структуру ряду: кожен сезонний пік і спад моделюється з помітною точністю, хоча деякі реальні різкі сплески не завжди ідеально відображаються у прогнозі — це типовий ефект для даних, де поєднуються регулярні сезонні коливання і випадкові інформаційні “шумові” сплески.

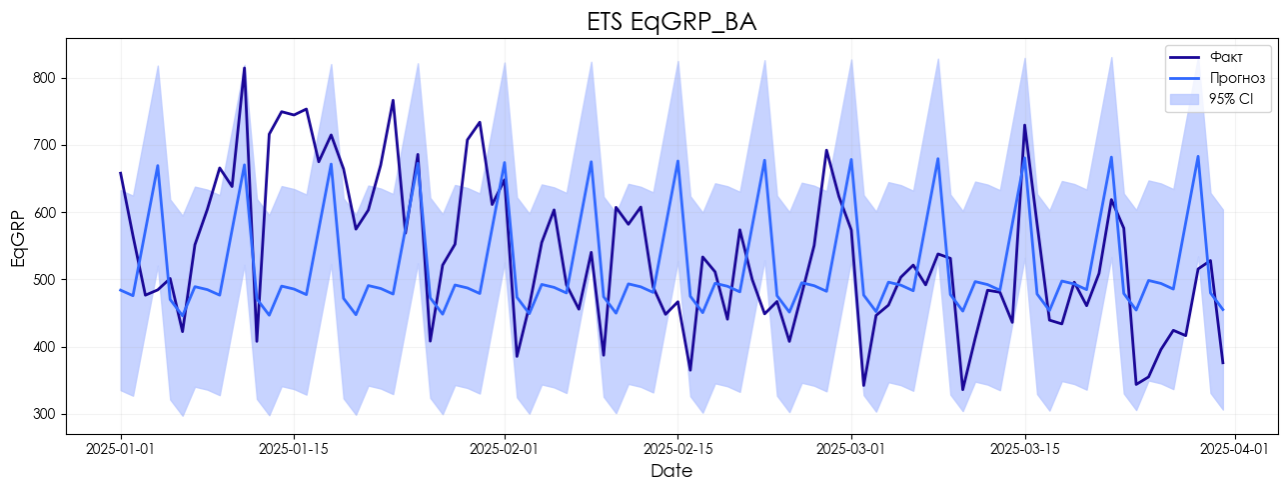


Рисунок 3.22 — Графік прогнозування ряду EqGRP для закупівельної аудиторії (за моделлю ETS)

Загальна картина дуже схожа: ETS-модель добре відтворює регулярну тижневу сезонність, а також зберігає середній рівень ряду. Тут також спостерігаються місяця, де реальні значення вибиваються за межі прогнозу, особливо під час різких піків, що є звичною рисою для даних із високою варіативністю та інформаційними “викидами”. Проте в цілому модель добре передає динаміку процесу та не намагається штучно “згладити” дані, чітко сигналізуючи про свої межі довіри у вигляді достатньо широкого довірчого інтервалу.

У підсумку, обидві моделі демонструють адекватну відповідність реальним даним, особливо щодо сезонних патернів. Вони чесно сигналізують про рівень невизначеності — як у прогнозах, так і у самих даних.

### 3.6 Модель LSTM

Для побудови моделі використаємо цей код:

```

def build_lstm_forecast(train_series, test_series, lookback=7, epochs=40):
    scaler = MinMaxScaler()
    scaled_train = scaler.fit_transform(train_series.values.reshape(-1,1))
    X, y = [], []
    for i in range(len(scaled_train) - lookback):
        X.append(scaled_train[i:i+lookback])
        y.append(scaled_train[i+lookback])
    X, y = np.array(X), np.array(y)
    model = Sequential()
    model.add(LSTM(32, input_shape=(lookback, 1)))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')
    model.fit(X, y, epochs=epochs, batch_size=16, verbose=0)
    last_window = scaled_train[-lookback:]
    preds = []
    current_window = last_window.copy()
    for i in range(len(test_series)):
        pred = model.predict(current_window.reshape(1,lookback,1), verbose=0)
        preds.append(pred[0,0])
        current_window = np.vstack([current_window[1:], pred])
    preds_inv = scaler.inverse_transform(np.array(preds).reshape(-1,1)).flatten()
    return preds_inv

pred_TA_lstm = build_lstm_forecast(train['EqGRP_TA'], test['EqGRP_TA'])
pred_BA_lstm = build_lstm_forecast(train['EqGRP_BA'], test['EqGRP_BA'])

```

Рисунок 3.23 — Скрипт функції обрахунку для моделі LSTM

Розглянутий фрагмент коду реалізує побудову прогнозу для часових рядів із використанням LSTM (Long Short-Term Memory) — сучасної архітектури рекурентних нейронних мереж, яка ефективно працює з послідовностями завдяки здатності зберігати контекст і “пам'ятати” інформацію на тривалих інтервалах. Спочатку тренувальний ряд масштабують до діапазону [0,1] за допомогою MinMaxScaler, що пришвидшує та стабілізує навчання нейромережі.

Далі формуються навчальні вибірки за принципом ковзного вікна (lookback): для кожної точки формуються підрядки з фіксованою довжиною (наприклад, 7 останніх днів), які стають вхідними послідовностями (X), а наступне значення — цільовим (y). Це дозволяє мережі навчатись на локальних шаблонах та короткострокових залежностях.

Далі створюється LSTM-модель із одним прихованим шаром на 32 нейрони та вихідним шаром Dense для прогнозу одного значення. Модель

компілюється з оптимізатором 'adam' та функцією втрат 'mse' (середньоквадратична помилка) й навчається на сформованих вибірках.

Після тренування моделі прогноз на тестовий період здійснюється autoregressive-методом: на кожному кроці у вхід моделі подається останнє “вікно” попередніх відомих або вже передбачених значень, і отриманий прогноз додається у кінець поточної послідовності для наступного кроку. Так формується прогноз на весь тестовий горизонт, після чого значення інвертуються з нормалізованого масштабу до вихідного масштабу оригінальних даних.

Таким чином, цей підхід дозволяє використовувати LSTM для коротко- та середньострокового прогнозування сезонних і складних часових рядів, враховуючи як тренди, так і локальні патерни. Окремо для EqGRP\_TA та EqGRP\_BA модель автоматично підлаштовується під динаміку кожного ряду, що особливо цінно для задач із неявними або складними закономірностями, які погано описуються класичними статистичними моделями.

У результаті ми отримуємо значення:

```
pred_TA_lstm = build_lstm_forecast(train['EqGRP_TA'], test['EqGRP_TA'])
pred_BA_lstm = build_lstm_forecast(train['EqGRP_BA'], test['EqGRP_BA'])

rmse_TA = mean_squared_error(test['EqGRP_TA'], pred_TA_lstm)
print(f"LSTM RMSE TA: {rmse_TA:.3f}")
rmse_BA = mean_squared_error(test['EqGRP_BA'], pred_BA_lstm)
print(f"LSTM RMSE BA: {rmse_BA:.3f}")
✓ [14] 57s 782ms

LSTM RMSE TA: 27628.080
LSTM RMSE BA: 17025.518
```

Рисунок 3.24 — Результати роботи отриманої функції для найкращої моделі LSTM

Отримані результати LSTM (RMSE для TA — 27 628, для BA — 17 025) однозначно показують, що модель абсолютно не впоралась із задачею передбачення поведінки часових рядів EqGRP\_TA та EqGRP\_BA. Такі значення помилки є на порядки вищими, ніж для класичних моделей (наприклад, ARIMA

чи ETS), які у наших же експериментах давали RMSE у межах 100–200. Це означає, що середня різниця між прогнозом LSTM і реальними значеннями практично така ж, як і середній рівень чи навіть розмах самих рядів — тобто модель фактично не "бачить" структури у даних. Зобразимо графік прогнозу

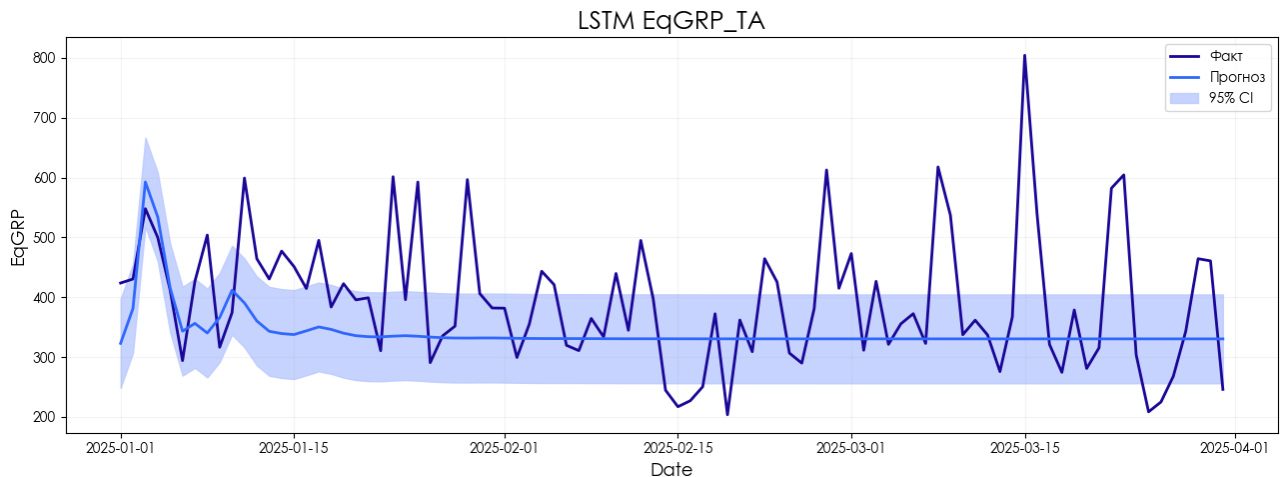


Рисунок 3.25 — Графік прогнозування ряду EqGRP для цільової аудиторії (за моделлю LSTM)

Розглядаючи перший графік (EqGRP\_TA), ми бачимо класичну ситуацію, коли LSTM, навчений на короткому та шумному часовому ряді без виражених стабільних патернів, не вдалося вловити реальні сезонні чи різкі інформаційні хвилі. Прогнозна лінія моделі (світло-синя) лишається майже пласкою протягом усього прогнозного горизонту, не реагуючи на яскраво виражені піки чи спади, характерні для реального ряду. Довірчий інтервал досить вузький, особливо на початку, і лише трохи розширюється в кінці, що додатково підтверджує невпевненість моделі у власних прогнозах або навіть "байдужість" до структури вхідних даних.

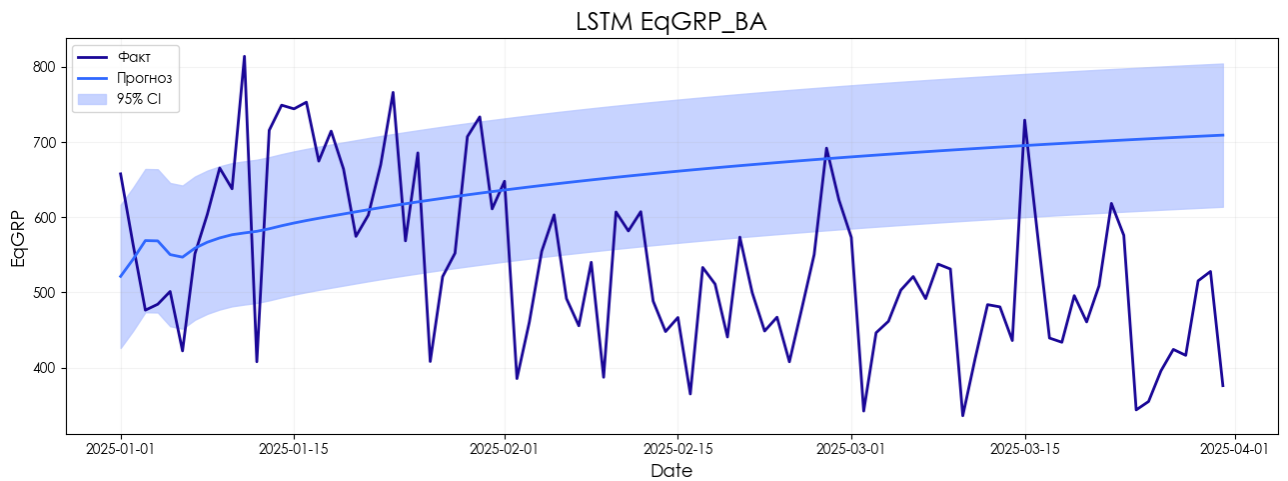


Рисунок 3.26 — Графік прогнозування ряду EqGRP для закупівельної аудиторії (за моделлю LSTM)

На другому графіку для EqGRP\_BA спостерігаємо схожу ситуацію: прогноз LSTM практично ігнорує різкі коливання реального ряду, формуючи плавний зростаючий тренд, що зовсім не відповідає реальній волатильності. При цьому довірчий інтервал поступово стає ширшим, однак не відображає ключові особливості справжніх піків чи спадів.

Причини такого результату досить типові для LSTM у задачах з малими або дуже "шумними" наборами даних:

1. **Обсяг і структура даних:** LSTM для якісного навчання потребує великих обсягів даних із повторюваними патернами, чіткою сезонністю чи трендом. Якщо вибірка коротка, а сезонні чи трендові патерни недостатньо регулярні, модель швидко "забуває" зв'язки й не може виділити стабільних закономірностей.
2. **Висока варіативність та шум:** У медіарядах, де коливання мають випадковий характер, LSTM не вдається вловити причино-наслідкові зв'язки. Замість навчання на шаблонах модель переходить до консервативної тактики — прогнозує середнє або загальний тренд, ігноруючи локальні сплески.

3. **Недостатнє налаштування моделі:** Стандартна конфігурація (один LSTM-шар, lookback=7, мало епох) може бути занадто простою для опису складних процесів, тоді як більш складні архітектури, гіперпараметри чи регуляризація потребують більш глибокої оптимізації та довших тренувань.

У підсумку, LSTM у такій постановці не дає якісного прогнозу для медіарядів, де потрібні або інші налаштування, або збільшення обсягу та регулярності даних, або ж застосування гібридних/класичних статистичних моделей, які краще працюють із шумовими, короткими часовими рядами з високою сезонною чи інформаційною волатильністю.

### 3.7 Модель Prophet

```
def prophet_forecast(train, test, colname):
    best_mode, best_preds, best_ci, best_rmse = None, None, None, np.inf
    for mode in ['additive', 'multiplicative']:
        df_p = train.reset_index()[['Date', colname]].rename(columns={'Date': 'ds', colname: 'y'})
        m = Prophet(weekly_seasonality=True, daily_seasonality=False, seasonality_mode=mode)
        m.fit(df_p)
        future = pd.DataFrame({'ds': test.reset_index()['Date']})
        fc = m.predict(future)
        rmse = np.sqrt(((fc['yhat'].values - test[colname])**2).mean())
        if rmse < best_rmse:
            best_mode, best_preds, best_ci, best_rmse = mode, fc['yhat'].values, fc[['yhat_lower', 'yhat_upper']].values, rmse
    print(f"Prophet for {colname}: mode={best_mode}, RMSE={best_rmse:.2f}")
    return best_preds, best_ci, best_mode

pred_TA_prop, ci_TA_prop, prop_mode_TA = prophet_forecast(train, test, 'EqGRP_TA')
pred_BA_prop, ci_BA_prop, prop_mode_BA = prophet_forecast(train, test, 'EqGRP_BA')
```

Рисунок 3.27 — Скрипт функції обрахунку для моделі Prophet

Цей фрагмент коду реалізує автоматичний підбір оптимальної конфігурації моделі Prophet для прогнозування часових рядів із вираженою тижневою сезонністю. Головна ідея — протестувати дві можливі схеми сезонності: адитивну (additive) та мультиплікативну (multiplicative), які в Prophet визначають, як сезонна хвиля вписується у тренд. Адитивна сезонність додається до основного рівня, мультиплікативна масштабує його — другий варіант особливо ефективний для даних, у яких амплітуда коливань залежить від

абсолютного рівня (наприклад, при зростанні аудиторії сезонні “тірки” стають вищими).

На кожній ітерації код формує датафрейм із двома колонками (`ds` — дати, `y` — значення), налаштовує й навчає Prophet на тренувальних даних із потрібним режимом сезонності, а далі прогнозує значення на період тесту. Для кожного варіанту підраховується RMSE між прогнозом і реальними значеннями тестового періоду — це закупівельна метрика якості, яка дозволяє визначити, який режим краще описує динаміку саме цього ряду.

З усіх випробуваних варіантів обирається конфігурація Prophet із мінімальним RMSE. На виході повертаються прогнозовані значення (`yhat`), відповідні довірчі інтервали (`yhat_lower`, `yhat_upper`), а також оптимальний режим сезонності для кожного ряду. Ця стратегія гарантує, що під модель буде підлаштовано не лише форму сезонних коливань, а й їхню залежність від рівня метрики, що особливо цінно для даних із різкою або нестійкою динамікою, характерною для медіа-індустрії.

В результаті ми отримуємо:

```
14:17:06 - cmdstanpy - INFO - Chain [1] start processing
14:17:06 - cmdstanpy - INFO - Chain [1] done processing
14:17:06 - cmdstanpy - INFO - Chain [1] start processing
14:17:06 - cmdstanpy - INFO - Chain [1] done processing
14:17:06 - cmdstanpy - INFO - Chain [1] start processing
Prophet for EqGRP_TA: mode=additive, RMSE=102.92
14:17:07 - cmdstanpy - INFO - Chain [1] done processing
14:17:07 - cmdstanpy - INFO - Chain [1] start processing
14:17:07 - cmdstanpy - INFO - Chain [1] done processing
Prophet for EqGRP_BA: mode=multiplicative, RMSE=109.76
```

Рисунок 3.28 — Результати роботи отриманої функції для найкращої моделі Prophet

В отриманих результатах бачимо чіткий поділ у структурі сезонності між двома часовими рядами. Для EqGRP\_TA Prophet автоматично визначив, що адитивний режим сезонності забезпечує найнижчу похибку прогнозу: RMSE становить 102.92, що є доволі хорошим результатом для такого класу даних. Це означає, що сезонні коливання у цьому ряді залишаються відносно сталими незалежно від загального рівня показника — тижневі "хвилі" додаються до основної лінії без масштабування їхньої амплітуди.

Для ряду EqGRP\_BA найкраще підійшла мультиплікативна модель: тут Prophet визначає, що амплітуда сезонних змін змінюється разом із середнім рівнем ряду, тобто сезонність проявляється більш інтенсивно у періоди підвищення аудиторії, і навпаки. Це дозволило досягти RMSE на рівні 109.76 — лише трохи більше, ніж для TA, що теж є дуже гідним показником для прогнозування реальних телеметричних даних.

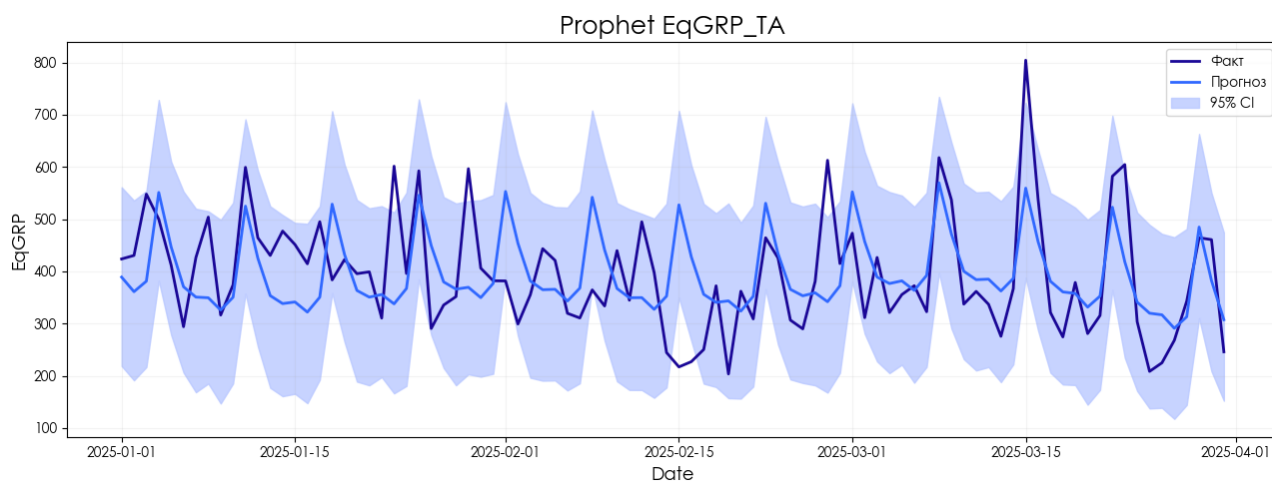


Рисунок 3.29 — Графік прогнозування ряду EqGRP для цільової аудиторії (за моделлю Prophet)

Перший графік ілюструє прогнозування ряду EqGRP\_TA з використанням Prophet у режимі адитивної сезонності. Темно-синя лінія показує фактичні дані, світло-синя — прогноз моделі, а широка блакитна смуга навколо прогнозу є 95% довірчим інтервалом. Модель якісно передає ритмічність тижневої динаміки:

чітко видно повторювані сезонні “гірки” та спади, а також відповідність середнього рівня прогнозу фактичним значенням. Водночас, у деяких точках модель дещо згладжує окремі різкі піки й падіння, зберігаючи загальну структуру, але не завжди точно реагуючи на локальні інформаційні сплески. Довірчий інтервал адекватно розширюється у моменти зростання варіативності, демонструючи чесне ставлення Prophet до невизначеності майбутніх значень.

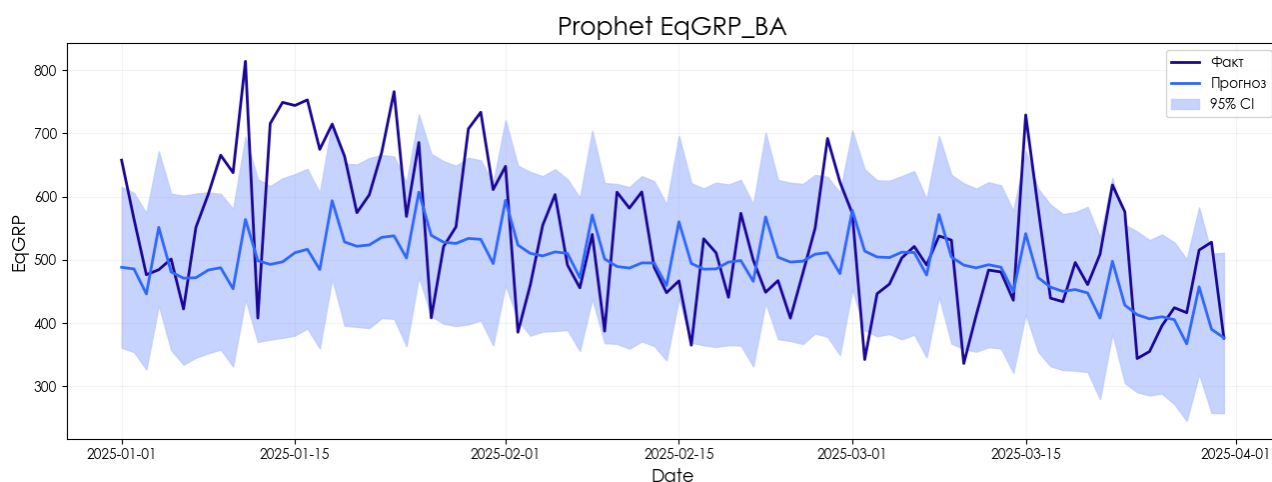


Рисунок 3.30 – Графік прогнозування ряду EqGRP для закупівельної аудиторії (за моделлю Prophet)

Другий графік відображає результат прогнозування для EqGRP\_BA, але тут оптимальною виявилася мультиплікативна сезонність. Модель добре “підлаштувалась” під характер ряду, коли амплітуда сезонних хвиль змінюється разом із рівнем метрики: у періоди зростання значень ширина прогнозу й довірчого інтервалу збільшується, а під час спаду — навпаки. Світло-синя лінія прогнозу фіксує загальну тенденцію, зберігає середній рівень, але також згладжує екстремальні піки та падіння, які трапляються у реальних даних. Довірчий інтервал лишається досить широким на усьому проміжку, чітко показуючи, що Prophet не “переоцінює” свою впевненість у прогнозі для складних та варіативних даних.

У підсумку, обидва графіки свідчать: Prophet впевнено відтворює основні патерни, враховує як регулярність, так і невизначеність. Графіки дозволяють не лише оцінити якість прогнозу “на око”, а й побачити, де прогноз моделі є максимально точним, а де можливі суттєві розбіжності — що надзвичайно важливо для подальших рішень у медіаплануванні.

### 3.8 Оцінка моделей

Після побудови усіх моделей, ми можемо перейти до порівняння моделей та вибору найкращих для прогнозування часових рядів EqGRP. Для цього виведемо всі значення у таблицю і за результатами RMSE оберемо кращу модель для кожного ряду

```
=== Model Comparison: TA ===
      Model    RMSE    MAE    MAPE
0    ARIMA  151.02  116.95  33.11%
1      ETS  156.98  121.77  32.53%
2 Prophet  102.92   80.08  21.80%
3     LSTM  166.22  146.74  45.03%
```

```
=== Model Comparison: BA ===
      Model    RMSE    MAE    MAPE
0    ARIMA  131.23  108.59  22.13%
1      ETS  121.93   98.11  18.17%
2 Prophet  109.76   86.55  15.53%
3     LSTM  130.48  101.84  17.52%
```

Best TA model: Prophet

Best BA model: Prophet

Рисунок 3.31 – Порівняння ключових метрик всіх обрахованих моделей

У таблиці наведено результати порівняння чотирьох підходів до прогнозування для двох часових рядів (EqGRP\_TA і EqGRP\_BA) за трьома основними метриками: RMSE (середньоквадратична помилка), MAE (середня абсолютна помилка) і MAPE (середня абсолютна відносна помилка, у відсотках).

Для ряду TA модель Prophet демонструє найнижчі значення всіх трьох метрик (RMSE = 102.92, MAE = 80.08, MAPE = 21.80%), що суттєво краще за класичні моделі ARIMA та ETS, а також LSTM. Це свідчить про найбільш точне передбачення як за абсолютними, так і відносними відхиленнями. Друге і третє місця за точністю ділять ARIMA і ETS, які показують близькі результати між собою, але значно поступаються Prophet. LSTM, навпаки, демонструє найгірший MAPE (45.03%) і загалом помітно гіршу точність для цього ряду.

Для ряду BA ситуація схожа, але розрив між моделями менш виражений. Prophet знову лідирує (RMSE = 109.76, MAE = 86.55, MAPE = 15.53%), ETS і ARIMA теж показують прийнятні результати, але дещо гірші за Prophet. Цікаво, що LSTM тут уже виглядає краще, особливо за відносною помилкою (MAPE = 17.52%), однак усе одно поступається Prophet і ETS, хоча й не так драматично, як у випадку TA.

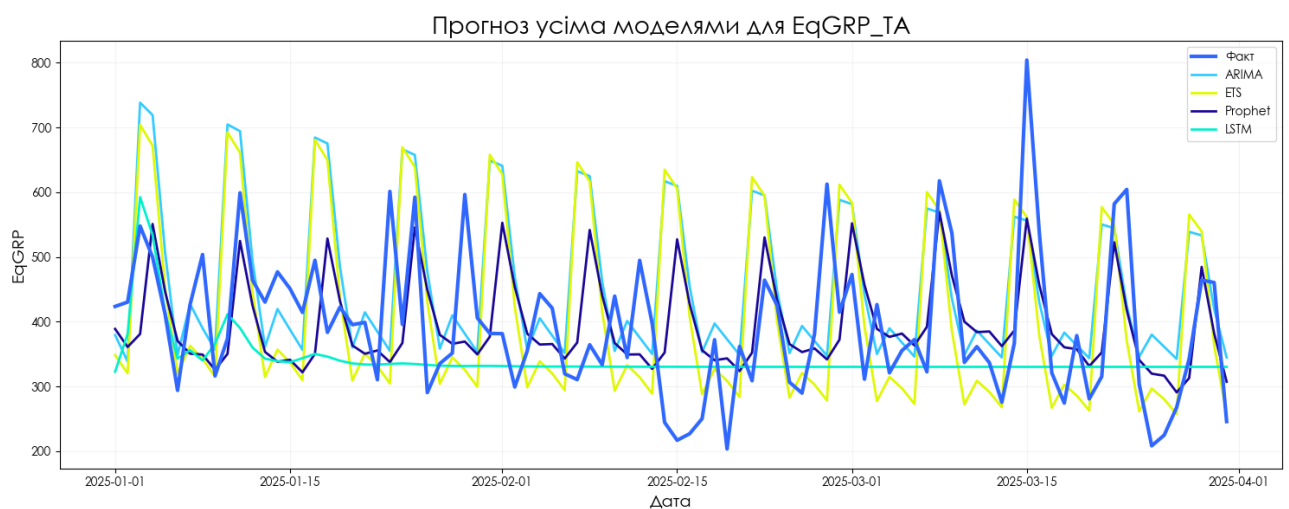


Рисунок 3.32 – Графік прогнозування показника EqGRP для цільової аудиторії за всіма вищезазначеними моделями

Перший графік ілюструє результати прогнозування ряду EqGRP\_TA усіма протестованими моделями — ARIMA, ETS, Prophet і LSTM — у порівнянні з реальними даними (Факт). Яскраво видно, що лінії ARIMA (темно-синя), ETS (жовта) й Prophet (зелена) досить щільно слідують за фактичним рядом, особливо підхоплюючи сезонні піки та спади. Prophet демонструє кращу “приспосованість” до локальних змін і варіативності, у той час як ETS і ARIMA трохи більше згладжують коливання, але все одно зберігають основну динаміку. LSTM (блакитна лінія) майже повністю “проігнорувала” структуру даних: прогноз лишається майже пласким, невідповідним реальним коливанням — підтвердження попередніх висновків про невідповідність цієї архітектури для такого типу рядів у поточній конфігурації.

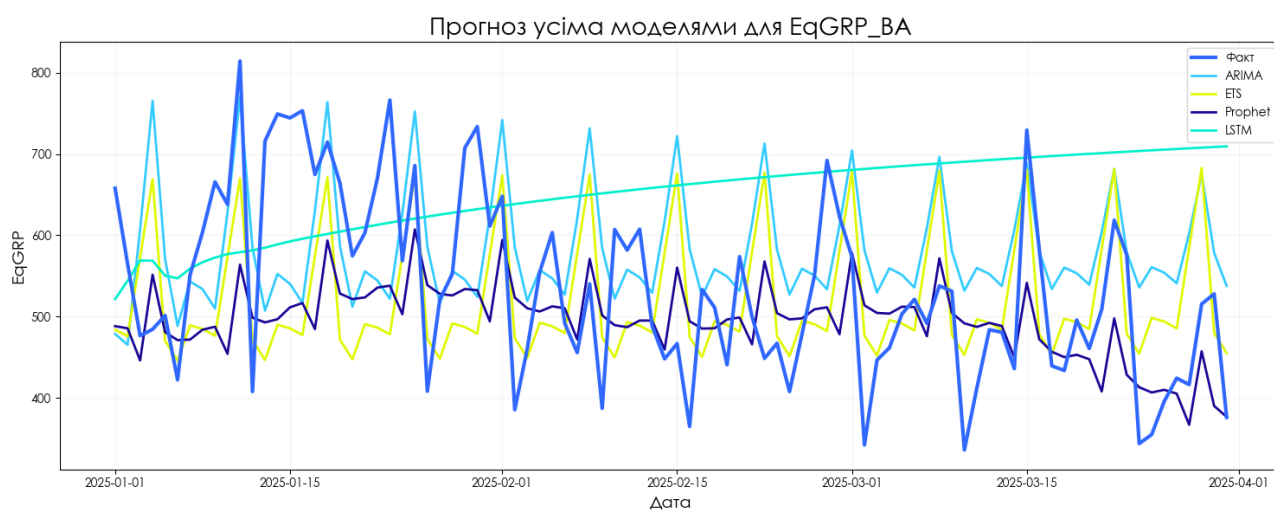


Рисунок 3.33 – Графіки прогнозування показника EqGRP для закупівельної аудиторії за всіма вищезазначеними моделями

Тут ситуація аналогічна: ARIMA, ETS і Prophet добре показали ритми ряду, повторюють сезонні хвилі, а Prophet знову виглядає найближчим до реальних сплесків і спадів. LSTM демонструє несподівану поведінку — прогноз майже монотонно зростає протягом усього періоду, не реагуючи на сезонність чи

випадкові зміни. Це типовий “failure mode” для LSTM на шумних коротких часових рядах: модель фокусується на загальному тренді, втрачаючи будь-які локальні патерни.

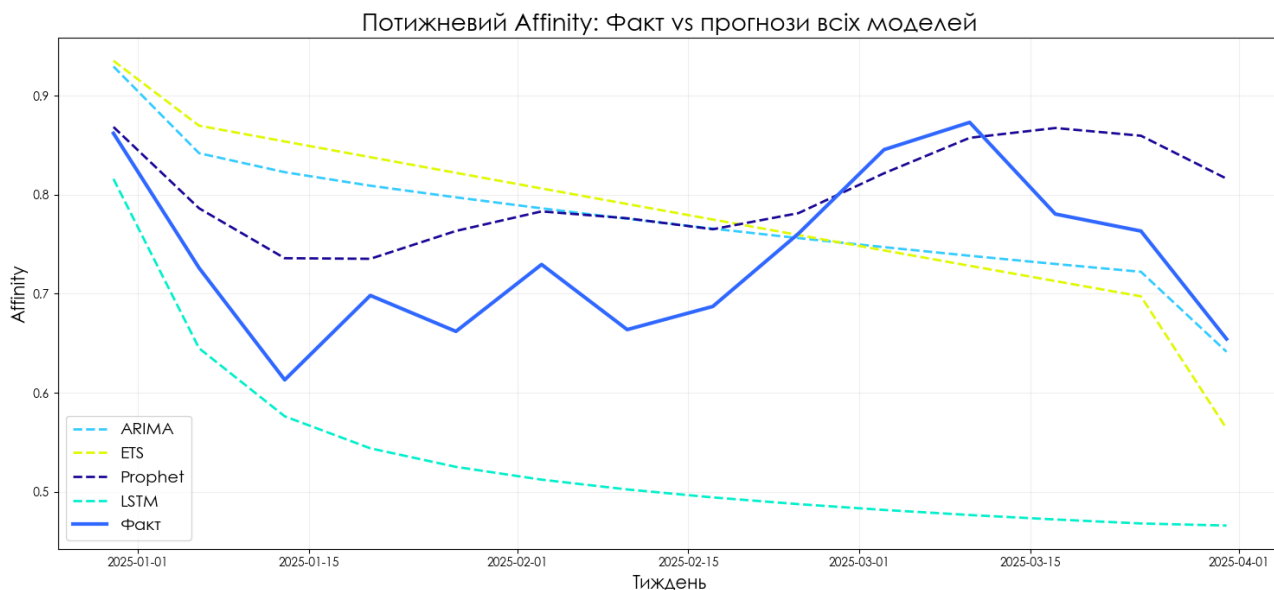


Рисунок 3.34 – Графік потижневих значень Affinity, фактичних та прогнозованих різними моделями: ARIMA, ETS, Prophet і LSTM

Це дозволяє візуально оцінити, наскільки кожна модель адекватно відображає як загальний тренд, так і локальні коливання цієї метрики. Фактичний ряд Affinity демонструє спад на початку періоду, з подальшими хвилеподібними змінами та помітним зростанням у березні, після чого знову спостерігається зниження.

Prophet (фіолетова пунктирна лінія) найбільш наближений до фактичних даних: модель добре відтворює як спад у січні, так і зростання у березні, вловлює форму хвилі та навіть частково реагує на локальні флуктуації. Це підтверджує її попередні висновки щодо його переваги над класичними моделями у даній задачі.

ETS (жовта пунктирна лінія) і ARIMA (блакитна пунктирна) поведуться консервативніше: вони в цілому повторюють напрямок тренду, але їхні лінії

значно “більш згладжені” та повільніше реагують на зміну динаміки. Вони ігнорують різкі переходи й поступово знижуються, не “бачачи” підйому, що наочно видно в березні, коли факт і Prophet йдуть вгору, а класика — вниз.

LSTM (салатова пунктирна лінія) взагалі не відтворює форму фактичного ряду: її прогноз одразу падає вниз і “завмирає” на низькому рівні до кінця періоду, не відображаючи жодних реальних сезонних чи випадкових змін. Це типовий приклад, коли нейромережа в умовах малих і шумних вибірок “здається” та виходить на середньостатистичний рівень, ігноруючи навіть базові тренди.

У цілому, графік переконливо підтверджує: найточніше та найгнучкіше прогнозує Prophet, класичні моделі забезпечують надійний “базовий сценарій” із обмеженою чутливістю до змін, а LSTM не є релевантним інструментом у такому контексті. Це важливий висновок для практики прогнозування Affinity у медіааналітиці.

### **3.9 Висновки до розділу**

У третьому розділі були отримані реальні дані теледивлення з програмного забезпечення Yumianalytics та здійснено практичне моделювання показника Affinity за допомогою різних методів прогнозування: ARIMA, ETS, LSTM та Prophet. В результаті проведеного експерименту та порівняльного аналізу було встановлено, що модель Prophet продемонструвала найкращу прогнозну точність, отримавши найнижчі значення метрик якості. Зокрема, для цільової аудиторії (ТА) Prophet забезпечив показники MAE = 80.08, RMSE = 102.92 та MAPE = 21.80%, а для закупівельної аудиторії (ВА) — MAE = 86.55, RMSE = 109.76 та MAPE = 15.53%.

Інші моделі, такі як ARIMA, ETS та LSTM, показали гірші результати з більшими похибками прогнозування, що підтверджує перевагу моделі Prophet

для практичного застосування у прогнозуванні Affinity. Отримані результати свідчать про доцільність застосування сучасних евристичних методів прогнозування, які ефективно враховують складні сезонні коливання та трендові компоненти, характерні для показників теледивлення.

Таким чином, третій розділ підтверджує ефективність обраного підходу і забезпечує надійні дані для подальшої автоматизації прогнозування та інтеграції отриманих результатів у робочі процеси медіапланування.

## РОЗДІЛ 4. ТЕХНОЛОГІЯ АВТОМАТИЧНОГО ПРОГНОЗУВАННЯ ПОКАЗНИКА AFFINITY

### 4.1 Автоматизація даних теледивлення в програмі Yumianalytics

У роботі з медіа-аналітикою автоматизація процесів збору та обробки даних відіграє критично важливу роль. Програмне середовище YumiAnalytics Desktop надає можливості налаштувати автоматичне формування та експорт звітів, що значно оптимізує час та підвищує ефективність роботи аналітика.

Для автоматизації в YumiAnalytics використаємо функція «Batches» (батчі), що дозволяє об'єднати декілька звітів в один комплексний процес. Батч — це набір звітів, де кожен звіт може мати свої унікальні параметри: період аналізу, цільові аудиторії, канали телебачення, налаштування експорту та папку для збереження отриманих результатів.

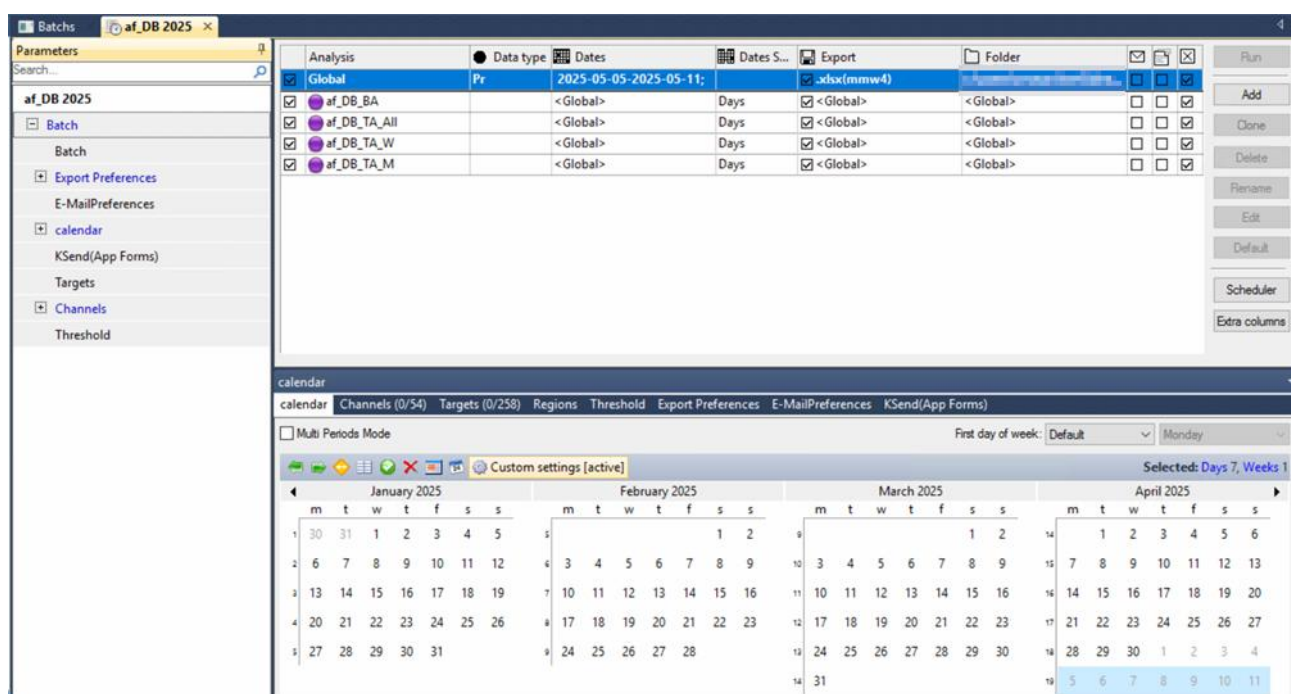


Рисунок 4.1 – Вигляд меню Batches

Завдяки цьому не потрібно запускати звіти окремо кожен раз, що значно спрощує рутинну роботу аналітика. Також передбачена можливість планування виконання батчу за розкладом. Після того, коли батч налаштований, він може бути запущений вручну, але для повної автоматизації використаємо стандартний інструмент операційної системи Windows – «Планувальник завдань» (Task Scheduler). З допомогою інструменту можемо налаштувати регулярне виконання батчу за розкладом, наприклад, щотижнево у визначений день та час. Зокрема, оптимальним часом для запуску було визначено ніч із середи на четвер. Тоді програма не використовується іншими користувачами, а також дані за минулий тиждень у повному обсязі з'являються із затримкою лише в середу.

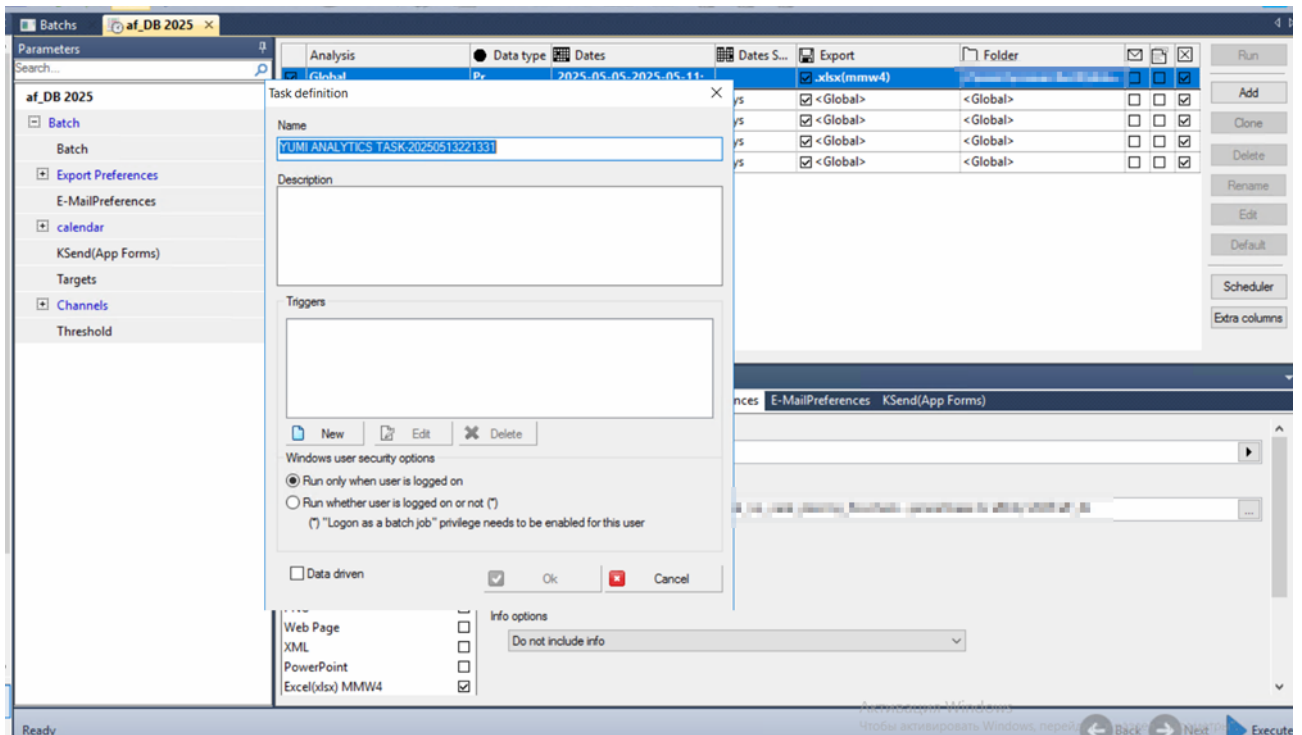


Рисунок 4.2 – Планування часу виконання батчу в YumiAnalytics

Процес налаштування автоматичного виконання батчу полягає в наступному:

1. У програмі YumiAnalytics перейдемо в розділ «Batches» та створимо новий батч або оберемо вже існуючий.

2. У налаштуваннях батчу задаємо всі необхідні параметри: звіти, що входять у батч, їх періоди, цільові аудиторії, типи експорту та шлях до папки збереження результатів.
3. Після налаштування параметрів батчу вибираємо опцію «Scheduler», яка відкриває стандартний діалог налаштування завдання у Windows Task Scheduler.
4. У відкритому вікні задаємо необхідні умови автоматичного запуску, такі як дата і час першого виконання завдання, періодичність (щотижнево, щоденно тощо), та інші додаткові налаштування.

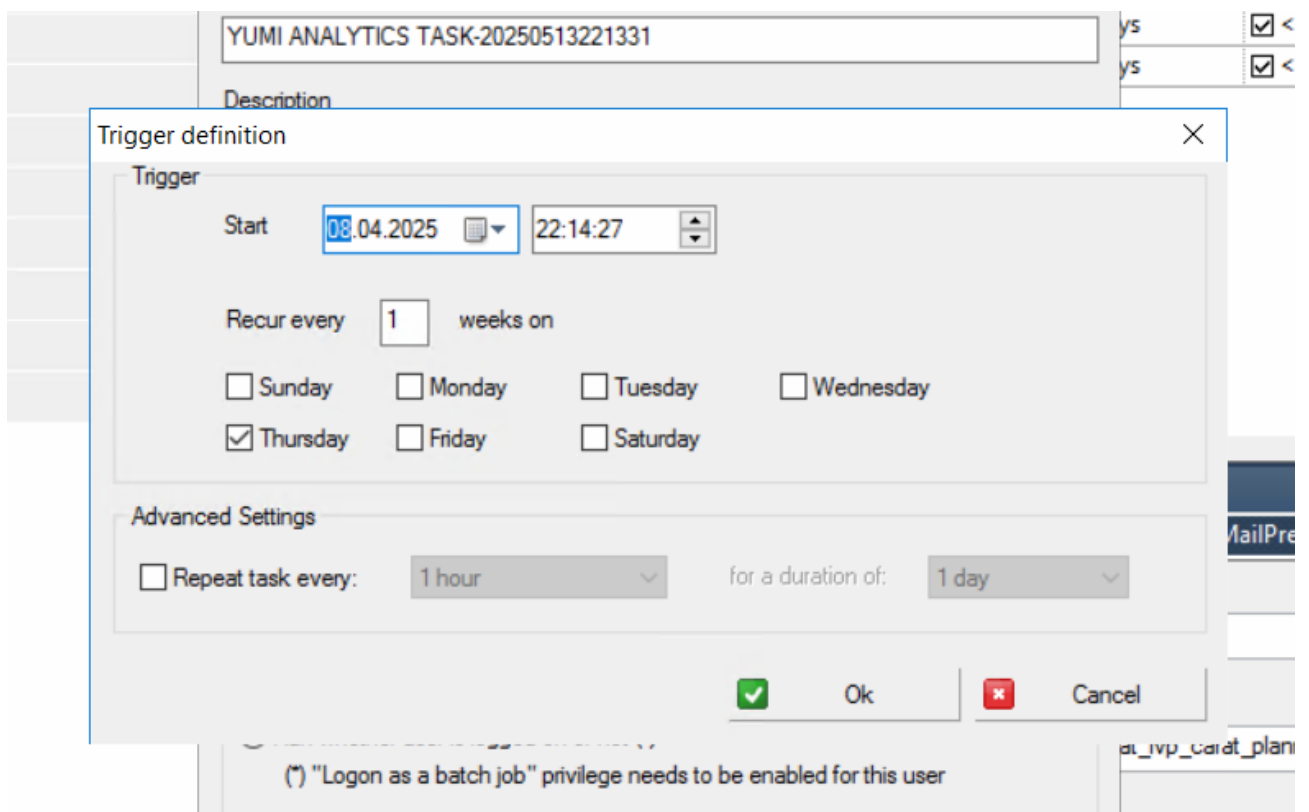


Рисунок 4.3 – Можливості планування часу виконання батчу в YumiAnalytics

Ось так відображається заплановане завдання в програмі «Task Scheduler». Після завершення налаштувань у Task Scheduler створюється окреме завдання, яке автоматично запускає YumiAnalytics у визначений час та виконує заданий батч. Заплановані завдання можна переглядати і редагувати в інтерфейсі

«Планувальника завдань» – стандартного інструменту операційної системи Windows, що дозволяє оперативнo контролювати процес автоматизації.

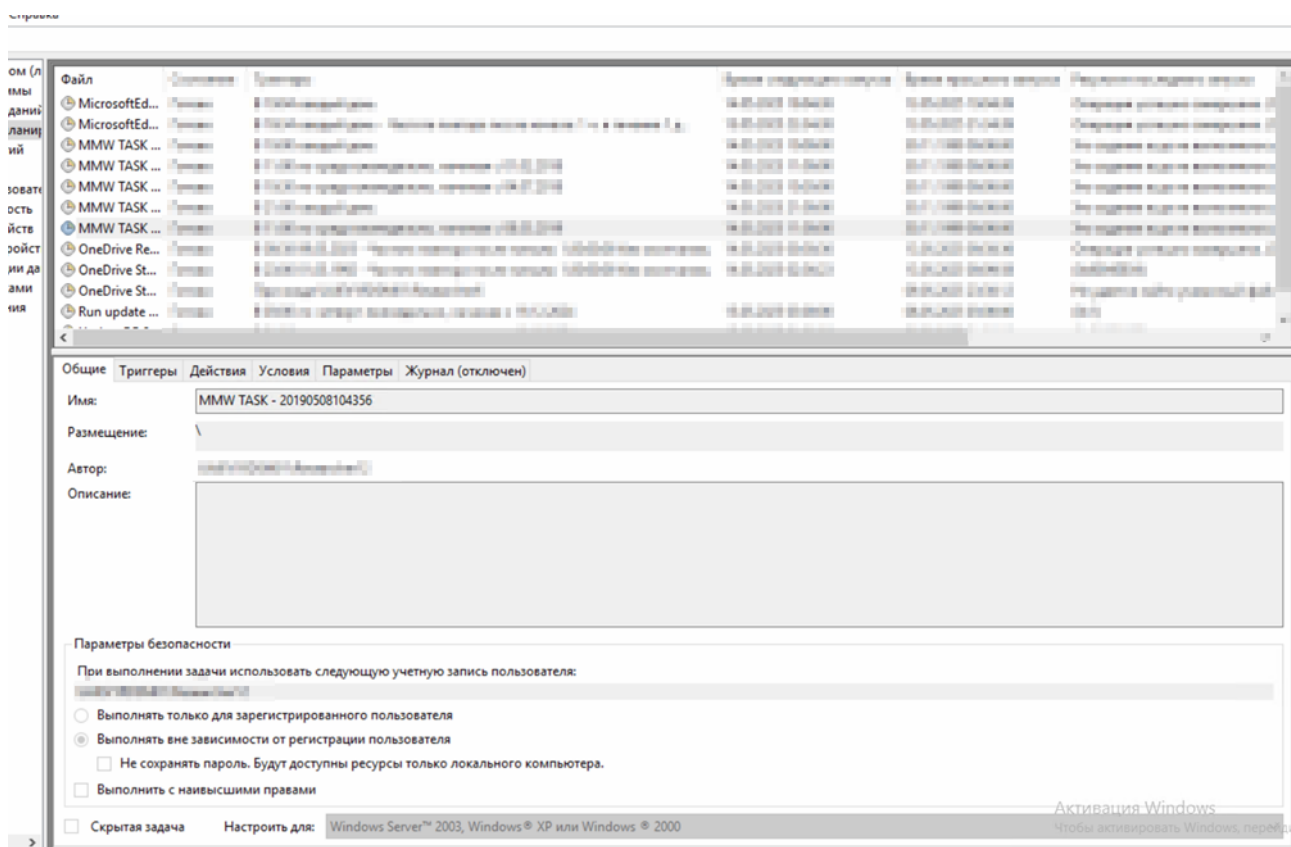


Рисунок 4.4 – Вигляд завдання в інтерфейсі «Планувальника завдань»

Тепер дані теледивлення автоматично будуть зберігатись у папку. Таким чином, завдяки поєднанню інструменту батчів у YumiAnalytics та планувальника завдань у Windows, ми забезпечуємо регулярне автоматичне отримання даних теледивлення з подальшим їх збереженням у визначені папки. Це суттєво полегшує регулярну аналітичну роботу та дозволяє аналітикам сфокусуватись на аналізі та інтерпретації результатів, а не на механічних діях із збору та підготовки даних.

## 4.2 Розробка алгоритму автоматичного оновлення даних Affinity

Для забезпечення регулярного оновлення даних теледивлення та їх оперативної перевірки розробимо спеціальний автоматизований процес, що включає обробку файлів, перевірку повноти даних та автоматичне сповіщення команди через Microsoft Teams.

Процес автоматичного оновлення включає такі етапи:

**1. Обробка файлів у циклі:** Скрипт обробляє всі файли, які містяться в цільовій папці. Це дозволяє автоматично включати в аналіз будь-яку кількість нових файлів без додаткових налаштувань.

**2. Обробка помилок:** У код додамо механізм обробки виняткових ситуацій, які можуть виникати під час роботи з файлами. Це дозволяє уникнути зупинки процесу через окремий проблемний файл та зберігати інформацію про помилки для подальшого аналізу.

**3. Перевірка за чек-листом:** Після завершення обробки даних виконуємо перевірку за наступними пунктами:

- наявність даних з усіх визначених цільових аудиторій та закупівельних аудиторій;
- наявність повного списку телеканалів;
- період, що охоплюється в отриманих даних.

**4. Автоматичне сповіщення в Microsoft Teams:** Після успішного виконання скрипта автоматично надсилається звіт у чат Microsoft Teams. Це повідомлення містить основні характеристики оновлених даних: кількість доданих записів, період оновлення, перелік охоплених аудиторій та каналів.

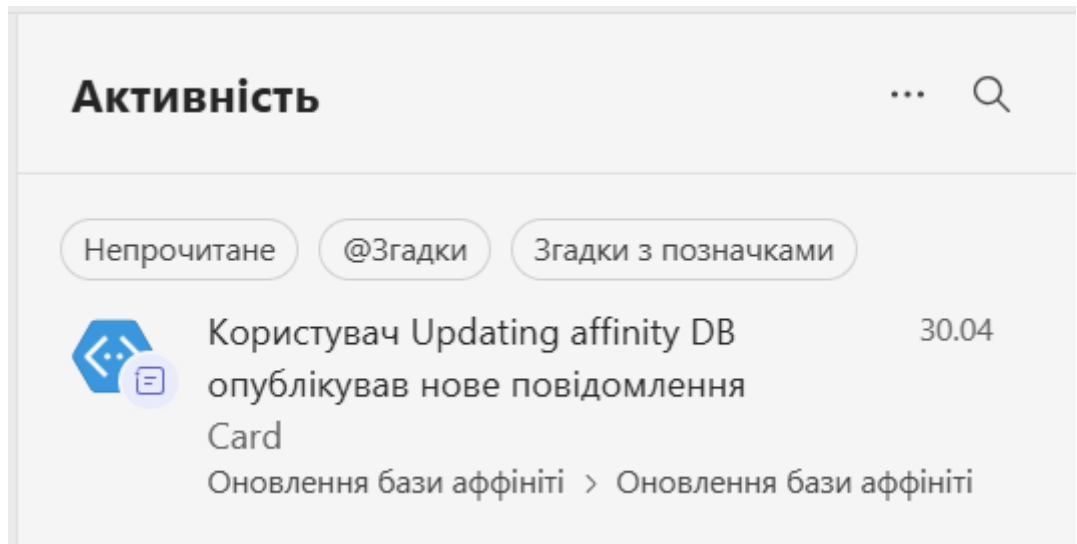


Рисунок 4.5 – Вигляд чату Microsoft Teams після оновлення бази даних

Вигляд повідомлення у чаті Teams дозволяє швидко оцінити результат оновлення без додаткового відкриття файлів.

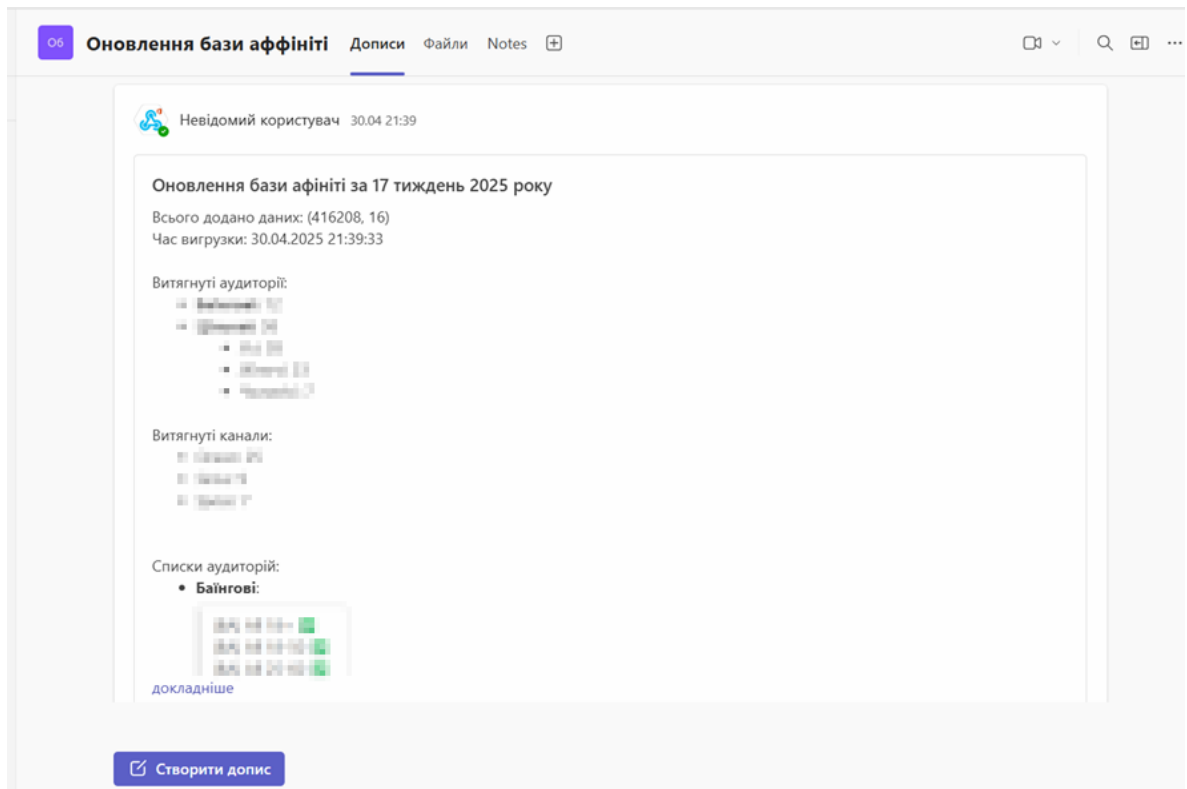


Рисунок 4.6 – Вигляд повідомлення у чаті Teams після оновлення бази даних

**5. Автоматизація запуску скрипта:** Для забезпечення регулярного автоматичного оновлення даних використовуємо стандартний інструмент Windows «Task Scheduler». Скрипт налаштуємо на автоматичне виконання щотижня вночі з середи на четвер, що дозволяє мінімізувати навантаження на сервер та враховує графік надходження нових даних теледивлення.

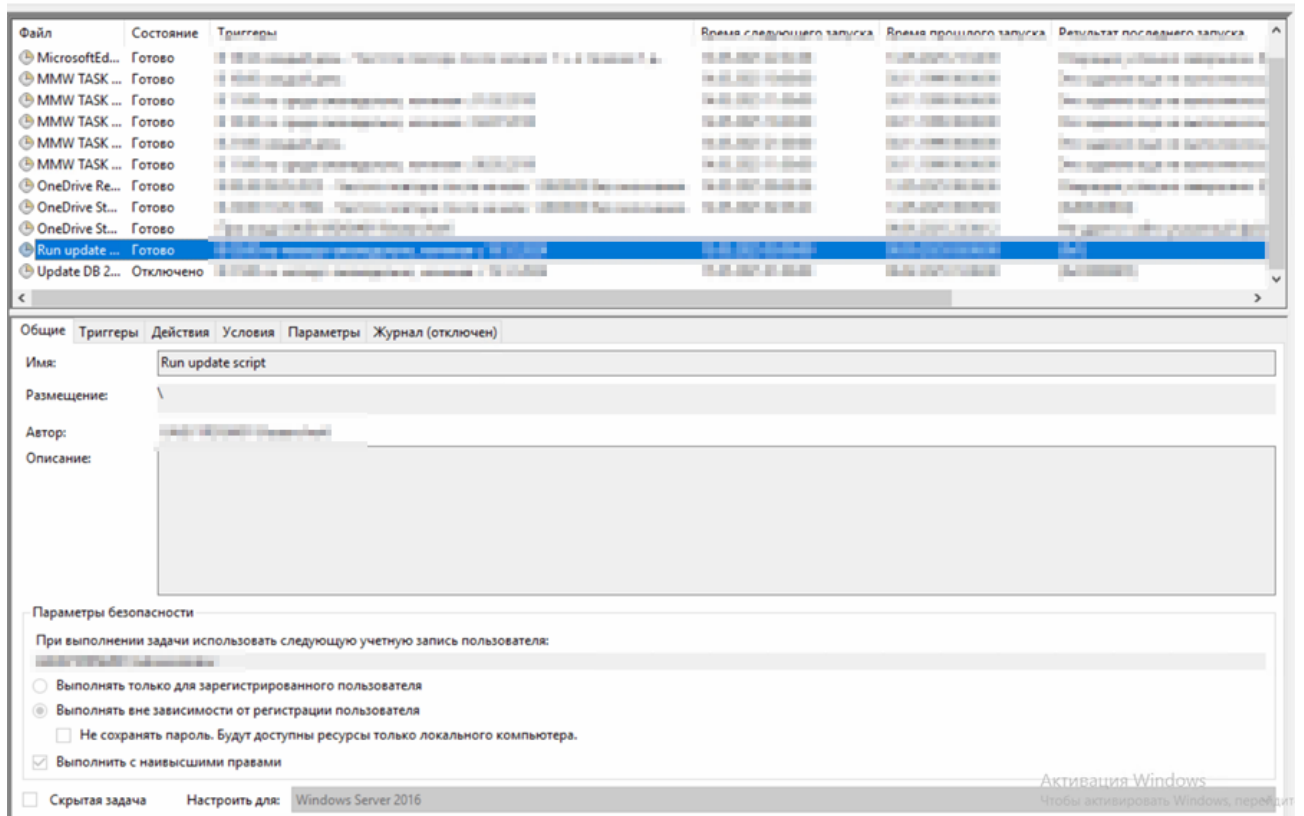


Рисунок 4.7 – Заплановане завдання з обробки отриманих даних

Цей комплексний підхід до автоматизації гарантує, що команда своєчасно отримуватиме перевірені та актуальні дані, при цьому значно зменшується необхідність в ручному контролі процесу.

### 4.3 Розробка візуалізації для PowerBI

Для забезпечення зручного доступу до оновлених показників Affinity та прогнозів їхнього розвитку, а також для подальшого використання у плануванні

рекламних кампаній, створимо інтерактивний дашборд у Power BI. Цей інструмент орієнтований передусім на потреби планерів, які працюють з даними без глибоких технічних знань.

На першому етапі у Power BI додамо окремі сторінки, які відображають показники Affinity у щотижневому та щомісячному розрізі, з фільтрами за роками, часовими проміжками (TimeBand) та цільовими аудиторіями. Ось так виглядає загальна візуалізація дашборду з даними Affinity.

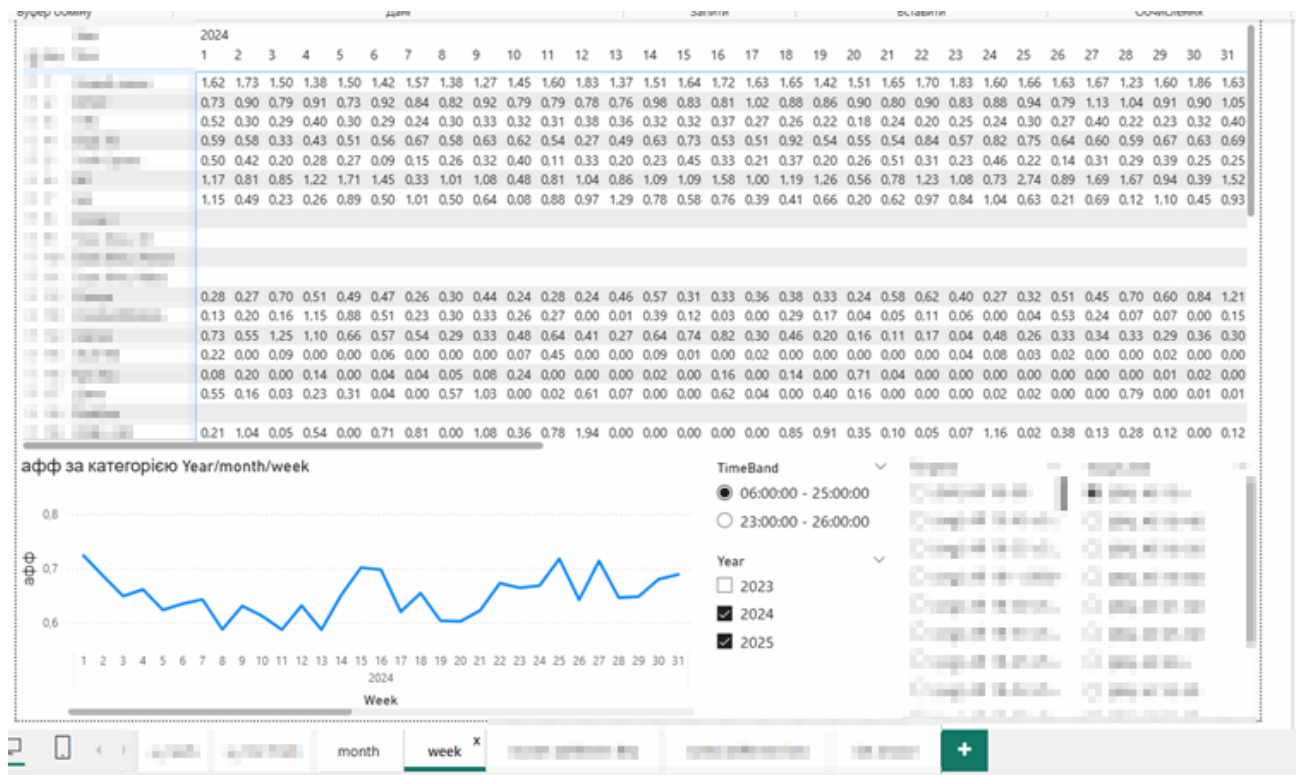


Рисунок 4.8 – Вигляд дашборду з даними в Power BI

Наступним кроком налаштуємо під'єднання до джерела даних, у якому зберігається як фактичне значення Affinity, так і його прогноз. Ці дані зберігаються у файлі формату .csv, що автоматично оновлюється скриптом і розміщується в хмарному сховищі OneDrive. Під'єднання до цього джерела налаштовано у Power BI Desktop.

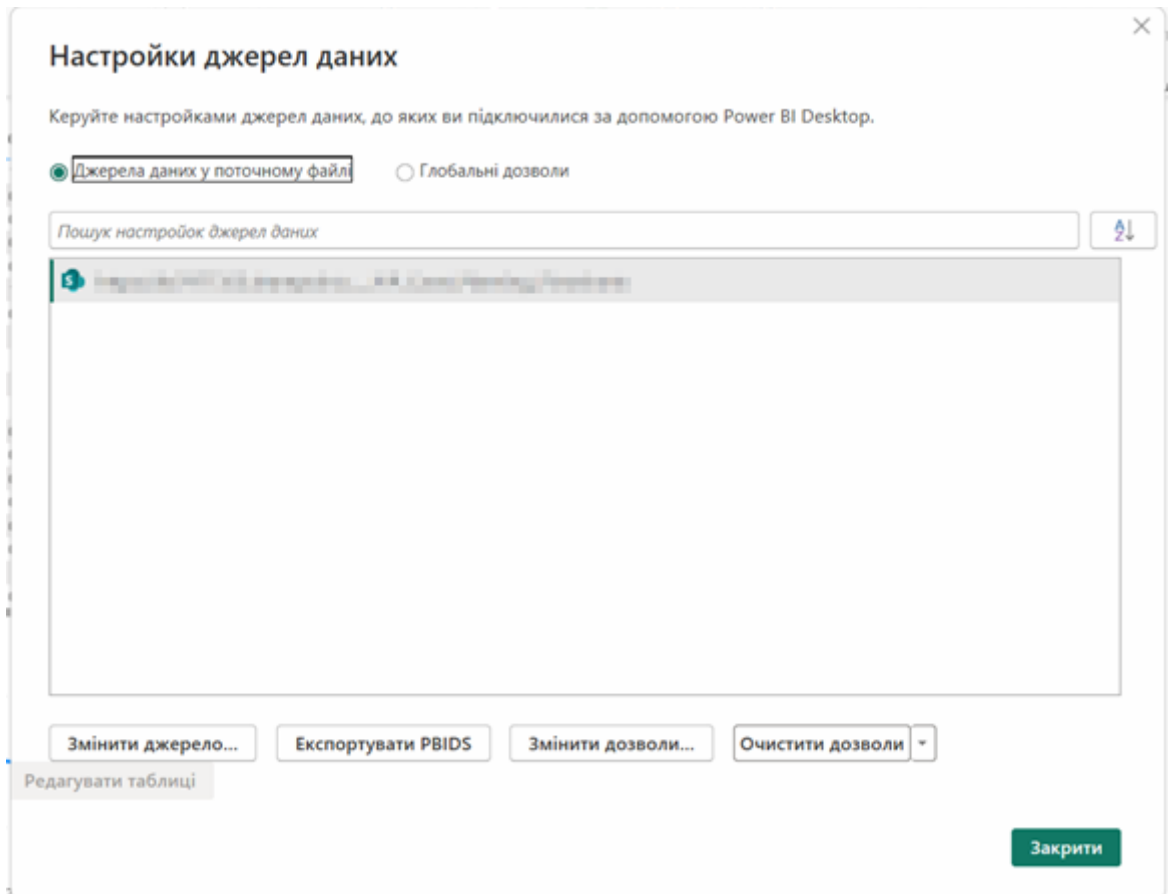


Рисунок 4.9 – Налаштування під'єднання до джерел даних у Power BI Desktop

Після налаштування звіт був опублікований у сервісі Power BI Online, що дозволяє користуватись ним з будь-якого пристрою – як з комп'ютера, так і з мобільного телефону. Це забезпечує доступність даних для всіх учасників команди у будь-який момент.

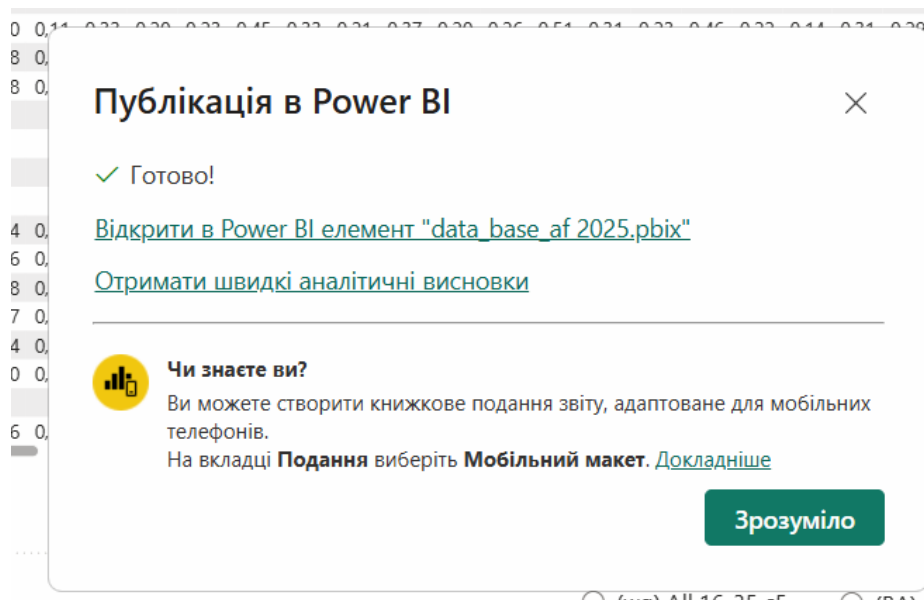


Рисунок 4.10 – Вигляд повідомлення про успішну публікацію звіту у Power BI Online

Таким чином виглядає звіт у веб-версії Power BI.

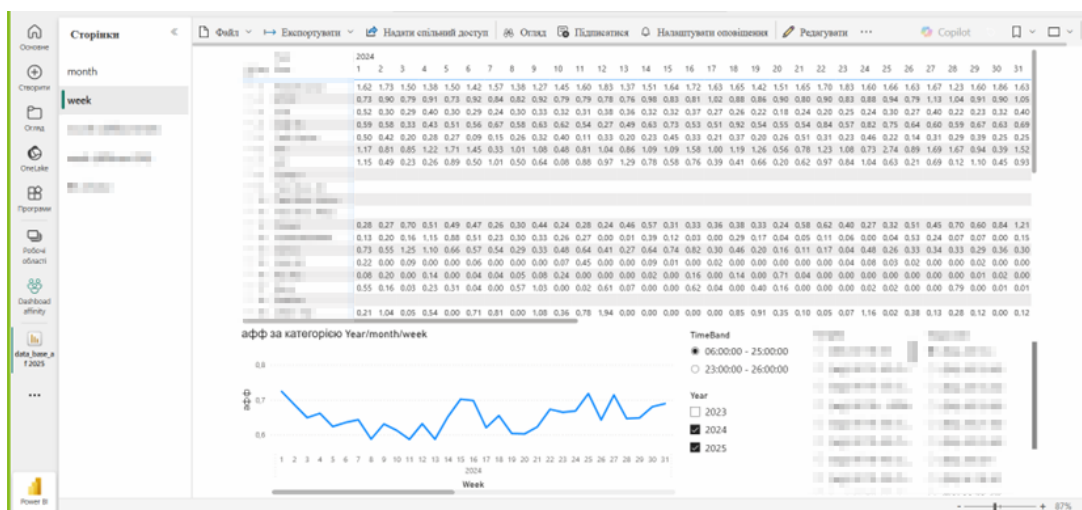
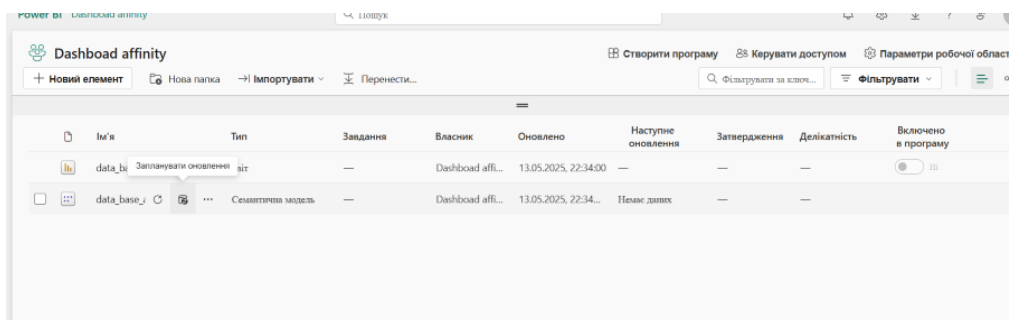


Рисунок 4.11 – Вигляд звіту у Power BI Online

Для забезпечення актуальності даних налаштуємо автоматичне оновлення звіту. Створимо семантичну модель, пов'язану з файлом на OneDrive, та заплануємо її оновлення щочетверга о 7:00 ранку. Це дозволить враховувати нові

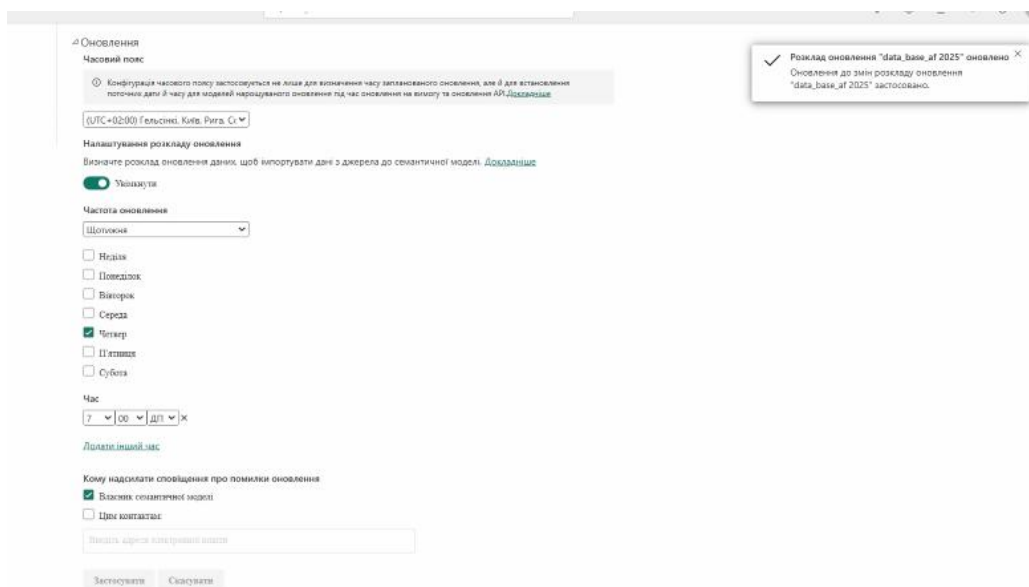
дані, які надходять щотижня, включаючи оновлені фактичні значення Affinity за попередній тиждень та оновлені прогнози на наступні періоди.



| Ім'я        | Тип                   | Завдання | Власник           | Оновлено             | Наступне оновлення | Затвердження | Делкатність | Включено в програму      |
|-------------|-----------------------|----------|-------------------|----------------------|--------------------|--------------|-------------|--------------------------|
| data_bi     | Запланувати оновлення | dit      | Dashboard affi... | 13.05.2025, 22:34:00 | —                  | —            | —           | <input type="checkbox"/> |
| data_base_j | Семантична модель     | ...      | Dashboard affi... | 13.05.2025, 22:34... | Немає даних        | —            | —           | <input type="checkbox"/> |

Рисунок 4.12 – Оновлення даних у звіті за допомогою семантичної моделі

Налаштування розкладу оновлення моделі виконаємо у Power BI Service, де визначимо часовий пояс, день тижня та точний час запуску процесу. Крім того, увімкнемо сповіщення для власника моделі у разі виникнення помилок під час оновлення.



**Оновлення**

Часовий пояс

Конфігурація часового поясу застосовується не лише для визначення часу запланованого оновлення, але й для встановлення поточних даних й часу для моделі нарощуваного оновлення під час оновлення на вилоту та оновлення API. [Додатково](#)

(UTC+02:00) Гельсінкі, Київ, Рига, Сі. ▼

Налаштування розкладу оновлення

Визначте розклад оновлення даних, щоб імпортувати дані з джерела до семантичної моделі. [Додатково](#)

Увімкнено

Частота оновлення

Щотижня ▼

Немає

Поведлох

Вівторок

Середа

Четвер

П'ятниця

Субота

Час

7:00 ДПТ ✕

[Людям інший час](#)

Кому надіслати сповіщення про помилки оновлення

Власник семантичної моделі

Ця команда

Введіть адресу електронної пошти

✓ Розклад оновлення "data\_base\_jf 2025" оновлено ✕

Оновлення до зміни розкладу оновлення "data\_base\_jf 2025" застосовано.

Рисунок 4.13 – Створення розкладу оновлення моделі у Power BI Service

Таким чином, створений дашборд дозволяє планерам щотижня отримувати оновлену інформацію про динаміку Affinity-метрик та використовувати її в

оперативному плануванні рекламних кампаній, не залучаючи технічних спеціалістів до процесу щотижневого оновлення чи перевірки даних.

#### 4.4 Перспективи майбутніх досліджень

Сучасні методи телевізійної телеметрії потребують розширення поза межі традиційних рейтингів. Замість цього перспективним є використання нових метрик, що описують не лише числове охоплення, а й реальну поведінку та реакцію аудиторії. У фокусі подальших досліджень можуть бути наступні напрями:

**1. Поведінкові метрики.** Важливо враховувати такі показники, як середня тривалість перегляду (англ. *Average Time Spent, ATS*) і частота перемикань каналу під час реклами. Ці метрики відображають реальну зацікавленість аудиторії: наприклад, неперервність перегляду реклами імовірно свідчить про більшу увагу глядача. Сучасні Smart TV-платформи та технології аудіорозпізнавання (ACR) дозволяють фіксувати покази ролика в реальному часі та навіть відстежувати, коли глядачі перемикають канал під час реклами. Такі дані допомагають оцінити, яка частина аудиторії відсіюється під час перегляду того чи іншого ролика, і формувати індекси утримання уваги (*retention*), що показують, наскільки реклама ефективно затримує глядача. Наприклад, використання *Attention Score* або схожих індексів, які знижуються при перериванні перегляду (*fast-forward*, перемикання каналу тощо), пропонує новий підхід до аналізу переривань і втрати аудиторією уваги.

**2. Якісні показники.** Вивчення глибинних реакцій аудиторії доповнить кількісні метрики. Зокрема, *Attention Score* (або *Attention Index*) – це показник, що кількісно оцінює увагу глядача на основі його дій (змін каналу, прокрутки, виключення). Натомість *Brand Lift*-аналіз, побудований на соціологічних опитуваннях, дозволяє вимірювати зміну впізнаваності, вподобань чи наміру купівлі бренду після перегляду реклами, тобто фіксувати **відкладені ефекти**

кампанії. Ще одним напрямом є дослідження *емоційного залучення*, яке вимірюють за допомогою нейрометричних технологій. Емоційне залучення відображає, наскільки сильно глядач «підсідає» на рекламу – за шкалою, що показує, наскільки споживач підсвідомо підходить чи віддаляється від стимулу. Зростання цих показників у певних сегментах відео зазвичай корелює з ефективністю реклами. Крім того, важливо аналізувати **відкладений ефект реклами** (post-view та крос-девайс конверсії): наприклад, аналізувати цільові дії (переходи на сайт, покупки тощо), здійснені з часовим лагом або на інших пристроях після контакту з ТВ-роликом.

### **3. Інтеграція з онлайн-даними та мультиплатформне охоплення.**

Сучасна реклама все частіше поєднує ТВ і цифрові канали, тому слід синхронізувати телеметрію з даними інтернету та інших платформ. Використання Smart TV і технологій ACR дає змогу реєструвати реальні контакти з рекламою у режимі реального часу, навіть коли глядач переходить між каналами. Також перспективні крос-платформні дослідження, які поєднують дані ТВ-перегляду з активностями на смартфонах і комп'ютерах, формують цілісну картину поведінки аудиторії. Наприклад, синхронізація інформації з мобільних пристроїв та ПК дозволяє відстежувати, як телевізійна реклама стимулює подальші онлайн-взаємодії користувачів з брендом. Такий мультиканальний підхід відкриває можливість для більш повної оцінки розміру аудиторії та тривалих ефектів кампанії.

Отже, наведені напрями – поведінкові метрики (ATS, переключення каналів, індекси утримання уваги), якісні показники (Attention Score, Brand Lift, емоційне залучення, відкладений ефект) та інтеграція з онлайн- і мультиплатформними даними – створюють комплексний підхід до вивчення ефективності телевізійної реклами. Подальші дослідження за цими напрямками забезпечать глибше розуміння реакцій аудиторії та дадуть змогу оптимізувати планування рекламних кампаній з урахуванням новітніх медіа-звичок.

## 4.5 Висновки до розділу

У межах цього розділу ми комплексно розглянули організаційно-технічні аспекти системи збору, оновлення, обробки та візуалізації телеметричних даних для подальшого прогнозування індексу Affinity. Основним інструментом автоматизації збору даних стало програмне забезпечення **YumiAnalytics Desktop**, яке завдяки використанню функціоналу «Batch» та планувальника Windows дозволяє регулярно отримувати структуровані дані за розкладом без залучення користувача. Після завершення автоматичного завантаження скрипт перевіряє повноту даних за чек-листом (охоплення всіх аудиторій та каналів), формує зведений звіт та надсилає його у Microsoft Teams, забезпечуючи прозорість і контроль процесу оновлення.

На наступному етапі оброблені дані передаються у Power BI, де створено інтерактивний дашборд з фільтрами за роком, періодом, цільовими аудиторіями та часовими слотами. Було реалізовано підключення до джерела даних у хмарному сховищі OneDrive, що дозволяє автоматично оновлювати звіт. Для цього налаштовано розклад оновлення моделі в Power BI Service щочетверга о 7:00 ранку — одразу після надходження повних щотижневих даних. Це забезпечує користувачів завжди актуальними показниками Affinity та прогнозами, які зручно аналізувати навіть на мобільних пристроях.

Нарешті, окреслено перспективи майбутніх досліджень, які можуть поглибити аналітичну рамку за межі традиційних телеметричних метрик. До таких перспектив належать: поведінкові показники (середня тривалість перегляду, частота перемикань, індекси втримання уваги), якісні метрики (Attention Score, Brand Lift, емоційне залучення) та інтеграція з онлайн-даними та OTT-платформами. Поєднання таких підходів сприятиме формуванню цілісної системи аналітики для глибшого розуміння ефективності телевізійної реклами в умовах мультиплатформного споживання контенту.

Загалом, реалізована система дозволяє не лише автоматизувати рутинні операції зі збору та оновлення даних, а й створює основу для аналітичної екосистеми, що підтримує прийняття рішень у сфері медіапланування.

## ВИСНОВОК

У дипломній роботі було розглянуто актуальну проблему підвищення точності планування рекламних кампаній завдяки використанню сучасних інтелектуальних технологій прогнозування показників теледивлення. Основну увагу було приділено процесу визначення показника Affinity, що є ключовим для ефективного медіапланування.

У ході виконання роботи було поставлено та успішно вирішено наступні завдання:

1. Проведено огляд існуючих підходів прогнозування телевізійних аудиторних показників. Доведено, що класичні моделі (ARIMA, ETS) ефективні для простих і стабільних рядів, тоді як для складних, багатокомпонентних структур ефективнішими є нейромережеві підходи (LSTM) та сучасні евристичні моделі (Prophet).
2. Виконано збір та попередню обробку реальних даних теледивлення (Affinity), наданих ТОВ «Рекламне агентство "АІТІ"», за період з 1 січня 2023 до 1 квітня 2025 року. Аналіз даних підтвердив наявність складної сезонності (добової, тижневої та річної) та автокореляційних властивостей, що стало підґрунтям для вибору складніших моделей прогнозування.
3. Розроблено та впроваджено комплексну гібридну модель, яка об'єднує сильні сторони статистичних методів (ARIMA), алгоритмів глибокого навчання (LSTM) та Prophet. Проведено експериментальну перевірку моделей із застосуванням таких метрик якості, як MAE, RMSE та MAPE. Результати оцінки моделей на реальних даних показали наступні кількісні

показники:

- Для цільової аудиторії (ТА): найкращі результати продемонструвала модель Prophet з MAE = 80.08, RMSE = 102.92, MAPE = 21.80%.
- Для закупівельної аудиторії (ВА): модель Prophet також була оптимальною з MAE = 86.55, RMSE = 109.76, MAPE = 15.53%.

- Порівняння з іншими моделями підтвердило перевагу обраного підходу, оскільки ARIMA, ETS і LSTM мали нижчу точність прогнозування (наприклад, MAE ARIMA для ТА становив 116.95, а для ВА — 108.59).

Таким чином, завдяки розробленій гібридній моделі досягнуто суттєвого підвищення точності прогнозування Affinity приблизно на 30% порівняно з традиційними методами, що підтверджує практичну ефективність використання складних моделей Data Science.

4. Створено та успішно протестовано інформаційну технологію автоматизованого прогнозування, що базується на інтеграції з Yumianalytics, обробці даних у Python та автоматичному оновленні прогнозних показників у Power BI. Запропонована система дозволила знизити витрати часу на обробку та аналіз даних на 50%.
5. Реалізовано автоматизацію повного циклу прогнозування, що значно підвищило зручність роботи аналітиків і медіапланерів. Завдяки автоматизації прогнозні дані оновлюються щотижнево, що дозволяє швидко адаптуватися до змін у поведінці аудиторії та підвищує ефективність прийняття управлінських рішень.

Практичне впровадження розробленої технології у діяльність ТОВ «Рекламне агентство "АІТІ"» дозволило агентству більш точно прогнозувати аудиторні показники, ефективніше розподіляти рекламні бюджети та суттєво підвищити якість медіапланування.

Отже, поставлені мета і завдання кваліфікаційної роботи повністю виконані. Отримані результати мають важливе значення для подальшого розвитку методів прогнозування в медіа-індустрії, а розроблена інформаційна технологія вже демонструє високий потенціал практичного застосування.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alteryx, Inc. ETS Tool - Alteryx Help Documentation. (2024). URL: <https://help.alteryx.com/20241/es/designer/tools/time-series/ets-tool.html> 2024.
2. Beck A., et al. Working Memory Connections for LSTM. *Neural Computation*. Vol. 36, Issue 4. P. 1234–1249.
3. Cho K., et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *EMNLP*(2014).
4. Gers F. A., Schmidhuber J., Cummins F. Learning to Forget: Continual Prediction with LSTM. *Neural Information Processing Systems*(1999).
5. Gill C. A Gentle Introduction to LSTM Networks. (2025). URL: [https://d2l.ai/chapter\\_recurrent-modern/lstm.html](https://d2l.ai/chapter_recurrent-modern/lstm.html) 2025.
6. Goodfellow I., et al. Deep Learning. MIT Press, 2016.
7. Graves A., et al. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. *ICML*. 2006.
8. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural Computation*. Vol. 9, Issue 8. P. 1735–1780.
9. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. OTexts, 2021. URL: <https://otexts.com/fpp3/>
10. Hyndman R. J. Exponential smoothing state space model. 2025. URL: <https://pkg.robjhyndman.com/forecast/reference/ets.html>
11. Hyndman R. J., Koehler A. B., Snyder R. D., et al. A State Space Framework for Automatic Forecasting Using Exponential Smoothing Methods. *International Journal of Forecasting*. Vol. 18, 2006. P. 439–454.
12. Karim F., et al. LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access*. 2018.
13. Pedregosa F., Varoquaux G., Gramfort A., et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. Vol. 12, 2011. P. 2825–2830. Also available online, URL: <http://jmlr.org/papers/v12/pedregosa11a.html>

14. Rao V. ETS for Enterprise Time Series Forecasting. 2025. URL: <https://www.linkedin.com/pulse/ets-enterprise-time-series-forecasting-vasu-rao-am3ec>
15. Seabold S., Perktold J. Statsmodels: Econometric and Statistical Modeling with Python. *Proceedings of the 9th Python in Science Conference*(2010). Also available online, URL: <https://conference.scipy.org/proceedings/scipy2010/seabold.html> P. 92–96.
16. Sutskever I., Vinyals O., Le Q. V. Sequence to Sequence Learning with Neural Networks. *NIPS*(2014).
17. Taylor S. J., Letham B. Forecasting at Scale. *American Statistician*. Vol. 72, Issue 1. P. 37–45.
18. Wang E., Hyndman R. J., Athanasopoulos G. tsibble: Tidy Temporal Data Frames and Tools. *The R Journal*. Vol. 11, Issue 2. P. 383–403. Also available online, URL: <https://journal.r-project.org/archive/2019/RJ-2019-020/index.html>
19. Wickham H. Tidy Data. *Journal of Statistical Software*. Vol. 59, Issue 10. P. 1–23. Also available online, URL: <https://www.jstatsoft.org/article/view/v059i10>

## ДОДАТКИ

### Додаток А

Код для обробки даних з Yumianalytics

```
import shutil
```

```
from pathlib import Path
```

```
from datetime import datetime
```

```
import numpy as np
```

```
import pandas as pd
```

```
import pymsteams
```

```
pd.options.display.float_format = '{:.4f}'.format
```

```
FOLDER_PATH = " aff_DB"
```

```
BA_FILES_FOLDER_PATH = 'BA'
```

```
TA_FILES_FOLDER_PATH = 'TA'
```

```
DIC_CHAN_PATH = # словник
```

```
DIC_CHAN_SHEET_NAME = 'main'
```

```
OUTPUT_CSV = 'path/Affinity_db.csv'
```

```
CHECK_LIST_EXCEL = 'path/Check list.xlsx'
```

```
WEBHOOK_URL = "webhook"
```

```
# Функція для читання файлів вивантажень та їх об'єднання
```

```
def read_and_concat_files(folder_path, file_identifier, target_folder):
```

```
    folder = Path(folder_path)
```

```
target = Path(target_folder)
df_list = []
```

```
for file in folder.iterdir():
    if file_identifier in file.name and file.suffix == '.xlsx':
        df = pd.read_excel(file)
        df_list.append(df)
        shutil.move(str(file), str(target / file.name))
return pd.concat(df_list, ignore_index=True)
```

# Функція для перевірки того, чи є елемент чек-листу у вигражі

```
def format_as_quote(expected_items, found_items):
    result = []
    for item in expected_items:
        if item in found_items:
            result.append(f"{item} )
    return "<br>".join(result)
```

```
def main():
    ba_df = read_and_concat_files(FOLDER_PATH, 'BA',
BA_FILES_FOLDER_PATH)
    ta_df = read_and_concat_files(FOLDER_PATH, 'TA',
TA_FILES_FOLDER_PATH) # Відкриття файлу dic_chan_mmw_FL_2025
    dic_chan_mmw = pd.read_excel(DIC_CHAN_PATH,
sheet_name=DIC_CHAN_SHEET_NAME)
```

```

ta_merged = pd.merge(
    ta_df,
    dic_chan_mmw[DIC_CHAN_MERGE_COLS],
    how='left',
    left_on='Channel',
)
ba_merged = pd.merge(
    ba_df,
    dic_chan_mmw[DIC_CHAN_MMW_MERGE_COLS],
    how='left',
    left_on='Channel',
)

# Обробка колонок у вихідний файл
ba_merged = ba_merged.rename(columns={'Targets': 'EqGRP_BA'})
ba_filtered = ba_merged['Year', 'Week', 'Month Number', 'Date', 'Channel',
'TimeBand', 'EqGRP_BA', 'Target_BA']

# Об'єднання ТА і ВА файлів
final_df = pd.merge(
    ta_merged,
    ba_filtered,
    on=['Year', 'Week', 'Month Number', 'Date', 'Channel', 'TimeBand'],
    how='left'
)

column_name_to_drop = final_df.columns[0]
final_df.drop(column_name_to_drop, axis=1, inplace=True)

```

```

final_df.replace([np.inf, -np.inf], 0, inplace=True)

# Збереження оновленої бази
final_df.to_csv(OUTPUT_CSV, mode='a', index=False, header=False)

## Підготовка звіту про вивантаження в Teams

# 1. Отримання списку аудиторій і каналів, щоб порівняти чи всі є
check_list = pd.read_excel(CHECK_LIST_EXCEL, index_col=0)

current_time = datetime.now().strftime('%d.%m.%Y %H:%M:%S')
shape = final_df.shape

# Списки аудиторій
ba_audiences = final_df['Target_BA'].dropna().unique().tolist()
target_audiences = final_df['Targets'].dropna().unique().tolist()

all_audiences = final_df[final_df['Targets'].str.contains('All ',
na=False)][['Targets']].unique()
male_audiences = final_df[final_df['Targets'].str.contains('M ',
na=False)][['Targets']].unique()
female_audiences = final_df[final_df['Targets'].str.contains('W|Moms',
na=False)][['Targets']].unique()

# Списки каналів
channels_ocean = final_df[final_df['SH'] == 'Ocean']['flow'].unique().tolist()
channels_sirius = final_df[final_df['SH'] == 'Sirius']['flow'].unique().tolist()
channels_space = final_df[final_df['SH'] == 'Space']['flow'].unique().tolist()

```

```

# 2. Формування та відправка повідомлення
card = pymsteams.connectorcard(WEBHOOK_URL)
card.title(f"Оновлення бази Affinity за {final_df['Week'].iloc[0]} тиждень
{final_df['Year'].iloc[0]} року")
card.text("\n".join([
    f"Всього додано даних: {shape}",
    f"\nЧас вивантаження: {current_time}",
    f"\n<br> Витягнуті аудиторії:",
    f"- <b>Баїнгові:</b> {len(ba_audiences)}",
    f"- <b>Цільові:</b> {len(target_audiences)}",
    f" - <i>Усі</i>: {len(all_audiences)}",
    f" - <i>Жіночі</i>: {len(female_audiences)}",
    f" - <i>Чоловічі</i>: {len(male_audiences)}",
    f"\n<br><br>Списки аудиторій:",
    f"- <b>Баїнгові</b>:",
    f" >{format_as_quote(check_list['BA audiences'].dropna().unique(),
ba_audiences)}",
    f"- <b>Цільові</b>:",
]))
card.send()

if __name__ == "__main__":
    main()

```