МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАДИМА ГЕТЬМАНА

Інститут інформаційних технологій в економіці
(назва навчально-наукового інституту / факультету)

Кафедра математичного моделювання та статистика
(назва кафедри)

ЕКОНОМІЧНА КІБЕРНЕТИКА І ДАТА САЙНС

Галузь знань 05 «Соціальні та поведінкові науки»

Спеціальність 051 «Економіка»

Форма навчання: очна (денна)
очна (денна), заочна, дистанційна

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему **«Розробка веб-додатку для візуалізації та прогнозування даних»**
(назва теми)

здобувача Єгупова Кирила Олексійовича
(ПІБ, підпис)

Науковий керівник: кандидат економічних наук, доцент Осипова О. І.
(науковий ступінь, учене звання, ПІБ)

_____
(підпис)

Робота допущена до захисту перед екзаменаційною комісією з атестації здобувачів вищої освіти (ЕК)

Завідувач кафедри: кандидат фізико-математичних наук, професор Великоіваненко Г.І.
(науковий ступінь, учене звання, ПІБ)

_____
(підпис)

Київ 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА

Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра математичного моделювання та статистики

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА	**ЕКОНОМІЧНА КІБЕРНЕТИКА І ДАТА САЙНС**
ГАЛУЗЬ ЗНАНЬ	**05 СОЦІАЛЬНІ ТА ПОВЕДІНКОВІ НАУКИ**
СПЕЦІАЛЬНІСТЬ	**051 ЕКОНОМІКА**

ПОГОДЖЕНО
Керівник проектної групи (гарант)
освітньо-професійної програми
О.В. Піскунова
(підпис)
20___ р.

ЗАТВЕРДЖУЮ
Завідувач кафедри математичного
моделювання та статистики
Г.І.Великоіваненко
(підпис)
20___ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувачу вищої освіти Єгупову Кирилу Олексійовичу
очної(денної) форми навчання
на підготовку кваліфікаційної магістерської роботи
на тему **«Розробка веб-додатку для візуалізації та прогнозування даних»**
Тему затверджено наказом ректора Університету від «12» вересня 2023 р. № 1328-ст

Кваліфікаційна магістерська робота виконується на матеріалах
ФОП «Єгупов Кирил Олексійович»

План кваліфікаційної магістерської роботи

Розділ 1	Огляд наявних інструментів для автоматизованої роботи із даними
(назва розділу)	
Розділ 2	Методичний підхід до розробки та обґрунтування вибору інструментарію для створення веб-додатку
(назва розділу)	
Розділ 3	Практична реалізація розробленого веб-додатку
(назва розділу)	

Об'єкт дослідження:	автоматизована обробка специфічного набору даних.
Предмет дослідження:	методи, технології та інструментальні засоби створення веб-додатку для автоматизації задач візуалізації та моделювання специфічного набору даних.
Мета кваліфікаційної магістерської роботи:	розробка веб-додатку на мові програмування R для автоматизації обробки та аналізу набору даних та вивчення потенціалу розробленого веб-додатку для подальшого використання в схожих задачах корпоративного сектору.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі 1	Провести аналіз та порівняння основних інструментів для автоматизованої обробки даних, визначивши їхні переваги та обмеження.
У розділі 2	Оцінити можливості бібліотеки R "Shiny" для створення веб-додатку та аргументувати вибір вказаної технології. Описати математичні моделі та методи, використані у розробленому веб-додатку.
У розділі 3	Розробити структуру веб-додатку, надати детальний огляд розроблених функцій та їхню практичну застосовність та описати роботу веб-додатку для аналізу та моделювання даних.

Завдання підготував науковий керівник _____ **О. І. Осипова**

(підпис) *(ініціали, прізвище)*

«_____» _____ 20___ р.

Завдання одержав здобувач **К.О. Єгупов**

_____

(підпис) *(ініціали, прізвище)*

«_____» _____ 20___ р.

Реферат

Кваліфікаційна магістерська робота містить 104 сторінки, 22 рисунки, список використаних джерел з 54 найменувань, додатки.

«Розробка веб-додатку для візуалізації та прогнозування даних»

Об'єктом дослідження є автоматизована обробка специфічного набору даних.

Предметом дослідження є методи, технології та інструментальні засоби створення веб-додатку для автоматизації задач візуалізації та моделювання специфічного набору даних.

Метою роботи є розробка веб-додатку на мові програмування R для автоматизації обробки та аналізу набору даних та вивчення потенціалу цього веб-додатку для подальшого використання в схожих задачах корпоративного сектору.

Відповідно до поставленої мети, сформульовані такі завдання:

- провести огляд наявних інструментів для візуалізації та аналізу даних та визначення можливих шляхів імплементації деяких елементів цих інструментів до розробленого веб-додатку;
- визначити структуру веб-додатку із детальним описом кожної її частини;
- застосувати спеціальні методи розробки веб-додатку для забезпечення коректного оновлення даних при взаємодії користувача з інтерфейсом;
- забезпечити коректне відображення результатів візуалізації та моделювання даних в межах роботи веб-додатку.

Робота має як *теоретичну значущість*, оскільки в ній розглядається значна кількість прикладів щодо використання тих чи інших методів для опису даних та для використання спеціалізованих автоматизованих систем для обробки, візуалізації та моделювання даних, так і *практичну значущість*, оскільки в роботі також описано процес побудови веб-додатку, що може використовуватись у підприємстві для розв'язання певних задач для аналізу даних. Розроблений інструмент розкриває потенціал використання зазначеного методу розробки автоматизованого інструменту для роботи з даними, та може бути використаний в інших галузях, в тому числі для вирішення інших проблем та задач.

Рік виконання кваліфікаційної магістерської роботи – 2023.

Рік захисту роботи – 2023.

Ключові слова: автоматизаційний інструмент, мова програмування R, модель Adstock, ряди Фур'є, веб-додаток, системи аналізу даних, бібліотека R "Shiny", UI, Server, візуалізація даних, TSLM, LM, NLS, МНК, налаштування змінних, моделювання.

В і д г у к
про кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту «Ісititут інформаційних технологій в економіці»
освітньо-професійної програми «Економічна кібернетика і Дата Сайнс»

Єгупова К.О.
(прізвище, ініціали)
на тему **« Розробка веб-додатку для візуалізації та прогнозування даних»**
(назва теми)

1. Актуальність теми: Актуальність дослідження обумовлена сучасними умовами бізнес-середовища, де великий обсяг даних є не тільки важливим ресурсом, але й викликом для ефективного управління. Розробка автоматизованого інструменту для аналізу та моделювання даних шляхом створення веб-додатку визначається важливістю використання аналітичних засобів для здійснення прогнозування, візуалізації та управління даними. Цей інструмент не лише сприяє оптимізації рутинних завдань, але й дозволяє зробити обґрунтовані рішення на основі актуальних даних, що є критичним для успішного функціонування в умовах сучасного бізнес-середовища.

2. Позитивні риси кваліфікаційної магістерської роботи: Робота характеризується логічністю та послідовністю викладення матеріалів дослідження; автор чітко визначив мету роботи та досягнув її, розробивши оригінальний веб-додаток для аналізу даних; позитивне враження справляє самостійність автора у вирішенні технічних завдань та розробці функціоналу веб-додатку; високий рівень опису використаних технічних інструментів та математичних моделей дозволяє чітко розуміти принципи роботи розробленого рішення.

3. Наявність самостійних розробок автора: В роботі автор представив розроблений веб-додаток, який є результатом його власних зусиль та творчого підходу; детальний опис розроблених автором функцій та їх функціональності свідчать про значну самостійність у виконанні завдань.

4. Цінність теоретичних висновків та практичних рекомендацій: Отримані результати володіють як теоретичною, так і практичною цінністю, оскільки у роботі наведено детальний опис розробленого веб-додатку, який написано виключно на мові програмування R, що сприяло спрощенню процесу його розробки. Розроблений додаток є повністю робочим і може знайти застосування в різних галузях для розв'язання широкого кола задач автоматизованої обробки даних.

5. Наявність недоліків: Суттєвих зауважень немає, водночас функціонал веб-додатку можна було б розширити за рахунок збільшення кількості математичних методів, якими може послуговуватися користувач. Проте висловлені зауваження носять рекомендаційний характер та не впливають на загальну позитивну оцінку роботи.

6. Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК: Загалом, робота демонструє високий рівень науково-дослідницьких здібностей автора, його глибоке розуміння теми та є завершеною науковою роботою. Вважаю, що кваліфікаційна магістерська робота Єгупова К.О. виконана на високому науковому та професійному рівні, відповідає встановленим вимогам, може бути допущена до захисту перед ЕК та заслуговує на високу оцінку.

Науковий керівник:
к.е.н., доцент кафедри математичного моделювання та статистики Осипова Ольга Ігорівна

_____ О.І. Осипова
(підпис)
“04” грудня 2023 р.

ЗМІСТ

ВСТУП **7**

РОЗДІЛ 1 ОГЛЯД НАЯВНИХ ІНСТРУМЕНТІВ ДЛЯ АВТОМАТИЗОВАНОЇ РОБОТИ ІЗ ДАНИМИ **11**

1.1 Огляд наукових джерел щодо використання автоматизованих систем для аналізу даних 11

1.2 Огляд деяких наявних підходів до автоматизованої роботи з даними 14

1.2.1 Microsoft Excel та Google Sheets 15

1.2.2 Tableau та Power BI 19

1.2.3 Microsoft Azure та Amazon Web Services 22

1.2.4 Akkio 25

РОЗДІЛ 2 МЕТОДИКА РОЗРОБКИ ТА ОБҐРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАРІЮ ДЛЯ СТВОРЕННЯ ВЕБ-ДОДАТКУ **29**

2.1 Огляд функціоналу та можливості бібліотеки R "Shiny" у контексті розробки веб-додатку 29

2.2 Опис використаних у веб-додатку математичних моделей та методів 38

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ **48**

3.1 Структура веб-додатку. Обробка даних для використання у веб-додатку 48

3.2 Опис розроблених функцій для використання у додатку 52

3.3 Опис роботи програми. UI та Server частини 65

3.3.1 Data Research 68

3.3.2 Features Tuning 75

3.3.3 Modeling 84

3.4 Перспективи та плани щодо подальшої розробки веб-додатку для аналізу даних 91

ВИСНОВКИ **94**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **97**

ДОДАТКИ **102**

ВСТУП

Автоматизовані рішення щодо розв’язання багатьох задач стали звичним елементом людського життя, і кількість таких рішень постійно зростає. Такий вектор розвитку промисловості став особливо популярним і, фактично, визначив загальний розвиток людства, під час індустріальних революцій, під час переходу багатьох сфер діяльності від ручної праці до машинної. Із розвитком інновацій збільшувалась кількість автоматизованих рішень у кожному можливому типі промисловості. За декілька століть людство пройшло шлях від газового освітлення і парових машин до використання мікропроцесорів у кожному другому пристрої, що знаходиться вдома кожної людини.

Але розвиток технологій протягом останніх десятиліть сягає таких масштабів, що людський мозок іноді може не встигати пристосуватись до одних умов, коли вже з’являються нові. І такий значний розвиток стає можливим, в першу чергу, завдяки збільшенню обчислювальних потужностей процесорів. При чому це збільшення спостерігається не в арифметичній послідовності, а рухається по експоненті. Адже ще у 1965 році майбутній співзасновник корпорації Intel, бізнесмен та інженер Гордон Мур у своїй роботі “Cramming sea of components onto integrated circuits” описав свій відомий закон Мура [1]. Це емпіричне дослідження стверджувало, що в довготривалому тренді кількість транзисторів на комп’ютерних чіпах буде збільшуватись приблизно кожні два роки. І хоча таке припущення було висунуте стосовно найближчих 10 років, та фактично воно залишається досить близьким до правди і досі, коли пройшло вже більше 50 років. Це доводить дослідження, проведене некомерційною онлайн платформою “Our World in Data” у 2023 році, графік з якого наведений у *Додатку А.1*. Також це дослідження надає висновки щодо динаміки зміни інших спостережень, що пов’язані з технологічним розвитком. Так, зазначається, що у період з 1975 по 2009 роки обчислювальна спроможність найкращих суперкомп’ютерів демонструвала схожу динаміку,

збільшуючись у 2 рази приблизно кожних 1,5 роки, що можна побачити на графіку у *Додатку А.2* [2].

Висновком щодо наведених тверджень є те, що з кожним роком кількість розрахунків в будь-якій сфері зростатиме ще більше, можливостей для використання цих надпотужностей ставатиме все більше, і, як наслідок, кількість автоматизованих методів обробки даних збільшуватиметься, а їхня якість значно покращуватиметься. Це відбувається через бажання людей використовувати свій час та ресурси для більш творчих речей, а не на рутинні задачі, виконання яких може замінити машина.

У даній роботі розглянуто ряд прикладів, де автоматизовані системи вже можуть значно допомагати спеціалістам по роботі із даними, у тому числі для обробки, трансформації, візуалізації та моделювання даних. Розглянуто основні переваги та недоліки таких систем. Але оскільки на даний час ще не було розроблено абсолютно уніфікованого автоматизованого методу для роботи із даними, то у цій роботі приведено приклад розробки автоматизованого інструменту для роботи із даними щодо вирішення специфічних задач, що стоять перед певним підприємством. Цим інструментом є веб-додаток, що був розроблений на мові програмування R. Варто зазначити, що розробка відбувалась виключно на цій мові програмування, на відміну від багатьох інших варіантів щодо подібної розробки, коли необхідно використовувати різноманітні додаткові мови розмітки інтерфейсу, такі як HTML, CSS, або JavaScript (у якості мови для front-end розробки).

Аналіз досліджень та публікацій. У роботі розглянуто наукові роботи, що аналізують проблему використання автоматизованих систем обробки даних для специфічних задач різних індустрій. Наприклад, у роботі [3] вирішується проблема відсутності наявного програмного забезпечення з графічним інтерфейсом для відображення лонгітюдних даних з конкретним обмеженням: показом лише маргінальних ефектів певних стовпців, що потребує застосування реактивного фільтрування даних. А в роботі [4], розглядається проблема аналізу біологічних даних у постгеномну еру, яка є дуже комплексною задачею, що потребує спеціально розробленого рішення.

Актуальність теми. Розробка запропонованого веб-додатку може бути актуальною для будь-якого підприємства. Такий інструмент може допомогти розв'язати значну кількість проблем, що пов'язані із ефективним використанням ресурсів компанії. Адже у разі регулярного проведення обробки певних даних без використання консолідованого автоматизованого для цього рішення, підприємство втрачатиме час на виконання поставлених задач. Це означає недостатньо ефективне використання власних ресурсів, що, відповідно, призводить до зменшення граничного прибутку, і, як наслідок, зменшення валового прибутку компанії. Представлений у роботі веб-додаток дозволяє вирішити складні задачі, економити час для аналізу, і може бути прикладом для розробки та використання подібних систем на інших підприємствах, зокрема – в межах України.

Метою роботи є розробка веб-додатку на мові програмування R для автоматизації обробки та аналізу набору даних та вивчення потенціалу цього веб-додатку для подальшого використання в схожих задачах корпоративного сектору.

Відповідно до поставленої мети, сформульовані такі завдання:

- провести огляд наявних інструментів для візуалізації та аналізу даних та визначення можливих шляхів імплементації деяких елементів цих інструментів до розробленого веб-додатку;
- визначити структуру веб-додатку із детальним описом кожної її частини;
- застосувати спеціальні методи розробки веб-додатку для забезпечення коректного оновлення даних при взаємодії користувача з інтерфейсом;
- забезпечити коректне відображення результатів візуалізації та моделювання даних в межах роботи веб-додатку.

Об'єктом дослідження є автоматизована обробка специфічного набору даних.

Предметом дослідження є методи, технології та інструментальні засоби створення веб-додатку для автоматизації задач візуалізації та моделювання специфічного набору даних.

Використані *методи дослідження*:

- Порівняння – для огляду різних наявних підходів та інструментів для автоматизованої роботи із даними;
- Аналіз та синтез – для дослідження та формулювання висновків щодо використаних у веб-додатку даних;
- Математичне моделювання – для обробки та аналізу даних всередині розробленого веб-додатку.

Робота має як *теоретичну значущість*, оскільки в ній розглядається значна кількість прикладів щодо використання тих чи інших методів для опису даних та для використання спеціалізованих автоматизованих систем для обробки, візуалізації та моделювання даних, так і *практичну значущість*, оскільки в роботі також описано процес побудови веб-додатку, що може використовуватись у підприємстві для розв'язання певних задач для аналізу даних. Розроблений інструмент розкриває потенціал використання подібних програм для роботи з даними, та може бути застосований в інших галузях для вирішення проблем та задач автоматизованої обробки даних.

У якості *інформаційної бази дослідження* використані наукові праці вітчизняних та зарубіжних науковців, статистичні дані офіційних сайтів, документації бібліотек для мов програмування.

Робота складається зі вступу, трьох основних розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1
ОГЛЯД НАЯВНИХ ІНСТРУМЕНТІВ ДЛЯ АВТОМАТИЗОВАНОЇ РОБОТИ ІЗ ДАНИМИ

1.1 Огляд наукових джерел щодо використання автоматизованих систем для аналізу даних

В сучасному світі, де панують інформаційні технології та оптимізація кожного можливого процесу, питання автоматизації завжди стоїть на перших місцях серед можливих варіантів покращення ефективності будь-якого бізнесу. Попри те, що розробка можливих автоматизованих рішень може займати величезну кількість часу, але кожне таке рішення – це інвестиція в майбутнє, оскільки витративши значну кількість робочих годин зараз можливо зекономити ще більше цих годин пізніше. Звісно ж, подібний підхід також стосується і сфери аналізу даних, про що надалі буде йти мова.

В контексті високого обсягу обробки даних, де час є критичним фактором, автоматизація є систематизованим і важливим процесом, що звільняє ресурси для високорівневих завдань. Повторювані операції та процеси рутинного характеру піддаються автоматизації, визволяючи ресурси для стратегічного планування, творчого мислення та інновацій.

Серед головних переваг використання автоматизованих систем для роботи із даними можна виділити наступні:

1) Обробка великого обсягу даних зазвичай займає велику кількість часу, а подібні системи дозволяють ефективно обробляти, класифікувати/кластеризувати дані, зменшуючи час, необхідний для обробки цих даних. Прикладами задач, що можуть виконуватись автоматизовано, замість ручного налаштування великих датасетів, можуть бути як звичайна підготовка даних та базова очистка від викидів,

пустих чи пропущених значень із подальшою фільтрацією згідно бажаних параметрів, так і повноцінна структуризація чи класифікація датасетів, налаштування моделей, та безпосереднє моделювання даних тощо;

2) Застосування машинного навчання для автоматизованого виявлення паттернів у даних та прогнозування подій на основі накопичених даних. Машинне навчання також використовується для налаштування параметрів моделей та для самого моделювання, що також може бути реалізовано в рамках автоматизованих систем роботи із даними;

3) Збільшення точності прогнозів та зменшення ймовірності помилки через людський фактор. Завдяки можливості реалізації складних систем моделювання та автоматизованого підбору параметрів багатьох складових моделей, описані підходи дозволяють збільшити точність результатів моделювання та прогнозування. Це також досягається завдяки тому, що зменшується людський фактор під час роботи із даними, адже значну частину роботи виконує саме певна система. Але враховуючи, що будь-яка така система також була створена людиною, то треба проводити різноманітні тести та враховувати велику кількість факторів, що можуть повпливати на результат аналізу, враховуючи комплексність підходу у аспекті роботи із даними;

4) Ефективне використання ресурсів. Використовуючи подібні системи, людські ресурси можуть бути зосереджені на більш творчих та стратегічних завданнях, в той час як рутинні та ресурсозатратні операції та обчислення виконуються автоматично.

Оскільки у даній роботі буде описаний веб-додаток, створений за допомогою Shiny – пакету мови програмування R, то розглянемо деякі приклади щодо використання цього пакету.

Перш за все, розглянемо, що таке Shiny - це потужний пакет мови R, який дозволяє створювати інтерактивні веб-застосунки, використовуючи мову програмування R. Він особливо корисний для створення застосунків, які працюють з даними та включають аналіз або візуалізацію даних. Однією з великих переваг

Shiny є його легкість розробки застосунків, використовуючи лише R. Хоча є можливість використання більш традиційних мов веб-дизайну, таких як власне CSS або JavaScript, для застосунку Shiny, проте це не є обов'язковим [5].

Прикладом використання автоматизованих систем для роботи із даними є робота “An interactive R-Shiny app for quickly visualizing a tidy, long dataset with multiple dimensions with an application in clinical trial simulations for platform trials” [3], де вирішується проблема відсутності готового програмного забезпечення з графічним інтерфейсом для візуалізації лонгітюдних даних з особливим обмеженням - показу лише маргінальних ефектів певних стовпців, що вимагає реактивного фільтрування даних. Додаток є корисним при обговоренні різних результатів симуляцій, таких як оцінка різних варіантів дизайну в клінічних симуляціях з командою лікарів у інтерактивній сесії. Використання додатка значно прискорює робочий процес аналізу та узагальнення даних зі симуляційних досліджень та підготовки графіків для використання в наукових або внутрішніх цілях. Додаток може використовуватися для обговорення клінічних випробувань та узагальнення даних для студентів магістратури та докторантури, а також у будь-якому випадку, де потрібні симуляції для кращого розуміння складних проблем.

Лонгітюдними (англ. longitudinal data) вважаються дані, якщо вони відстежують один й той самий тип інформації щодо тих самих об’єктів протягом різних моментів часу. Наприклад, частина лонгітюдного набору може містити конкретних студентів та їхні стандартизовані бали на тестах протягом шести послідовних тестах [6].

Представлений R Shiny додаток для візуалізації лонгітюдних даних, може бути використаний для створення графіків, готових до публікації. Додаток особливо корисний, коли структури даних передбачають дослідження впливу різних параметрів введення на результати симуляційних досліджень. Розробка додатка продовжується, і в майбутньому планується додавання різних опцій для графіків, включення горизонтальних та вертикальних ліній, анімованих графіків, підтримки різних файлових систем та автоматичного створення R-коду.

Також серед прикладів можна розглянути роботу “shinyChromosome: An R/Shiny Application for Interactive Creation of Non-circular Plots of Whole Genomes” [4], в якій розглядається проблема аналізу біологічних даних у постгеномну еру, яка є складною задачею. Зазначається, що візуалізація даних часто використовується для передачі концепцій, комунікації нових відкриттів, узагальнення та аналізу даних, а також розвитку гіпотез. Виділяється, що Circos plots - це поширений метод візуалізації геномних даних у круговому форматі, і існує багато інструментів для їхнього створення. Автори вказують на те, що представлені засоби для лінійних відображень геному вздовж усіх хромосом менші у порівнянні з інструментами для створення кругових графіків.

Автори зазначають обмеження та особливості кожного використаного пакету R, вказуючи, наприклад, що для створення некругових діаграм цілого геному з кількома панелями даних користувачам часто доводиться самостійно встановлювати положення та розмір кожної панелі даних.

Особлива увага приділяється пакету karyoploteR, який може створювати некругові діаграми геному, але має обмеження, пов'язане з фіксованим положенням хромосом у горизонтальному положенні. Висвітлюється, що потреба у візуалізації геномних даних вимагає відмови від цього обмеження та можливості вирівнювати хромосоми вздовж вертикальної осі.

Зрештою, у роботі описується, що авторам вдалось розробити пакет shinyChromosome на базі пакету ggplot2 для візуалізації бажаних результатів щодо відображення геномів.

1.2 Огляд деяких наявних підходів до автоматизованої роботи з даними

В даній роботі детально описана реалізація розробки автоматизованої системи для візуалізації та прогнозування даних. Але будь-яка розробка подібних

систем на даний час базується на вже існуючих інструментах, які можуть виконувати різні функції щодо роботи із даними. Кожен такий існуючий інструмент стає певним референсом під час створення власної програми, оскільки хочеться взяти все найкраще з вже існуючого та додати те, що хотілось би бачити в цій системі як з точки зору розробника, так і з точки зору користувача.

Тому розглянемо певні програмні продукти, що використовуються для аналізу даних та мають автоматизовану реалізацію певних задач щодо роботи із даними.

1.2.1 Microsoft Excel та Google Sheets

Microsoft Excel (MS Excel) та Google Sheets представляють собою основних гравців на світовому ринку програмних засобів (ПЗ), що використовують електронні таблиці (табличні процесори), для роботи із даними.

Електронна таблиця (англ. electronic spreadsheet — рідко електронний аркуш) — це інтерактивний, комп'ютерний застосунок для налагодження, аналізу та збереження даних у табличному форматі. Спочатку електронні таблиці дозволяли обробляти виключно двовимірні таблиці, передусім із числовими даними, але потім з'явилися продукти, які крім цього дозволяли з'єднувати кілька таблиць для спільної роботи й опрацювання, включати текстові, графічні й інші мультимедійні елементи. Інструментарій електронних таблиць включає різноманітні математичні функції, що дозволяє виконувати складні статистичні, фінансові та інші розрахунки. Багато електронних таблиць включають вбудовану скриптову мову програмування для автоматизації типових робіт [7].

Застосунки на базі електронних таблиць зазвичай включають в себе використання таких понять, як:

- Аркуші. Кожен аркуш містить окрему електронну таблицю;

- Клітинка (комірка). Кожна клітинка має свою адресу, що складається з номерів стовпця та рядка. Адреси в свою чергу можна використовувати для різних взаємодій даних (у формулах, при написанні коду для створення макросів чи функцій тощо);
- Діапазон – це сукупність клітинок, які можна опрацьовувати як єдине ціле [8].
- Формули. Кожен програмний продукт має свій ряд формул, що можуть бути використані для взаємодії даних між собою. Ці формули можуть включати як звичайні арифметичні розрахунки, так і розрахунок статистичних показників чи навіть створення моделей (напр. функція LINEST у Microsoft Excel, яка розраховує лінійну регресійну модель за допомогою методу найменших квадратів) [9];
- Функції, підпрограми, макроси. Ці елементи можуть відрізнятись в залежності від програми, що використовується, але їхня основна суть полягає у створенні власних методів для взаємодії між комірками та діапазонами даних. Тобто, це може включати як створення власних формул, так і складні комплекси взаємодій різних елементів електронних таблиць. Зазвичай ці елементи створюються за допомогою певної мови програмування (напр. у Microsoft Excel використовується мова програмування VBA – Visual Basic for Applications – що є інтерпретованою мовою та може бути використана виключно в рамках певного програмного додатку, в який вона вбудована [10]).

MS Excel − це електронна таблиця з досить потужними математичними можливостями, в якій деякі статистичні функції є просто додатковими вбудованими формулами. MS Excel добре підходить для накопичення даних, проміжного перетворення, попередніх статистичних обчислень, для побудови деяких видів діаграм. Проте остаточний статистичний аналіз необхідно робити в програмах, які спеціально створені для цих цілей. Існують макроси- доповнення для MS Excel, що включають додаткові статистичні функції, які в основних випадках є достатніми для звичайного застосування [11].

Google Таблиці — додаток для роботи з електронними таблицями, що входить до складу безкоштовного вебпрограмного пакету програмного забезпечення, пропонованого компанією Google у межах служби Google Диск. Сервіс доступний онлайн, а також як мобільний додаток для Android, iOS, Windows, BlackBerry, а також як настільний додаток у Google ChromeOS. Інтерфейс додатка схожий з інтерфейсом Microsoft Excel, що входить до складу Microsoft Office та сумісний з форматами файлів Microsoft Excel. В режимі реального часу користувач має можливість самостійно працювати з таблицями (редагувати, форматувати тощо), а також надати спільний доступ іншим користувачам для спільної роботи. Правки відстежуються користувачем, а історія редагувань представляє зміни. Положення редактора виділяється певним для редактора кольором та курсором, а система дозволів регулює, що користувачі можуть робити. Оновленнями було представлено функції за допомогою машинного навчання, включаючи «Ознайомитись», пропонуючи відповіді на основі природних мовних питань у таблиці [12].

Ці два програмних продукти представляють дещо різний підхід до взаємодії із даними між різними користувачами одного файлу/електронної таблиці, але все ж мають між собою дуже багато спільного, та фактично можуть заміняти один одного в більшості задач.

Важливим є функціонал цих інструментів, адже кожен з них дозволяє за лічені секунди створити візуалізацію будь-яких даних у багатьох різних форматах графіків та діаграм. Також ці інструменти дозволяються використовувати величезну кількість формул для швидких обрахунків необхідних показників чи для створення нових таблиць. Вони також дозволяють створювати звідні таблиці на базі вже існуючих, що дозволяє значно швидше та зручніше фільтрувати та відображати дані у необхідному форматі. Крім того, ці програмні рішення дозволяють розраховувати результати певних математичних моделей, (напр. лінійну регресійну модель чи лінійний тренд набору значень) та використовувати ці результати для прогнозування чи аналізу даних.

В цих програмних засобах є реалізованою можливість додавання лінії тренду на графіки, що досить часто може бути корисним для базового аналізу. При чому вбудований функціонал дозволяє відображати на графіку розрахований показник R2 (коефіцієнт кореляції) між побудованим трендом та фактичними значеннями, відображати формулу, за якою побудована лінія тренду, використовувати різні підходи апроксимації до побудови лінії тренду (лінійна, експоненційна, логарифмічна, поліноміальна тощо) (рис. 1.1).

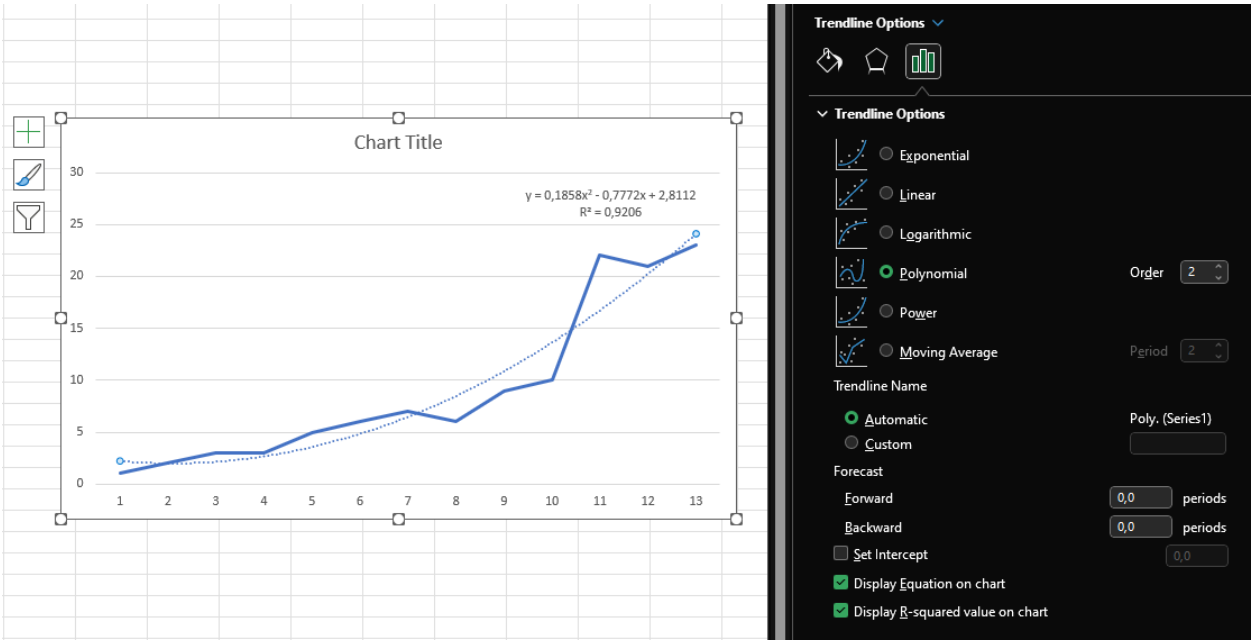


Рисунок 1.1 – Приклад використання вбудованого інструменту для побудови лінії тренду в MS Excel

Джерело: розроблено автором

Підсумовуючи, можна зазначити, що у використанні MS Excel, Google Sheets та інших подібних ПЗ є багато переваг, серед яких можна виділити те, що вони надають широкий спектр функціональності. Ці програми надають можливість використовувати велику кількість вбудованих інструментів для обробки та аналізу даних, таких як функції, звідні таблиці, графіки та діаграми тощо. Ці програми також дають можливість автоматизувати багато рутинних задач, що стосуються первинної обробки даних, візуалізації чи реформатування датасетів. Деякі подібні

інструменти дають можливість зберігати файли віддалено, а не локально, що також зменшує ймовірність втрати важливих даних. Ці інструменти є дуже швидкими у освоюванні, адже є інтуїтивно зрозумілими, і, відповідно, вони є дуже популярними по усьому світу, що завжди буде великим плюсом, адже завдяки своїй розповсюдженості користувач завжди зможе знайти відповідь практично на будь-яке своє питання чи запит.

Але більшість ПЗ, що основані на використанні електронних таблиць, мають також значні недоліки, що не дозволяють використовувати їх для багатьох задач, що пов'язані із аналізом даних. Серед головних таких недоліків можна зазначити те, що ці інструменти завжди мають обмежену масштабованість, оскільки мають обмеження щодо кількості даних, які можна використовувати в рамках кожної електронної таблиці. При цьому чим більшим буде об'єм таких даних, тим значно повільніше буде відбуватись обчислення в таких таблицях, адже в цьому форматі даних кожне значення в таблиці представлене у якості окремого елементу, на який може посилатись формула, макрос тощо. Також значним обмеженням виступає те, що подібні ПЗ програють у функціональності рішенням на базі мов програмування (таких як Python, R, C++ тощо), де, перш за все, є значно більша кількість підходів щодо обробки даних, моделювання та прогнозування, і по-друге, де рівнозначні розрахунки проводяться значно швидше, особливо якщо це стосується великих об'ємів даних.

1.2.2 Tableau та Power BI

Tableau та Power BI представляють собою лідерів ринку у сфері ПЗ для бізнес аналітики (англ. BI tools – Business Intelligence tools). Згідно з даними, що представлені на сервісі Google Trends [13], Power BI і Tableau є безумовними лідерами протягом останніх 5 років, при чому рівень зацікавленості у Tableau є

відносно стабільним протягом усіх цих років, в той час як кількість користувачів Power BI постійно збільшується, що видно на графіку (рис. 1.2).

Tableau Desktop - це програмний інструмент для візуалізації та аналізу даних, що забезпечує високий рівень інтерактивності та гнучкість в представленні великих обсягів інформації. Розроблений компанією Tableau Software, цей продукт призначений для фахівців у галузі бізнесу, аналітики та досліджень для ефективного виявлення патернів, трендів та взаємозв'язків у даних [14].

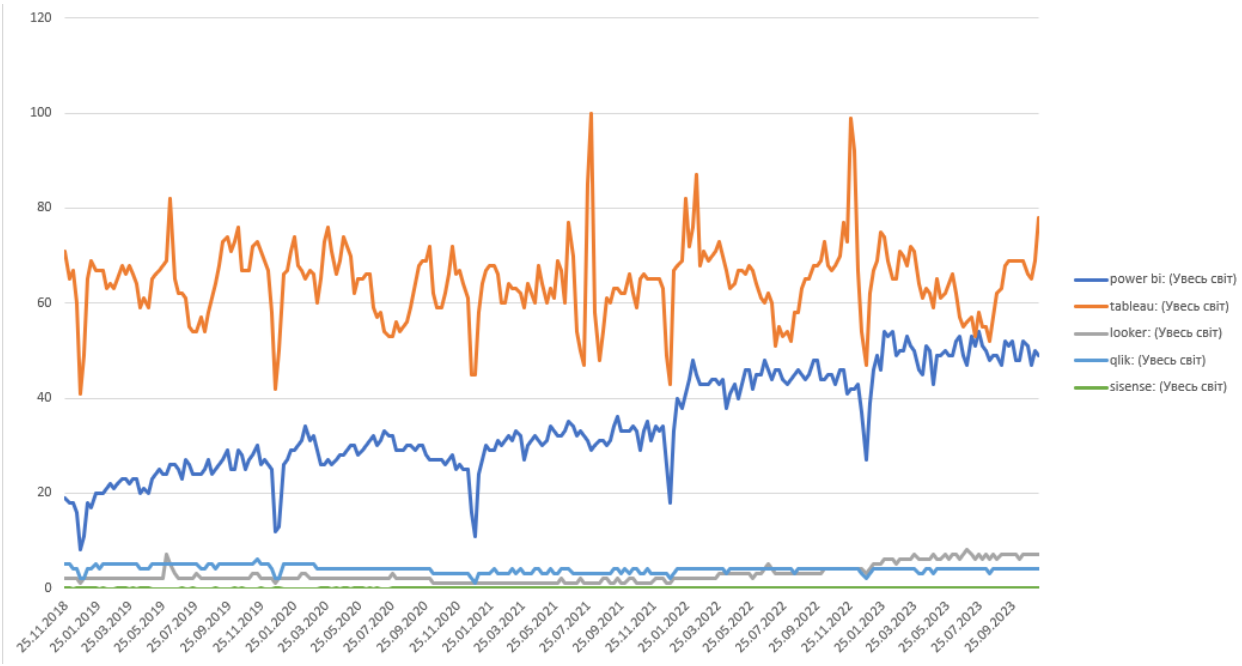


Рисунок 1.2 – Частота пошукових запитів щодо найпопулярніших BI tools.

Джерело: розроблено автором за даними [13]

Програмне забезпечення володіє високою реалізацією концепції "drag-and-drop", що робить його доступним для користувачів з різним рівнем технічної підготовки. Tableau Desktop дозволяє ефективно здійснювати аналіз та визначати важливі закономірності в наборах даних, що допомагає в прийнятті обґрунтованих бізнес-рішень та формулюванні інсайтів. Його здатність працювати з різними джерелами даних, включаючи бази даних, електронні таблиці та хмарні сервіси, робить Tableau Desktop потужним інструментом для аналізу та візуалізації даних.

Power BI – це інтегрована платформа для візуалізації та аналізу даних, розроблена компанією Microsoft. Цей інструмент постійно вдосконалюється розробниками, додаючи багато різної функціональності для можливості створення інтерактивних дашбордів, проведення аналізу даних та отримання бажаних інсайтів. Розробники приділяються особливу увагу захисту та конфіденційності даних, адже ця платформа дозволяє зберігати свої дані на серверах Microsoft, а не лише на локальному сервері [15].

Також Power BI представив інтеграцію Copilot всередину свого інструменту, що дозволяє користувачам створювати інтерактивні елементи, такі як графіки, діаграми, метрики та описовий текст даних, за допомогою штучного інтелекту (ШІ), який здатний перетворювати запити людською мовою у бажаний результат. Copilot – це хмарний інструмент на базі ШІ, що був розроблений компаніями GitHub (якою володіє Microsoft) та OpenAI для допомоги користувачам програм Visual Studio Code, Visual Studio, Neovim, JetBrains шляхом автоматичного дописання коду, чи перетворення текстових запитів у програмний код [16].

Microsoft Power BI має глибоку інтеграцію з іншими інструментами та сервісами цієї компанії, такими як Azure, Microsoft 365 тощо. Також тут є можливість використовувати застосунок корпоративно, що покращує ефективність роботи із дашбордами, доступ до яких має бути наданий декільком співробітникам. У разі використання Power BI як хмарного ПЗ, це дозволяє значно спростити процес підготовки даних до перенесення у необхідні сервіси, зменшує час на створення візуалізацій, дашбордів, інсайт-панелей тощо. Цей інструмент можна назвати розширеним та значно удосконаленим продовженням MS Excel, оскільки тут ми звісно ж можемо завантажити дані напряму з MS Excel, як і з багатьох інших форматів та джерел, а після певної обробки та об’єднання у єдину модель даних, використовувати для бажаної візуалізації та моделювання.

Тож такі інструменти, як Power BI та Tableau дозволяють використовувати великі об’єми дані та на їхній основі будувати графіки, діаграми та створювати “actionable insights”, які допомагають бізнесу досягати поставлених цілей, звертати

увагу на важливі деталі у своїй діяльності. Імплементація в роботу цих ПЗ інструментів на основі ШІ дозволяє створювати усе описане вище ще швидше та простіше, звільнюючи час на більш важливі та творчі завдання.

Але навіть враховуючи усе описане вище, подібні інструменти не завжди можуть покрити увесь спектр задач, що необхідно виконати із даними. Досить часто для реалізації певних задач необхідно створити щось нове, унікальне. При чому це може бути навіть досить проста задача, але через обмеження, що мають вбудовані можливості BI tools, такі рішення неможливо побудувати. Іноді рішення до подібних задач можуть бути розроблені в рамках MS Excel або подібних ПЗ, але в такому випадку варто зважати на описані вище обмеження щодо використання таких ПЗ.

Отже, Power BI, Tableau та інші BI tools – це потужні та функціональні інструменти, вони допомагають створити дуже необхідні бізнесам інсайт-панелі та дашборди для контролю за різними показниками та даними загалом. Як і будь-який інструмент, ці ПЗ мають свої обмеження, але на даний час розвиток багатьох BI tools знаходиться на дуже високому рівні, що навіть дозволяє використання інструментів з ШІ у якості вбудованих функцій.

1.2.3 Microsoft Azure та Amazon Web Services

Microsoft Azure – це хмарна платформа для обчислень, що належить компанії Microsoft. Вона надає можливість доступу, управління та розробку застосунків і сервісів через глобальні центри обробки даних. Також надає ряд функціональностей, включаючи програмне забезпечення як сервіс (SaaS), платформу як сервіс (PaaS) та інфраструктуру як сервіс (IaaS). Microsoft Azure підтримує багато мов програмування, інструментів та фреймворків, включаючи

власне вироблене Microsoft та програмне забезпечення та системи сторонніх виробників [17].

Microsoft Azure Machine Learning (Azure ML) – це платформа для розробки, тренування та розгортання моделей машинного навчання в хмарному середовищі Microsoft, що надає інструменти та фреймворки для розробників з можливістю створення власних послуг машинного навчання та ШІ.

За допомогою Azure ML у користувачів є можливість провести повний цикл розробки моделі з використанням машинного навчання, при цьому використовуючи потужності серверів Microsoft, а не власного комп'ютеру. Azure ML дозволяє створювати моделі машинного навчання за допомогою зрозумілого візуального інтерфейсу, а також використовувати мови програмування Python та R. Також цей інструмент дозволяє тренувати та оптимізувати моделі з використанням різноманітних алгоритмів та методів машинного навчання, управляти експериментами для аналізу та порівняння різних підходів до створення моделей, інтегруватись з іншими додатками та сервісами у хмарному середовищі, здійснювати моніторинг та управління циклом життя моделей, включаючи їхню ефективність після розгортання.

Amazon Web Services (AWS) є дочірньою компанією Amazon.com, що надає платформу хмарних обчислень в оренду приватним особам, компаніям та урядам на основі платної підписки. Існує і безкоштовна підписка, яка доступна протягом перших 12 місяців. Технологія дозволяє абонентам мати у своєму розпорядженні повноцінний віртуальний кластер комп'ютерів, який завжди доступний через Інтернет. Віртуальні комп'ютери AWS мають більшість атрибутів реального комп'ютера, включаючи апаратні пристрої (процесор, відеокарту, локальну та оперативну пам'ять, жорсткий диск або SSD-накопичувач); операційну систему на вибір; мережу; і попередньо встановлені прикладні програми, такі як вебсервер, база даних, CRM і т. д. Кожна система AWS також віртуалізує консольний ввід/вивід (клавіатура, дисплей і миша), що дозволяє користувачам AWS підключитися до своєї системи AWS за допомогою браузера. Браузер виступає як

вікно у віртуальний комп'ютер, дозволяючи користувачу входити в систему, налаштовувати та використовувати свої віртуальні системи так само, як справжній, фізичний комп'ютер. Це дозволяє їм налаштувати систему так, щоб надавати інтернет-орієнтовані сервіси та послуги своїм клієнтам [18].

Схожим інструментом до Azure ML є AWS Machine Learning. Amazon Machine Learning (Amazon ML) — це хмарний сервіс, який дозволяє розробникам будь-якого рівня кваліфікації легко використовувати технологію машинного навчання. Amazon ML надає інструменти візуалізації, які допомагають створити моделі машинного навчання (ML) без необхідності вивчати складні алгоритми та технології ML. Після того як моделі готові, Amazon ML спрощує отримання прогнозів для програми за допомогою простих API, без необхідності впроваджувати спеціальний код генерації прогнозів або керувати будь-якою інфраструктурою [19].

Amazon Machine Learning базується на тій самій високомасштабованій технології ML, яка роками використовується внутрішньою спільнотою спеціалістів з обробки даних Amazon. Сервіс використовує потужні алгоритми для створення моделей ML шляхом пошуку шаблонів у наявних даних. Amazon Machine Learning має високу масштабованість і може генерувати мільярди прогнозів щодня та обслуговувати ці прогнози в режимі реального часу та з високою пропускною здатністю. Завдяки Amazon Machine Learning немає потреби у жодних попередніх інвестиціях у апаратне чи програмне забезпечення, тому можна почати з малого та масштабувати свою програму [20].

Підсумовуючи вищезазначене про автоматизовані системи для роботи з технологіями машинного навчання, можна зазначити, що ці інструменти безумовно мають такі переваги, як їхня глибока інтеграція в рамках власних платформ (Microsoft Azure або AWS), можливість використання значної кількості різноманітних інструментів, створення власних алгоритмів та розробка моделей у візуальному середовищі тощо.

Але при цьому варто розуміти які недоліки можуть мати такі інтегровані системи розробки рішень на базі машинного навчання. Серед них особливо варто відмічати складність використання новачками у сфері аналізу даних через складність конфігурації моделей; значну вартість подібних рішень, адже кожен з таких інструментів може надавати безкоштовно лише обмежену функціональність, або надаватиме пробний період для ознайомлення із сервісом. Також серед недоліків можна назвати обмеженість обчислювальних ресурсів, оскільки треба пам’ятати, що ці сервіси використовують віддалені віртуальні машини для своїх розрахунків. А отже, ці машини можуть мати обмеженість в кількості обчислень або у кількості оперативної пам’яті (особливо це стосуватиметься безкоштовних версій подібних сервісів), що використовується, що може значно вплинути на якість та швидкість отриманих результатів моделювання.

1.2.4 Akkio

Akkio – це одна з платформ для використання машинного навчання та штучного інтелекту для аналізу даних, яка використовує чат-бот для взаємодії між користувачем та його даними. Можна попросити нейромережу провести аналіз, скласти графік чи просто швидко уточнити потрібну інформацію (рис. 1.3). Це абсолютно новий рівень взаємодії людини із даними [21].

Рисунок 1.3 – Приклад написання запиту до інтегрованого чат-боту для побудови візуалізації даних

Джерело: розроблено автором на основі [21]

Akkio є «nocode» (без використання програмного коду користувачем) рішенням, що може бути використаним для багатьох різних задач, починаючи обробкою даних із виявленням аномальних значень та очисткою бажаних частин даних, закінчуючи побудовою комплексних моделей на базі машинного навчання, із автоматичним підбором усіх необхідних параметрів, наданням результатів у зручному інтерактивному вигляді та, зрештою, оформленням усіх результатів у формат звіту або аналогу дашборду.

Серед переваг цього інструменту можна назвати те, що він дозволяє завантажувати та підключати дані з багатьох різноманітних джерел - як мережевих, так і локальних. Після підключення даних інструмент пропонує згенерувати звіт на базі даних, що були завантажені. На даний час цей функціонал працює у бета-версії, але це однозначно має величезний потенціал, оскільки якщо це буде працювати якісно, то це дозволить завантажувати необхідний набір даних у подібний інструмент, описувати звичайним текстом бажаний результат і за лічені

хвилини отримувати цей результат у вигляді графіків, текстовому описі даних та побудованих моделей прогнозу тощо.

Поява подібного інструменту стала можлива завдяки черговому збільшенню обчислювальної потужності процесорів, завдяки інноваційному підходу багатьох компаній на ринку та, безумовно, завдяки значному розповсюдженню чат-ботів, що використовують генеративний штучний інтелект (англ. GAI – Generative Artificial Intelligence), таких як ChatGPT від OpenAI, Bard від Google тощо. Розвиток моделей GAI та моделей з обробки природнього тексту (англ. NLP – Natural Language Processing) відкриває людству все нові можливості щодо взаємодії один з одним, та, в особливості, у взаємодії з комп'ютером.

Приклад Akkio показує як створення та розповсюдження одного інструменту призвело до створення іншого, зовсім нового підходу до вирішення певної задачі. І у той час, як розробники вже існуючих ПЗ запроваджують інтегровані методи «спілкування» із даними користувача за допомогою чат-ботів (напр. Tableau, Qlik, Power BI інтегрували ChatGPT у свої продукти), інші починають створювати абсолютно нові системи, в основі яких якраз лежить підхід із використанням GAI, що створює додаткову конкуренція для вже існуючих учасників ринку. Фактично, світове розповсюдження ChatGPT (як найбільш популярного чат-боту із GAI) дало величезний поштовх усій індустрії інформаційних технологій, адже кожна компанія зараз намагається використати цей інструмент у своїх цілях, для оптимізації вирішення власних задач, для створення нових підходів та рішень щодо існуючих проблем та питань тощо.

Але в той час, як використання чат-ботів із GAI допомогає вирішувати велику кількість задач значно швидше та якісніше, ці інструменти поки складно назвати тими, що могли б повноцінно замінити будь-який з вже існуючих підходів до систематизованого аналізу даних, який включає обробку даних та моделювання певних прогнозів чи аналізу щодо взаємозв'язку різних показників із розумінням бізнесової складової питання. Тому можна зазначити, що подібні до Akkio інструменти наразі дозволяють виконувати швидке завантаження даних із

різноманітних ресурсів; дає можливість автоматичного створення звітів, описавши бажаний результат інтегрованому чат-боту; надає зручність та швидкість для попередньої обробки даних; дає можливість швидкого базового аналізу даних, використовуючи групування, фільтрацію та візуалізацію даних; дозволяє проводити певне моделювання та прогнозування даних.

При цьому серед недоліків при використання подібних сервісів наразі можна зазначити такі елементи, як те, що загальний запит не часто може надати дійсно якісні та потрібні інсайти щодо даних, що були завантажені; інструмент іноді видає пусті графіки чи безінформативні таблиці тощо; існує складність та недостатня ефективність використання інструменту спеціалістами, хто не розуміється на статистиці та хоча на б базовому рівні розуміє як працює чат-бот, моделі нейронних мереж, які дані варто використовувати та як їх краще трансформувати для якісного аналізу; присутня чутливість до даних, та потреба у точному налаштуванні змінних для аналізу, бо рідко інструмент дійсно «розуміє» які змінні та у якому вигляді ми хочемо бачити; обмежена можливість налаштування моделей для аналізу та прогнозування; відносно висока вартість підписки для використання сервісу (якщо казати про Akkio), адже відсутня можливість безкоштовного використання інструменту, крім пробного періоду для ознайомлення із продуктом.

Підсумовуючи, можна зазначити, що подібний інструмент із GAI наразі навряд чи зможе замінити спеціаліста із роботи з даними, але точно зможе допомогти такому спеціалісту виконувати поставлені задачі швидше та ефективніше.

РОЗДІЛ 2
МЕТОДИКА РОЗРОБКИ ТА ОБҐРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАРІЮ ДЛЯ СТВОРЕННЯ ВЕБ-ДОДАТКУ

2.1 Огляд функціоналу та можливості бібліотеки R "Shiny" у контексті розробки веб-додатку

Призначення бібліотеки R "Shiny" та обґрунтування переваг створення веб-додатку для аналізу даних на основі R.

Раніше було описано, що Shiny - це пакет мови R, який дозволяє створювати інтерактивні веб-застосунки, використовуючи мову програмування R. Варто описати чому цей інструмент є потужним інструментом, та які він має переваги.

Оскільки програма виконує R-код на серверній частині, вона може виконати будь-який обчислення R, яке можна виконати на своєму робочому столі. Розробник може прописати код таким чином, аби застосунок обробляв набір даних на основі введених користувачем параметрів. Або, можливо, щоб веб-застосунок розраховував лінійні моделі чи методи машинного навчання на вибраних користувачем даних. У подібних та безлічі інших випадках Shiny може бути корисним.

Веб-фреймворк Shiny фундаментально дозволяє збирати значення введення з веб-сторінки, легко доступні для програми в R, і мати результати R-коду виведеними як значення виводу на веб-сторінці. У найпростішій формі Shiny-застосунок вимагає наявності функції сервера для обчислень та інтерфейсу користувача. Shiny-застосунки складаються з двох компонентів: визначення інтерфейсу користувача (“UI” частина) та серверного скрипта (“Server” частина) [22].

Сьогодні Shiny використовується практично в такій самій кількості сфер та галузей, як і сам R. Він використовується в університетах та коледжах як засіб навчання статистичним концепціям, методом зацікавити студентів у вивченні написання коду, ефектним середовищем для демонстрації новаторських статистичних методів чи моделей. Великими фармацевтичними компаніями він використовується для прискорення співпраці між вченими та аналітиками під час розробки ліків. В Silicon Valley технологічні компанії використовують його для налаштування панелей метрик у реальному часі, які включають в себе високорівневий аналіз [23].

Якщо зазначати які переваги має використання Shiny R у якості інструменту для розробки інтерактивного веб-додатку, то варто згадати наступні:

- Інтерактивність. З точки зору взаємодії з користувачем Shiny надає можливість створювати інтерактивні веб-інтерфейси, що дозволяє користувачам взаємодіяти з даними безпосередньо через веб-додаток. Це може значно полегшити взаємодію з даними, зокрема для тих, хто не володіє глибокими знаннями в області програмування чи аналізу даних;
- Швидкість і простота розробки. Shiny дозволяє швидко розробляти та тестувати ідеї, створюючи прототипи веб-додатків. Це особливо корисно на етапі визначення вимог та розробки концепцій. Використання R для програмування веб-додатків дозволяє скористатися сильними статистичними та аналітичними можливостями мови, спрощуючи розробку та забезпечуючи розширені можливості аналізу даних.
- Широкі аналітичні можливості за рахунок інтеграції з R-пакетами. Shiny легко інтегрується з існуючими R-пакетами для аналізу даних. Це означає, що користувачі можуть використовувати велику кількість статистичних і аналітичних інструментів безпосередньо в веб-додатку.
- Візуалізація даних. Shiny дозволяє створювати динамічні графіки та візуальні елементи, що дозволяє краще розуміти та представляти дані.

• Масштабованість. Веб-додатки можна легко масштабувати для обробки більшої кількості даних або забезпечення обслуговування більшої кількості користувачів.

Структура роботи веб-додатку на базі Shiny. Пояснення що таке UI та Server частини.

Кожен веб-додаток на базі Shiny R має в своїй основі дві головні функції, що взаємодіють між собою. Це функції ui() та server(), кожна з яких відповідає за свою частину роботи застосунку. Фактично, ці частини є ідентичними до частин frontend та backend, коли мова йде про роботу будь-якого веб-сайту.

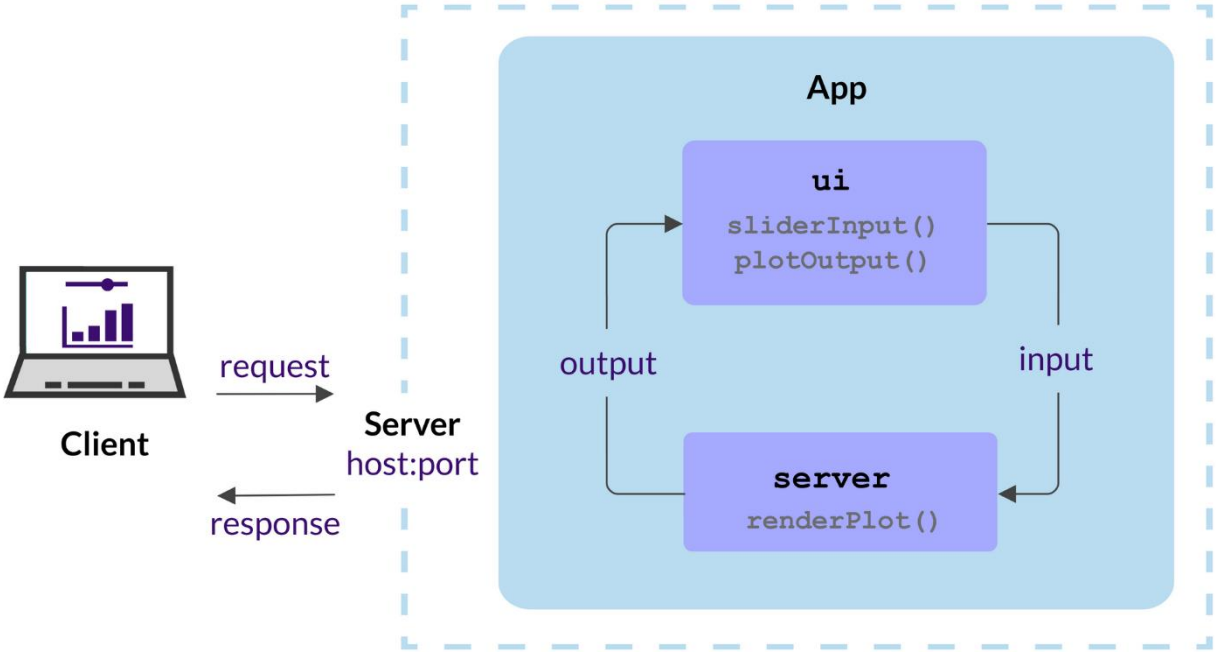


Рисунок 2.1 – Приклад базової структури веб-додатку, що розроблений на Shiny R [24]

Складова ui (англ. UI – User Interface) відповідає за користувацький інтерфейс і за те, як кінцевий споживач може бачити та взаємодіяти із додатком.

В більшості випадків написання веб-додатку починається з функції fluidPage, як зовнішнього рівня інтерфейсу. Це функція, яка буде

використовуватись для передачі важливих елементів інтерфейсу користувача, наприклад який тип макету (англ. layout – макет) хочеться використовувати.

Перед створенням веб-додатку варто розуміти який вигляд він повинен мати, яким є основний функціонал і що саме може контролювати користувач та яким чином. Це все можна задати якраз у частині ui, де перш за все варто обрати яке буде розташування елементів.

Одним із варіантів layout є sidebarLayout, який є одним із найбільш поширених макетів, і який має вузьку бічну панель ліворуч (за замовчуванням), куди зазвичай додаються елементи керування введенням (inputs), і головну панель праворуч, на якій зазвичай відображаються результати (outputs). Функції sidebarLayout містять 2 необхідні аргументи, sidebarPanel і mainPanel, які є функціями, які можна використовувати для визначення елементів керування введенням для бічної панелі та вихідного вмісту, який відображується на головній панелі, відповідно.

Іншим популярним методом створення макету веб-додатку є варіант Grid Layout. Цей тип макету надає трохи більше контролю над тим, як упорядкувати сторінку. Спочатку треба скористатись функцією fluidRow, аби додати рядки до програми, а потім використовувати функцію column, щоб створити стовпці в цих рядках. Ці функції дозволяють створювати будь-який макет сітки, включно з рівними сітками в сітках тощо.

Важливою частиною створення користувацького інтерфейсу веб-додатку є визначення елементів управління введенням (input controls). Користувачі будуть взаємодіяти з додатком, вибираючи певні значення, які будуть зберігатись в змінну, названу "input", яка буде використовуватись в розділі серверної частини додатка.

Існує 13 стандартних елементів керування, від простих прапорців (check box) до повзунків (slider). Щоб додати елемент управління до Shiny-додатка, необхідно вставити функцію елемента управління вибору всередину однієї з функцій Panel. Всі функції керування елементами вимагають щонайменше два аргументи: перший - "inputID", за допомогою якого можна отримати значення

введення користувача у коді, та другий - "label", де можна ввести опис елемента управління, видимий користувачеві.

Застосунки, які використовують sidebarLayout, часто додають елементи керування в функцію sidePanel, оскільки краще залишити більший mainPanel для відображення графіків.

Можливо налаштувати ui Shiny-додатка, написавши свій власний HTML-код. Це не є обов'язковим, і більшість людей, які використовують Shiny, не будуть писати HTML власноруч, а використовуватимуть функції R, які створять для них сторінку HTML інтерфейсу. Однак деякі розробники Shiny можуть знати HTML або працювати в команді, яка володіє цією мовою розмітки, і можуть хотіти налаштувати свій інтерфейс. Можливість написання власного HTML-коду дозволяє створювати більш персоналізовані веб-додатки.

Є 3 варіанти написання HTML інтерфейсу:

- Використовувати R-функції для написання власного HTML-коду: R-функції, такі як tags, можна додавати до вашого інтерфейсу, і, як і всі інші R-функції Shiny, вони будуть перекладені в HTML;
- Написати весь інтерфейс як HTML-документ (index.html);
- Використовувати HTML для написання HTML-компонентів, а R - для написання компонентів Shiny, використовуючи функцію htmlTemplate.

Друга складова – server - відповідає за серверну частину додатку, тобто у цій частині проходять усі розрахунки та відбувається усе те, що можна розрахувати, використовуючи звичайний скрипт в R.

Функція сервера - це місце, де відбувається більшість обчислень. Саме в функції сервера використовуються значення, які користувачі ввели через елементи керування, щоб виконувати операції, такі як обробка даних та створення виведень, таких як графіки. Ці виведення будуть зберігатися в спеціальному об'єкті, названому "output", на який можна посилатися у інтерфейсі для їх відображення в Shiny-додатку.

Корисно пам'ятати наступні 4 речі при створенні нового output додатка:

- У функції сервера треба звертатись до значень введення користувача за допомогою змінної "input$inputID";
- У функції сервера треба будувати output (наприклад, графік, таблицю, текст і т. д.) за допомогою однієї з функцій рендерингу (наприклад, renderPlot, renderDataTable, renderText і т. д.), вибираючи функцію, яка відповідає типу виводу. Обов'язково треба призначати це виведення об'єкту "output";
- У розділі UI отримується доступ до виводу за допомогою змінної "output$output_name";
- У розділі UI відображається вивід за допомогою однієї з відповідних функцій "Output" в залежності від типу виводу (наприклад, plotOutput, tableOutput, textOutput і т. д.).

Найчастіше розробники R обирають шлях створення додатка Shiny, оскільки вони хочуть, щоб їхні користувачі могли взаємодіяти з їхніми даними. Іншими словами, мета додатка Shiny полягає в тому, щоб він реагував на взаємодію користувачів з ним. З цього приводу корисно розуміти основи того, як працює реактивна модель Shiny.

Тому варто розібратись що таке реактивне програмування (reactive programming). Найчастіше код R виконується лінійно зверху вниз. Рядок за рядком програма виконується в порядку, в якому вона була написана. У реактивному програмуванні програма також залежить від подій для визначення того, як і коли код повинен бути виконаний. Події моніторяться, і коли вони виникають, код реагує на ці події.

Shiny не має інтуїтивної здатності визначати, які частини коду повинні бути перевиконані та коли. Під час написання Shiny-додатка потрібно повідомити Shiny, які частини коду повинні реагувати на події, такі як зміна значень введення в елементах керування. Оскільки бажано, щоб графіки, таблиці та інші виведення оновлювалися кожен раз, коли користувач змінює введення, будь-які вирази у розділі сервера, які залежать від змінної введення, повинні бути зроблені реактивними. Щоб зробити фрагмент коду реактивним, можна використовувати

одну з численних реактивних функцій Shiny, таких як renderPlot, renderDataTable і т. д.

Різниця між командами (commands) та рецептами (recipes) в програмуванні є однією з ключових відмінностей між двома важливими стилями програмування:

- У імперативному програмуванні (imperative programming) видається конкретна команда, і вона виконується одразу. Це стиль програмування, який відомий з скриптів аналізу: R вказується завантажити дані, трансформувати їх, візуалізувати та зберегти результати на диск.
- У декларативному програмуванні (declarative programming) виконуються більш високорівневі цілі чи описуються важливі обмеження та полагоджуються з іншими для вирішення того, як і/або коли це виконувати. Це стиль програмування, який використовується в Shiny.

Більшість часу декларативне програмування надзвичайно визволяє: описуються загальні цілі, і програмне забезпечення визначає, як досягти їх без подальшого втручання. Звісно, існує важливий недолік - іноді розробник точно знає, що йому потрібно, але він не можете зрозуміти, як це виразити так, щоб декларативна система це зрозуміла.

Однією з переваг декларативного програмування в Shiny є можливість дуже “лінивого” використання додатків. Додаток Shiny буде виконувати лише мінімальну кількість роботи, необхідної для оновлення елементів управління виведенням, які користувач може бачити в момент використання. Однак ця лінивість має важливий недолік – наприклад, якщо розробник помилиться в назві якихось змінних, то це може не викликати помилки, але це може призвети до того, що користувач не отримає потрібного output.

Лінивість Shiny має ще одну важливу властивість. У більшості R-коду можна зрозуміти порядок виконання, читаючи код зверху вниз. Це не працює в Shiny, оскільки код виконується лише за потреби. Щоб зрозуміти порядок виконання, слід звертатися до реактивного графа, який описує, як пов'язані вхідні та вихідні дані.

Реактивний граф додатку, який має один input із id = “name” та один output із id = “greeting” буде мати наступний вигляд:

Рисунок 2.2 – Реактивний граф, який показує як input та output частини поєднані між собою у найпростішому варіанті взаємодії [23]

Реактивний граф містить один символ для кожного введення та виведення, і input і output поєднуються кожний раз, коли output отримує доступ до input. Цей граф показує, що “greeting” буде перераховуватися кожного разу, коли змінюється “name”. Часто цей зв'язок описується як те, що “greeting” має реактивну залежність від “name”.

Є ще один важливий компонент, який використовується в будь-якому додатку Shiny – функцію reactive. Можна розглядати цей інструмент, як той, який зменшує дублювання в реактивному коді, додаючи додаткові вузли до реактивного графа.

Реактивні вирази важливі, оскільки вони надають Shiny більше інформації, щоб виконувалось менше перерахунків при зміні inputs, зроблюючи додатки більш ефективними, і полегшують розуміння додатка саме людям, спрощуючи реактивний граф. Реактивні вирази мають властивості як inputs, так і outputs:

- Так само, як і inputs, можна використовувати результати реактивного виразу в виведенні.
- Так само, як і outputs, реактивні вирази залежать від inputs та автоматично визначають, коли їх слід оновити.

Порівнюючи приклад простого веб-додатку із використанням реактивного програмування, та без нього, стає очевидним чому це варто використовувати: це економить час на розробку елементів, що мають використовувати неодноразово в програмі; це економить час при виконанні розрахунків програмою. У додаток до

цих переваг, розробляючи комплексні веб-додатки із багатьма inputs та outputs реактивне програмування дозволяє контролювати що і коли має виконуватись.

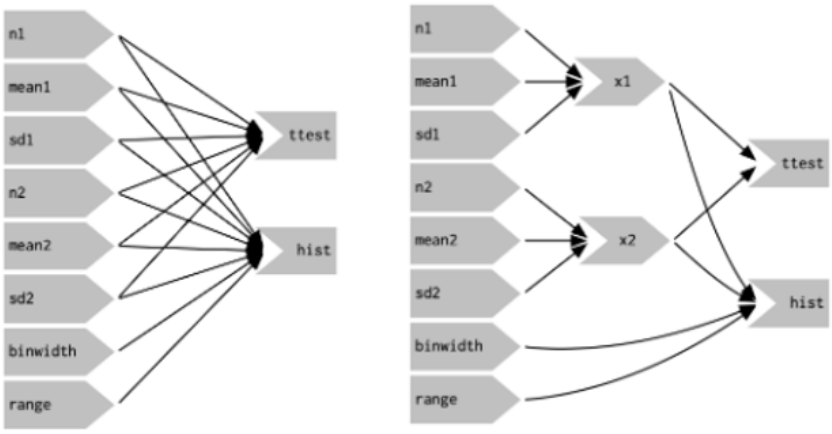


Рисунок 2.3 – Порівняння двох підходів до створення простого веб-додатку. Ліворуч – без використання реактивного програмування, праворуч – із використанням [23]

Деякими іншими важливими елементами, що використовуються під час написання коду для веб-додатку на Shiny є наступні:

- eventReactive – створює реактивний вираз, який обчислюється тільки при виникненні певної події, якими можуть бути, наприклад, натискання кнопки в інтерфейсі або зміна значення input;
- observeEvent – використовується для виконання коду при виникненні певної події, але при цьому не повертає жодного значення. Тобто результатом виконання цієї функції не можуть бути якісь значення, які можна потім ще використати.

При цьому відмінністю цих функцій від reactive є те, що вони перераховуються лише при виконанні певної події, а не при кожній зміні усіх input, що знаходяться всередині функції.

Це лише назначна частина особливостей та можливостей, які можна виконати в рамках розробки веб-додатку Shiny, але ця кількість інформації має надати достатньо інформації для поверхневого ознайомлення із принципом роботи такого веб-додатку.

Після того як додаток був створений, наступним кроком є вибір способу його розповсюдження серед інших. Серед опцій розгортання (deployment) веб-додатку є такі:

- RStudio Connect - це платформа публікації, яка може хостити багато різних типів даних, створених в R, включаючи Shiny-додатки. Вона пропонує функції, такі як розгортання за допомогою одного натискання кнопки і API для більш гнучкого робочого процесу розгортання. Це платний програмний продукт, який працює на сервері Linux і зазвичай вважається вибором для підприємств.
- Shiny Server Open Source - це безкоштовна альтернатива, яка також працює на Linux. На відміну від RStudio Connect, вона вимагає ручного розгортання Shiny-додатків, не пропонує підтримки інтеграції автентифікації з популярними постачальниками автентифікації, не може хостити інші типи артефактів даних R, таких як Plumber APIs та документи R Markdown, і обмежена одним процесом R.
- shinyapps.io - це середовище спільного хостингу, кероване RStudio. Існують як безкоштовні, так і платні рівні доступу. Ця платформа хостингу найчастіше використовується в не підприємницьких середовищах.

Усі ці середовища хостингу, по суті, діють як звичайні веб-сервери, за винятком того, що вони також можуть виконувати процес R Shiny, який необхідний для роботи Shiny-додатка.

2.2 Опис використаних у веб-додатку математичних моделей та методів

Зага́льна лінійна моде́ль (англ. LM – Linear Model) — це статистична лінійна модель, що визначається наступним рівнянням:

$$Y = XB + U, \tag{2.1}$$

де Y – це матриця що описує виміри;
X – матриця, яка може бути матриця розрахунку;

B – матриця, параметри якої, як правило, повинні бути оцінені;
U – матриця, яка містить характеристику випадкової помилки або шум.

Помилки, як правило, є наслідком багатовимірного нормального розподілу. Якщо помилки не йдуть за багатовимірним нормальним розподілом, узагальнені лінійні моделі можуть бути використані, щоб спростити припущення про Y та U.

Множинна лінійна регресія є узагальненням лінійної регресії з урахуванням більш ніж однієї незалежної змінної, а окремий випадок загальної лінійної моделі формується за рахунок обмеження кількості залежних змінних до одного. Базовою моделлю для лінійної регресії є[25]:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip} + u_i \quad (2.2)$$

Параметри моделі знаходяться методом найменших квадратів (МНК) – метод знаходження наближеного розв'язку надлишково-визначеної системи. Часто застосовується в регресійному аналізі. На практиці найчастіше використовується лінійний метод найменших квадратів, що використовується у випадку системи лінійних рівнянь. Зокрема важливим застосуванням у цьому випадку є оцінка параметрів у лінійній регресії, що широко застосовується у математичній статистиці і економетриці.

В розробленому веб-додатку буде використана множинна лінійна регресія, але враховуючи особливості побудови веб-додатку було вирішено використовувати функцію TSLM, а не lm.fit, наприклад.

TSLM (Time Series Linear Model) – лінійна регресійна модель, що також може включати в себе параметри часових рядів, таких як, сезонну складову, тренд та ряди Фур’є. Для визначення параметрів моделі регресії використовується МНК.

Звичним методом для визначення якості моделі є визначення параметру R2, тобто коефіцієнту детермінації. Цей параметр може бути розрахований, як квадрат кореляції між фактичними значеннями Y та тими, що були розраховані моделлю. Це також може бути розраховано за наступною формулою:

$$R^2 = \frac{\sum(\hat{y}_t - \bar{y})^2}{\sum(y_t - \bar{y})^2} \quad (2.3)$$

Сумування тут відбувається по усім спостереженням. Таким чином цей показник відображає пропорцію вибірки в прогнозі, що була врахована, або пояснена регресійною моделлю [26].

Як було зазначено, TSLM може включати параметри, що відповідають за часові ряди. Одними з головних таких параметрів є сезонність та тренд, які можуть бути викликані у якості частини аргументу формули як season() та trend() відповідно.

Більшість часових рядів мають тренд у своїй складовій, і використовуючи функцію trend() у якості одного з аргументів формули для TSLM можна додати лінійний тренд у якості додаткового предиктора вихідної змінної. Такий тренд буде рахуватись за наступною формулою:

$$y_t = \beta_0 + \beta_1 t + u, \quad (2.4)$$

де t = 1, ..., T.

До моделі також може бути доданий нелінійний тренд, шляхом задання точок перелому тренду у часовому періоді.

Сезонна складова може бути додана шляхом використання функції season() у якості одного з аргументів формули для TSLM, і при цьому серед налаштувань цієї компоненти можна зазначити бажані “вікна” сезону як числове значення, або ж вказати цей параметр як “periodic”, що буде означати, що сезон не має змінюватись протягом років. Але ці параметри можна не вказувати і функція самостійно підбере необхідні параметри та додасть фіктивні змінні для сезонної складової на обраному вихідному параметрі з даних.

Альтернативою використанню сезонних фіктивних змінних, особливо для тривалих сезонних періодів, є використання рядів Фур'є.

Функції тренда з періодичним характером часто є тригонометричними або можуть бути виражені лінійною комбінацією тригонометричних функцій. Отже, якщо зміни досліджуваного показника періодичні, то такі зміни доцільно представити у вигляді періодичної функції Фур'є [27].

Жан-Батист Фур'є був французьким математиком, народженим в 1700-х роках, який показав, що послідовність синусових та косинусових рядів правильних частот може наблизити будь-яку періодичну функцію. Це можна використовувати для знаходження сезонних закономірностей та більш точного моделювання [26].

Ряди Фур’є досить широко використовуються у сучасній науці завдяки своїй універсальності та широкому спектру можливих застосувань. Наприклад, у роботі [28] автор розробив математичну модель на основі перетворення Фур'є та реалізував програму для моніторингу валютних курсів у режимі реального часу. Ця програма не лише відстежує коливання валют, але й надає характеристики найпоширеніших індикаторів. Застосування цих індикаторів та аналіз їхньої кореляції з частотними характеристиками часових рядів валютних котирувань дозволяє виявляти розбіжності між ціною та індикаторами. Крім того, використання дискретного косинусного перетворення Фур'є уможливлює ефективну фільтрацію низькочастотних коливань, відомих як "ринковий шум".

У випадку використання рядів Фур'є у моделі TSLM на базі мови програмування R ці ряди використовують поєднання синусоїдних та косинусоїдних коливань. Приклад розрахунку перших рядів Фур'є, де m – це сезонний період [26]:

$$x_{1,t} = \sin\left(\frac{2\pi t}{m}\right),\ x_{2,t} = \cos\left(\frac{2\pi t}{m}\right), x_{3,t} = \sin\left(\frac{4\pi t}{m}\right), \qquad (2.5,\ 2.6,\ 2.7)$$

$$x_{4,t} = \cos\left(\frac{4\pi t}{m}\right),\ x_{5,t} = \sin\left(\frac{6\pi t}{m}\right), x_{6,t} = \cos\left(\frac{6\pi t}{m}\right) \qquad (2.8,\ 2.9,\ 2.10)$$

Якщо в даних є місячна сезонність, і використовуються перші 11 з цих предикторів, то будуть розраховані точно такі ж прогнози, як і при використанні 11 фіктивних змінних сезонності.

З термінами Фур'є часто потрібно менше предикторів, ніж з фіктивними змінними, особливо коли m велике. Це робить їх корисними для щотижневих даних, наприклад, де m≈52. Для коротких сезонних періодів (наприклад, квартальних даних) використання термінів Фур'є майже не має переваги перед сезонними фіктивними змінними.

Аргумент ‘k’ у функції fourier() вказує, скільки пар синусових та косинусових термінів включати. Максимально допустиме значення – ‘k’ = m/2 , де m - сезонний період.

Якщо використовуються лише перші два ряди Фур'є ($x_{1,t}$ і $x_{2,t}$), сезонний паттерн буде слідувати простій синусоїді. Модель регресії, що містить ряди Фур'є, часто називається гармонічною регресією, оскільки послідовні терміни Фур'є представляють гармоніки перших двох рядів Фур'є.

NLS (Nonlinear Least Squares) – це метод оптимізації, який можна використовувати для побудови регресійних моделей для наборів даних, що містять нелінійні ознаки. Моделі для таких наборів даних є нелінійними за своїми коефіцієнтами.

У NLS метою є знаходження вектора параметрів моделі β, який би мінімізував суму квадратів залишкових помилок:

$$RSS\ (Residual\ Sum\ of\ Squares) = \sum_i^n r_i = \sum_i^n (y_{obs_i} - \hat{\mu}_i)^2 \qquad (2.11)$$

де $\hat{\mu}_i$ *– змодельовані значення для і-того рядку набору даних – функція моделі параметрів вектору* $\hat{\beta}$ *регресійних змінних* x_i.

Наприклад:

$$\hat{\mu}(x_i) = f(\hat{\beta}, x_i) \qquad (2.12)$$

Існує кілька алгоритмів для ефективного оновлення вектора $\hat{\beta}$ до досягнення оптимального набору значень, яке мінімізує RSS. Найбільше поширені серед них - методи на основі області надійності, такі як алгоритм Trust Region Reflective, алгоритм Левенберга–Марквардта та алгоритм Dogbox [29]. Деякі з цих алгоритмів підтримуються у мовах програмування R, Python.

Метод NLS широко використовується у сучасній науці та бізнес-рішеннях. Наприклад у роботі [30] було вивчено гібридний підхід, який комбінує Генетичне Програмування (GP) та локальний пошук для еволюції моделі та оптимізації параметрів в задачах символічної регресії, і проведено порівняння цих методів із GP NLS та OSGP NLS. Метод виявився ефективним, забезпечуючи покращення узагальненої здатності моделі та високу точність прогнозів на тестовому наборі. Незважаючи на певний зріст часу обчислень через локальний пошук, переваги у якості рішень, швидкості та розмірі моделі роблять цей гібридний підхід привабливим в контексті задач символічного аналізу.

Крім згаданих складових щодо специфіки роботи цієї моделі варто також зазначити те, як саме цей підхід реалізований у рамках мови програмування R – у бібліотеці stats [31], функцією nls(). Для забезпечення функціонування цього підходу в рамках R користувач може зазначати досить велику кількість параметрів та мати більший контроль над змінними (наприклад, на відміну від згаданих лінійних моделей LM чи TSLM), серед яких є такі:

- formula – текстовий параметр - формула, що має використовуватись моделлю для проведення апроксимації параметрів та знаходження потрібних коефіцієнтів. Ця формула має включати назви самих параметрів (згідно з назвами у наборі даних) та самі параметри, що мають додатково бути створені та призначені у бажаному порядку. Наприклад, якщо у наборі даних є змінні Sales, Price, Penetration, то варіантом того, як прописати формулу у такому випадку буде наступний варіант: “Sales ~ a1 * Price + a2 * Penetration”, де ai – коефіцієнти, що

будуть проходити оптимізацію моделлю, та за допомогою яких буде відбуватись розрахунок параметра Sales;

- data – набір даних, що мають використовуватись функцією;
- start – дані формату list або vector, що мають містити стартові значення для кожного з коефіцієнтів моделі (якщо взяти минулий приклад, то це мають бути значення для коефіцієнтів a1 та a2). Цей параметр є досить важливим, оскільки при неадекватному призначення стартових точок модель може видати помилку;
- algorithm – текстовий параметр – потрібен для вибору алгоритму, що буде використовуватись у моделі для підгону коефіцієнтів моделі. За замовченням використовується алгоритм Гаусса-Ньютона. Серед можливих варіантів також є: “plinear” – для алгоритму Голуба-Перейри для підходу із застосуванням часткового МНК; “port” – для алгоритму “nl2sol” з бібліотеки Port.
- control – аргумент, що можна задати, якщо потрібно обмежити кількість ітерація для навчання моделі, змінити рівень толерантності (для забезпечення відносної конвергенції критеріїв моделі), не переривати виконання коду у разі виникнення попередження (warning) в результаті виконання моделі тощо;
- lower, upper – дані формату list або vector, що мають містити нижні та верхні межі для кожного з коефіцієнтів моделі. Ці межі можуть бути використані лише при виборі алгоритму “port”, інакше будуть проігноровані під час компіляції.

Останні 2 аргументи будуть особливо важливими у контексті розробки веб-додатку, що описаний у даній роботі, оскільки дані, що будуть використовуватись включають в себе параметри, що іноді мають бути обмежені для забезпечення економічної обґрунтованості та логічності результатів моделі з точки зору бізнесу.

Adstock – функція для якіснішої оцінки впливу маркетингових інструментів на ринкові показники.

Advertising adstock - це продовжений або запізнений ефект реклами на користувацьку поведінку споживачів. Adstock є важливою складовою моделей “marketing-mix model”. Термін "adstock" був створений Саймоном Бродбентом [32].

Adstock - це модель того, як “відгук” на рекламу будується і зникає у споживачів на ринку. Реклама намагається розширити споживання двома способами: вона нагадує та навчає споживачів. Вона нагадує споживачам на ринку з метою впливу на їх моментальний вибір бренду (англ. immediate brand choice) та навчає їх збільшувати впізнаваність (англ. brand awareness) та значущість (англ. brand salience) бренду, що полегшує вплив реклами на вибір бренду в майбутньому. Adstock - це математичне виявлення цього поведінкового процесу.

Теорія adstock базується на припущенні, що показ телевізійної реклами збільшує awareness у думках споживачів, впливаючи на їхнє рішення про покупку. Кожен новий контакт з рекламою збільшує awareness, і ця awareness бренду буде вищою, якщо були недавні контакти, і нижчою, якщо їх не було. У відсутності подальших показів adstock з часом розпадається до незначних рівнів. Вимірювання та визначення adstock, особливо при розробці моделі marketing-mix modeling, є ключовою складовою визначення ефективності маркетингу. Існують два аспекти рекламного adstock:

- ефект ослаблення (англ. decay) або ефект запізнення (англ. lagged effect);
- ефект насичення (англ. saturation) або ефект спадної віддачі (англ. diminishing returns) [33].

Основна теорія adstock полягає в тому, що показ телевізійної реклами збільшує усвідомленість споживачів на ринку, що призводить до продажів. Кожен новий контакт з рекламою підвищує awareness на новий рівень. Ефект розпаду adstock з часом зменшує awareness до базового рівня, якщо не зменшити цей ефект decay новими показами реклами. Цей ефект ослаблення може бути математично змодельований і зазвичай виражається в термінах "піврозпаду" реклами. "Півтора тижня піврозпаду" означає, що для того, щоб awareness реклами зменшилась до половини свого поточного рівня, потрібно два тижні. Припускається, що кожен рекламний матеріал має унікальний піврозпад. Деякі академічні дослідження вказують на діапазон піврозпаду від приблизно 7–12 тижнів, тоді як практики індустрії зазвичай повідомляють про піврозпад від 2 до 5 тижнів, з середнім

значенням для брендів FMCG на рівні 2,5 тижні. Піврозпад adstock може бути оцінений через lag-модель з розподіленими затримками із змінної валових рейтингів ТБ (GRP), використовуючи метод найменших квадратів [33].

Базове формулювання моделі Adstock може бути представлене у такій формулі:

$$A_t = T_t + \lambda A_{t-1}, t = 1, \dots, n, \tag{2.13}$$

де A_t – це Adstock в момент часу t;

T_t – це значення показнику рекламної активності в момент часу t;

λ – це коефіцієнт, що означає ефект decay, або ефект запізнення.

Цей підхід до побудови моделі Adstock буде реалізований в описаному в цій роботі веб-додатку. Також будуть запропоновані методи для визначення найкращого коефіцієнту λ.

Збільшення обсягу реклами збільшує відсоток аудиторії, до якої реклама була запропонована, і, отже, збільшує попит, проте лінійне зростання показу реклами не має аналогічного лінійного ефекту на попит. Зазвичай кожне додаткове збільшення реклами призводить до прогресивно меншого ефекту на збільшення попиту. Це і є ефектом насичення від реклами, або ефектом спадної віддачі. Насичення відбувається лише вище певного порогового рівня (англ. Saturation Threshold), який може бути визначений аналізом Adstock. Приклад такого аналізу представлено на рис. 2.4.

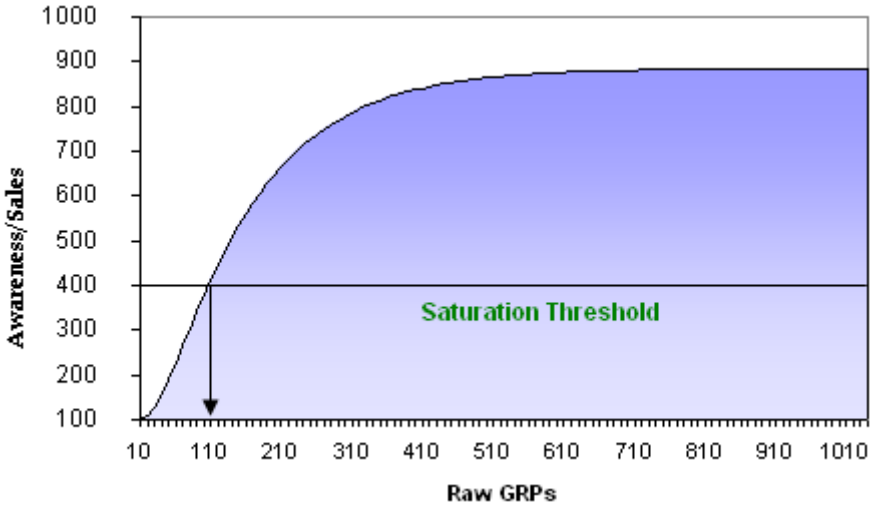


Рисунок 2.4 – Приклад застосування ефекту насичення, або ефекту спадної віддачі [33]

Модель Adstock активно використовується у сучасному бізнесі, оскільки будь-яка компанія, яка рекламує свою продукцію, бажає знати справжній ефект від своєї активності в медіа.

РОЗДІЛ 3
ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ

3.1 Структура веб-додатку. Обробка даних для використання у веб-додатку

Описаний в роботі веб-додаток був розроблений в цілях використання для більш ефективної взаємодії з даними, що надсилаються з боку клієнтів. Основними задачами, що ставились під час розробки додатку, були такі:

- Можливість автоматизованої обробки та зведення великих наборів даних в єдину таблицю;
- Зручна та швидка фільтрація даних і приведення цих даних у потрібний формат, що використовується у компанії. Можливість вивантаження відфільтрованих та очищених даних у окремий Excel файл формату “.xlsx”;
- Візуалізація певних ринкових та медіа показників з можливістю швидкого порівняння цих даних між різними брендами, що зазначені у наборі даних;
- Можливість використання відфільтрованих даних у моделюванні для оцінки впливу медіа активності на ринкові показники та для прогнозування бажаних показників на майбутні періоди, використовуючи регресійний аналіз та аналіз часових рядів;
- Можливість оцінки результатів моделювання в рамках роботи веб-додатку;
- Можливість додавання змінних до відфільтрованих даних, які б допомагали описувати вихідну змінну та покращували ефективність моделювання.

Опис датасету, для якого розроблявся веб-додаток

Датасет, представлений у роботі є прикладом даних, що отримує компанія від деяких клієнтів для подальшої обробки. Ці дані містять ринкові та медіа дані самого клієнта та брендів конкурентів. Варто зауважити, що усі представлені в даній роботі приклади використання веб-додатку матимуть зашифровані дані, оскільки реальні значення не мають права бути розповсюдженими у відкритий доступ згідно із NDA (Non-Disclosure Agreement), підписаного із відповідною компанією-постачальником цих даних.

Дані представлені у потижневому та помісячному розрізі та у якості дати є три числові змінні – Рік, Місяць і Тиждень.

У файлі з даними містяться два листи – Sales та Penetration. Перший з них представлений у потижневому розрізі та містить дані продажів та активності на ТБ відповідних брендів. Другий лист представлений у помісячному форматі та містить дані пенетрації відповідних СКЮ на ринку України. Другий лист містить лише змінні Року, Місяцю, Бренду, СКЮ та Пенетрації.

Серед категорійних змінних є такі:

- Бренд – назва бренду, щодо якого представлені дані у відповідному рядку;
- Категорія – відповідає за назву категорії, до якої належить той чи інший бренд;
- Бренд/Конкурент – бінарна категорійна змінна, що вказує на те, чи є даний бренд основним брендом, чи він є конкурентом до основного бренду у відповідній категорії;
- CORE/OTHERS – бінарна категорійна змінна, яка вказує на те, чи належить бренд до основних конкурентів – “CORE”, або ж є другорядним конкурентом – “OTHERS”. Якщо це рядок основного бренду, а не конкуренту, то значення також вказується як “CORE”;
- СКЮ – (англ. SKU – stock keeping unit) – назва, ідентифікатор конкретного продукту бренду. Усі дані в базі представлені у розрізі SKU.

Також в датасеті є такі числові змінні:

- Упаковки – кількість проданих упаковок відповідного СКЮ у відповідний період (тиждень);
- Грн – продажі СКЮ у грн;
- TRP Dir 5, TRP Dir 10, TRP Dir 15, TRP Dir 20, TRP Dir 30, TRP Other 5, TRP Total – змінні, що відповідають за активність відповідного СКЮ на ТБ. “Dir” вказує на те, що мова йде про пряме розміщення реклами, “Other” – про те, що це може бути інший тип реклами (міжпрограмне розміщення, спонсорські плашки, пряме спонспорство тощо), “Total” – є сумою усіх типів прояву за відповідний період часу;
- Penetration – показник пенетрації відповідного СКЮ у відповідний період (місяць). Пенетрація в даному датасеті відповідає за показник проникнення відповідного товару у магазини/аптеки/АЗС тощо (відповідно до категорії товару).

TRP (Target Rating Points) – показник, що використовується для оцінки кількості та якості розміщення певної медіа активності, і використовується для оцінки медіа проявів у ТБ, друкованих виданнях, соціальних мережах і тд. Для розрахунку TRP необхідно знати, або, принаймні, мати естимацію розміру аудиторії, на яку таргетується певна реклама, та загальний розмір аудиторії, на яку реклама була показана. Зрештою, отримуємо таку формулу розрахунку TRPs [34]:

$$TRPs = \frac{\text{Impressions on advertisement within target audience}}{\text{Total intended target audience}} \times 100 \qquad (3.1)$$

Також у медіа індустрії існує показник GRP (Gross Rating Points), який відразняється від розрахунку TRP тим, що не враховує розмір цільової аудиторії, а розраховує рейтинги на генеральній популяції.

Як результат, GRPs краще використовувати для оцінки awareness, в той час як TRPs краще використовувати для оцінки охоплення реклами (англ. ad reach).

В даній роботі використовуватимуться саме TRPs, виходячи з наявних у наборі даних змінних.

Опис основних частин веб-додатку. Призначення кожної з частин та ідея при їх створенні.

Структура веб-додатку з точки зору користувача (ui) є розділеною на 3 основні частини:

1) Data Research – частина веб-додатку, де користувач має можливість відфільтрувати дані за бажаними параметрами, обрати тип візуалізації даних, та, власне отримати готові графіки щодо обраного типу візуалізації;

2) Features Tuning – частина веб-додатку, де користувач також має можливість відфільтрувати дані за певними параметрами, але з певними обмеженнями, у порівнянні до 1 частини веб-додатку. Після обрання усіх бажаних фільтрів користувач може обрати які саме змінні він бажає налаштувати та додати до отриманого набору даних. Зрештою, отриманий набір даних буде використаний також у 3 частині веб-додатку для проведення моделювання даних;

3) Modeling – частина веб-додатку, де користувач має можливість обрати яку саме модель використати для проведення аналізу, та налаштувати цю модель за деякими параметрами. За обраними параметрами будується модель та відображаються її результати по тренувальній та тестовій частинах даних (співвідношення яких також можна налаштувати). Результати представлені у вигляді графіку, таблиці похибок та загальному виводу результатів відповідної моделі методами R.

З точки зору розробки цей додаток розділений на 3 скрипт-файли формату ".R":

- ui.r – складається з опису користувацького інтерфейсу. Основу складають функції щодо розмітки вигляду веб-додатку (вбудовані HTML функції та спеціальні функції бібліотеки Shiny). В середині функцій розмітки знаходяться відповідні функції для використання елементів вводу (inputs) та виводу (outputs), а також додаткові функції для відображення текстової та візуальної інформації;
- server.r – потрібен для виклику функції server (якби ui та server не були розділені на різні файли – веб-додаток, відповідно, мав би дві головні функції для

запуску додатку, такі як ui() та server()). В цій частині коду програма отримує усі ввідні та вивідні значення, що були зазначені в файлі ui, виконує усі необхідні розрахунки відповідно до вхідних параметрів, та генерує (рендерить) усі необхідні outputs для відображення у веб-додатку.

- helpers.r – допоміжний скрипт-файл, в якому відбувається підключення усіх необхідних бібліотек, зазначається директорія із місцем розташування файлу із даними, відбувається завантаження та базова обробка даних. Також у цій частині веб-додатку відбувається ініціалізація додаткових параметрів, значень та власних функцій, що використовуються у двох основних частинах веб-додатку.

3.2 Опис розроблених функцій для використання у додатку

Adstock functions.

У цій роботі були описані основні причини застосування методу Adstock коли мова йде про оцінювання впливу медіа активності на ринкові показники певного бренду. І також було зазначено про значну важливість коректної оцінки коефіцієнту Adstock, що буде використовуватись для забезпечення так званих ефектів ослаблення або запізнення. Тому для визначення цього коефіцієнту для кожної конкретної ситуації було розроблено декілька функцій, які можуть допомогти автоматично підібрати найкращий коефіцієнт для кожного типу медіа та формату розміщення.

Перш за все було створено функцію ‘adstock’, за допомогою якої і виконуються розрахунки для кожного окремого формату медіа. Ця функція буквально є інтерпретацією формули (номер adstock формули). Аргументами функції є ‘X’ та ‘koeff’, де ‘X’ – це вектор значень медіа активності, на яку потрібно застосувати метод Adstock, а ‘koeff’ – це, відповідно, коефіцієнт, який застосовується у циклі. На виході функції повертає числовий вектор значень такої

ж довжини як вхідний вектор із розрахованим коефіцієнтом Adstock. Надалі ця функція буде використовуватись у кожному з підходів для вибору найкращого можливого коефіцієнту, та за допомогою цієї функції будуть відбуватись перетворення значень медіа активності у датасеті, що завантажений до веб-додатку.

Перш ніж більш детально розглянути кожну із функцій для підбору коефіцієнту Adstock варто зазначити, що аргументи усіх функцій є однаковими, а саме:

- ‘df’ – датасет, що подається функції, з якої беруться усі необхідні значення для розрахунку коефіцієнтів;
- ‘lowest_k’, ‘highest_k’ – межі коефіцієнтів, що використовуються для розрахунку коефіцієнтів. По замовченню для нижньої межі використовується значення 0.1, оскільки в ідеї створення та використання моделі Adstock лежить те, що кожен медіа прояв має певний вплив на покупців із затримкою, або у формі накопичення знання. І тому мінімальне значення 0.1 дозволить застосувати хоча б мінімальний ефект накопичення до кожного з форматів медіа активності. Для максимального значення по замовченню встановлено значення 0.8, оскільки використовуючи більший коефіцієнт можна отримати занадто довготривалий ефект впливу медіа прояву на поведінку споживача, що може призвести до неадекватної оцінки впливу медіа на ринкові показники. Оптимальним вважається значення коефіцієнту Adstock в межах 0.3-0.6;
- ‘y’ – текстовий аргумент, який відповідає за назву змінної з датасету ‘df’, по якому функція має розраховувати відповідні метрики для вибору найкращого коефіцієнту.

Було розроблено 3 функції, кожна з яких дещо відрізняється підходом до розрахунку оптимального значення коефіцієнту Adstock. Розглянемо кожну з них більш детально.

‘adstock_cor_func’ - Adstock Correlation Function. Ця функція має у своїй основі ідею того, що найкращим коефіцієнтом буде той, використовуючи який

відповідна медіа активність буде мати найвищий коефіцієнт кореляції із залежною змінною ‘y’. Поетапно ця функція виконує такі етапи:

1) Використовуючи вбудовані в мову програмування R регулярні вирази (англ. regular expressions), а саме, функцію ‘grep’, визначаються та зберігаються назви змінних у датасеті ‘df’, які мають використовуватись у якості показників медіа активності. У якості аргументу паттерну до функції grep було подано значення “TRP|TV|Outdoor|Digital|Radio”, що означає що будь-яке часткове співпадіння назви змінної із будь-яким значенням, що розділене знаком «|», буде означати, що до цієї змінної має застосовуватись функція Adstock. Серед значень, що використовуються у цьому регулярному виразі, зазначені найбільш популярні можливі варіанти медіа активності, тому дотримуючись умови використання одного із цих значень у назві потрібних змінних, користувач зможе бути впевненим, що до відповідних змінних був застосований коефіцієнт;

2) Створюється пустий список (list) для зберігання розрахованих коефіцієнтів, під назвою ‘adstock_res’;

3) Створюється послідовність значень – ‘seq_ad’ – за якими функція надалі буде проводити вибір найкращого коефіцієнту. Послідовність створюється, використовуючи задані в аргументах функції ‘lowest_k’ та ‘highest_k’;

4) Наступним етапом є використання циклу для розрахунку та зберігання найкращого значення коефіцієнту для кожної визначеної змінної датасету. Це реалізовано наступним чином:

a. Цикл використовує збережені у 1 пункті назви змінних;
b. Для кожної змінної виконується перевірка умови ненульової суми значень відповідної змінної у даних;
c. Якщо умова виконана, то:
    I. Створюється тимчасовий датасет ‘temp_t’ на базі вхідного ‘df’, що спочатку складається тільки зі змінних ‘y’ та однієї відповідної змінної медіа активності;

II. Після цього виконується додатковий цикл, у якому у якості ітератора ‘k’ використовується послідовність ‘seq_ad’. Цикл розраховує значення змінної медіа із використанням кожного можливого коефіцієнту Adstock (за допомогою описаної раніше функції ‘adstock’) та додає такі змінні до створеного на початку цього підпункту тимчасового датасету ‘temp_t’;

III. Після виконання внутрішнього циклу відбувається розрахунок коефіцієнтів кореляції для кожної пари ‘y’:adstocked_AD(‘k’);

IV. Зрештою обирається найвища серед розрахованих кореляція із змінною y та зберігається у список ‘adstock_res’, що був створений у пункті 2;

V. У список додається елемент із назвою медіа показнику, якому присвоюється найкращий розрахований коефіцієнт Adstock;

d. Якщо умова не виконана, то у список ‘adstock_res’ додається елемент NA;

5) Зрештою функція ‘adstock_cor_func’ повертає результат у форматі list, де кожна назва відповідає назві медіа показнику, і відповідні присвоєні значення відповідають найкращому коефіцієнту Adstock, що були розраховані за допомогою функції.

‘adstock_cor_lag_func’ – Adstock Correlation Function with Lag. Ця функція використовує ту ж ідею, що реалізована у функції ‘adstock_cor_func’, але відрізняється тим, що застосовує лагову затримку до кожного медіа показника, що відповідає логіці самого принципу Adstock, оскільки певна медіа активність може мати не тільки накопичений ефект, але це також може відбуватись із певною затримкою. Розглядаючи шлях реалізації цієї функції розпочнемо з етапу 4c, оскільки до цього вони є ідентичними:

4) Внутрішній цикл:

c. Якщо умова (пункту 4b) виконується, то:

I. Створюється тимчасовий порожній датасет ‘temp_t’ зі змінними ‘coef’, ‘lag’, ‘cor’, кожна з яких відповідає за застосований коефіцієнт Adstock, лагову затримку застосовану до медіа показника та кореляцію розраховану між медіа змінною та цільовою змінною ‘y’;

II. Додається внутрішній цикл, де ітератором ‘i’ є послідовність коефіцієнтів – ‘seq_ad’;

III. Всередину циклу додається ще один цикл, де ітератором ‘lag_n’ є послідовність можливих застосованих лагів до показника медіа активності. Ця послідовність включає в себе 3 можливих значення лагу – 0, 1, 2;

IV. У подвійному внутрішньому циклі відбувається розрахунок усіх можливих комбінацій коефіцієнту ‘i’ та лагової затримки ‘lag_n’, що застосовується до відповідної медіа змінної. По кожній такій комбінації розраховується коефіцієнт кореляції з цільовою змінною ‘y’, і результати відповідно записуються у таблицю ‘temp_t’;

V. Після заповнення таблиці ‘temp_t’, вона відсортовується за спаданням за показником коефіцієнту кореляції, і ця таблиця зберігається до списку ‘adstock_res’;

5) Зрештою функція ‘adstock_cor_lag_func’ повертає результат у форматі list, де кожна назва відповідає назві медіа показнику, а кожна змінна списку містить в собі відсортовану таблицю значень, яку за бажання можна переглянути після виконання функції. Якщо потрібно лише використати результати розрахунків до набору даних, то можна просто використати коефіцієнти, що збережені у верхньому рядку кожної таблиці.

‘adstock_lm_maxTRP_func’ – Adstock with Linear Model and Max Ad coefficient. Ця функція має ідею підбору коефіцієнту таким чином, аби при побудові лінійної регресії із застосуванням такого коефіцієнту до певного медіа показнику, похибка прогнозу була мінімальною, і при цьому коефіцієнт у моделі регресії відповідного медіа показнику має бути максимально можливим. Тобто для

зберігання коефіцієнту Adstock за цим методом має виконуватись одразу дві ці умови. Розпочнемо з етапу 4c, оскільки до цього етапи є ідентичними із попередніми описами функцій:

4) Внутрішній цикл:

c. Якщо умова (пункту 4b) виконується, то:

I. Відбувається ініцілізація змінних для зберігання найкращого коефіцієнту Adstock – ‘best_rate’, найнижчої похибки прогнозу за регресійною моделлю – ‘lowest_error’, найвищого коефіцієнту показнику медіа активності в регресійній моделі – ‘max_coef’;

II. Створюється цикл із ітератором ‘seq_ad’;

III. Всередині циклу відбувається створення тимчасового датасету, який ініціалізується на основі ‘df’, але де залишається тільки та змінна медіа активності, що використовується згідно зовнішнього циклу. Далі відбувається розрахунок з використанням функції ‘adstock’ та заміною змінної з медіа активністю на розрахований із Adstock. Відбувається побудова базової моделі LM із методом МНК у своїй основі;

IV. На розрахованій моделі розраховується похибка RMSE;

V. Відбувається одночасна перевірка двох умов. Перша – що розрахована похибка менше за ті, що були розраховані раніше по зовнішньому циклу. Друга – що коефіцієнт медіа активності у побудованій моделі є вищим за той, що був збережений раніше. Якщо умова виконується – відповідний коефіцієнт Adstock зберігається у змінну ‘adstock_res’. Якщо ні – пропускається.

5) Зрештою функція ‘adstock_lm_maxTRP_func’ повертає результат у форматі list, де кожна назва відповідає назві медіа показнику, і відповідні присвоєні значення відповідають найкращому коефіцієнту Adstock, що були розраховані за допомогою функції.

Використання цих функцій реалізовано в інтерактивному вигляді у веб-додатку та буде описано детальніше у наступних розділах.

Save to Excel function.

Під час роботи веб-додатку виникала проблема у відображенні даних програмою у вигляді таблиці, а також під час зберігання цих даних у файл формату ".xlsx". Ці проблеми виникали через використання специфічного формату дати "yearweek" з пакету "tsibble" [36]. Цей формат використовується для використання потижневих даних і приклад такого формату виглядає як "2023 W25". Проблема цього формату полягає в тому, що він не підтримується пакетом "DT", а саме функцією 'renderDT', яка використовується у веб-додатку для створення і відображення інтерактивних таблиць. Схожа ситуація із тим, коли потрібно вивантажити датасет, де є змінна у форматі "yearweek" в MS Excel. У такому випадку ця змінна може бути відображена у вигляді числових значень у вивантаженому файлі. Ці числові значення можна перевести у формат дати, адже вони по суті відображають кількість секунд від початкової заданої дати в MS Excel, а саме 1 січня 1900 року [35]. Але оскільки цей веб-додаток може використовуватись тими, хто не знайомий з такими деталями щодо роботи MS Excel, то було вирішено розробити функцію, яка буде брати змінну періоду у форматі "yearweek" та створювати на базі неї 3 змінні – Рік, Місяць, Тиждень.

Розроблена функція 'save_to_excel' вирішує описану проблему. У якості аргументу функції тут використовується датасет 'df'. Сама функція виконує переформатування 'df' у формат "tibble" для забезпечення коректної роботи із цими даними. Після цього відбувається додавання описаних вище змінних – Year, Month, Week – у числовому форматі, з округленням до 0 знаків після коми (що необхідно для більш зручного відображення у веб-додатку, наприклад). Після цього змінна періоду з 'df' видаляється і функція повертає перероблений датасет.

NLS set function.

Функція 'nls_set' була розроблена з метою оптимізації процесу використання функції 'nls' з пакету "stats" [37], яка має досить багато аргументів для налаштування моделі, і деякі з цих аргументів мають дуже специфічні та відносно незручні формати для використання.

Аргументами функції ‘nls_set’ були визначені такі параметри:

- ‘df’ – датасет, щодо якого відбувається налаштування параметрів для подальшого виклику функції ‘nls’;
- ‘y_name’ – аргумент у текстовому форматі, який вказує на назву цільової змінної з датасету ‘df’;
- ‘cols_to_exclude’ – аргумент у текстовому форматі, що буде використовуватись у функції ‘nls_set’ для обрання змінних, що не мають бути включеними у модель NLS, використовуючи регулярні вирази. Це потрібно для того, аби при кожній побудові моделі NLS не потрібно було прибирати певні змінні з датасету, а це все вже буде прописано у описаній розробленій функції. За замовченням тут стоять значення, що унеможливлюють використання змінних дати та деяких категорійних змінних, що можуть траплятись у датасеті. Користувач функції може прописати сюди бажані змінні, що не мають використовуватись у моделюванні;
- ‘cols_to_apply_min_val’, ‘cols_to_apply_max_val’ – аргументи, що мають формат векторів, де кожна назва у векторі має відповідати частині назви змінних з датасету ‘df’, а кожне присвоєне значення відповідає за мінімально та, відповідно, максимально можливі значення, що мають бути присвоєні відповідним коефіцієнтам під час навчання моделі. Для пошуку потрібних змінних у датасеті також використовуються регулярні вирази.

Не описуючи детально програмний код функції, зазначимо основні етапи, що виконує функція ‘nls_set’ під час її виклику:

1) Підготовка даних. Використовуючи вхідний датасет ‘df’ функція очищує дані для подальшого використання. Це включає такі елементи, як видалення змінних у текстовому форматі та видалення змінних у числовому форматі, які містять лише нульові чи пусті значення. Ці елементи реалізовані за рахунок функції ‘sapply’ з базового пакету R. Ця функція дозволяє застосовувати інші функції до кожного елементу вектору, списку чи датасету, що наданий ‘sapply’ у якості аргументу [38]. Такий спосіб роботи із даними значно пришвидшує

виконання коду та дозволяє скоротити його об’єм, у порівнянні із використанням звичайних циклів, таких як ‘for’ або ‘while’. Також на етапі підготовки даних відбувається видалення зазначених у аргументі ‘cols_to_exclude’ змінних;

2) Визначення стартових значень, нижніх та верхніх меж для кожного коефіцієнта моделі NLS. На цьому етапі виконується призначення необхідних мінімальних та максимальних значень відповідних змінних. Для цього відбувається перебір кожної назви змінної датасету ‘df’, перевірка умови того, що вона містить в собі одне з імен аргументів ‘cols_to_apply_min_val’ або ‘cols_to_apply_max_val’, та у разі задоволення умови виконується призначення відповідних меж до потрібної змінної датасету. Стартові значення коефіцієнтів змінних призначаються шляхом перевірки умов щодо нижніх та верхніх меж відповідних змінних, для запобігання призначення стартового значення, що не буде відповідати обмеженням;

3) Останнім етапом відбувається визначення формули, що буде використана в моделі NLS. Важливим зауваженням є те, що функція ‘nls’ використовує не звичайний вигляд формули, як наприклад у функції ‘lm’ або ‘TSLM’, де формула може мати вигляд “Sales ~ Price + Penetration”. У випадку ‘nls’ аналогічна формула має виглядати так: “Sales ~ b + a1*Price + a2*Penetration”. Це дуже обтяжує можливість зручної побудови цієї моделі, тому це також стало важливим фактором для розробки функції ‘nls_set’;

4) Вихідним значенням функції є список, кожен елемент якого одразу можна використовувати для побудови моделі ‘nls’. Цей список включає такі параметри: формула, підготовлений датасет, стартові значення, верхні та нижні межі коефіцієнтів, алгоритм (який встановлений як “port” – див. Розділ 2.2.3), назви змінних, додатковий опціональний показник ‘control’ (де задані значення максимальної кількості ітерацій для побудови моделі, мінімальний рівень толерантності та обмеження щодо переривання виконання функції у разі виникнення “warning” – див. Розділ 2.2.3).

В результаті, розробка цієї функції дозволила значно оптимізувати процес використання функції ‘nls’ та дозволила реалізувати додаткові можливості при розробці веб-додатку, про що буде описано далі.

R^2 function for NLS та Test Data Errors function.

При використанні функції ‘nls’ виникла також проблема із оцінкою точності прогнозування. Базовими можливостями оцінки якості побудови моделей лінійної регресії та лінійної регресії із використанням аналізу часових рядів є розрахунок показника R^2, який автоматично відображається під час виклику функції ‘summary’ до розрахованої моделі. Але подібний підхід не реалізований для функції ‘nls’, тому було вирішено розробити функцію ‘r.sq_func’ для розрахунку цього показника, для можливості порівняння моделей лінійної регресії із моделлю NLS.

R^2 , або коефіцієнт детермінації – це оцінка, що надає інформацію про якість побудованої моделі. В контексті регресійних моделей R^2 відображає наскільки якісно лінія регресії апроксимує фактичні значення [39].

Для розрахунку цього показнику можуть бути використані різні підходи. У будь-якому варіанті відображення цієї метрики в основі буде лежати ідея порівняння дисперсії залишків змодельованих значень і дисперсії залишків різниці фактичних та середніх значень. Наприклад, R^2 може бути розрахований за формулою:

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2} \quad (3.2)$$

Інтерпретація цієї формули була реалізована у вигляді програмного коду. Аргументом функції ‘r.sq_func’ має бути розрахована модель ‘nls’, з якої функція отримує розраховані моделлю значення, залишки згідно цих розрахунків, кількість предикторів. Далі розраховуються значення RSS та TSS, і, відповідно, значення самого R^2.

Основна проблема R^2 полягає в тому, що він не завжди враховує кількість змінних у моделі, і, отже, може підвищити своє значення при додаванні нових змінних навіть у випадках, коли це не поліпшує модель.

Відкоригований R^2 (R^2_{adj}) вирішує цю проблему шляхом врахування кількості змінних у моделі. Він використовується як корекція для R^2, і його формула виглядає так:

$$R^2_{adj} = 1 - \frac{(1-R^2)\times(n-1)}{n-k-1} \quad (3.3)$$

де n – це кількість спостережень;

k – кількість незалежних змінних у моделі.

Основна ідея полягає в тому, що R^2_{adj} буде менше ніж R^2, якщо додавання нової змінної не допомагає покращити модель, і більше ніж R^2, якщо нова змінна є корисною для пояснення дисперсії в залежній змінній.

У функції ‘r.sq_func’ також були реалізовані ці розрахунки. Зрештою, розроблена функція після її виклику повертає список, де значення ‘r.sq’ дорівнює значенню R^2, ‘adj.r.sq’ – значенню R^2_{adj}.

Після розрахунку будь-якої моделі обов’язковим також є її оцінка на тестовій вибірці для розуміння того, як ця модель може себе показувати при реальному прогнозуванні, а не лише при моделюванні тренувальної частини. Для такої оцінки зазвичай використовуються готові функції з різних пакетів мов програмування. В даному випадку була написана досить проста функція ‘errors_func’, яка виконує розрахунок певних бажаних похибок на тестовій частині даних за допомогою відповідних функцій з пакету “fabletools” [41] та повертає ці значення у зручному табличному вигляді із розшифруванням кожної похибки (напр. “MAPE (Mean absolute percentage error, %)”).

У якості аргументів функція має отримати фактичні значення цільової змінної датасету (actual data) та залишки після розрахунку прогнозів (residuals).

Використовуючи ці значення функція ‘errors_func’ розраховує та повертає такі похибки, як RMSE (Root Mean Squared Error), MAPE (Mean Absolute Percentage Error) та MAE (Mean Absolute Error) у табличному вигляді з відповідними підписами.

Fourier term function

У Розділі 2.2.2 було описано можливість використання рядів Фур’є у регресійній моделі TSLM. Головним питанням, що виникло під час їхнього використання, став підбір правильного параметру ‘k’, що відповідає за порядок ряду Фур’є у розумінні комбінування синусоїдних та косинусоїдних термінів, що суттєво впливає на якість розрахованої моделі.

З огляду на це, було вирішено розробити функцію ‘fourier_set’, яка дозволяє автоматично підібрати найкращий можливий параметр ‘k’ для конкретного набору даних. Основною ідеєю при розробці функції була побудова моделі TSLM для кожного можливого параметру ‘k’ та тестування результатів моделі на певному наборі даних. При чому користувач функції може обирати на якому саме наборі даних розраховувати похибки прогнозу для вибору найкращого параметру ‘k’ – на тому ж, тренувальному наборі даних, або ж використовувати підготовлений тестовий набір, завдяки чому протестувати як певний порядок рядів Фур’є буде показувати себе на даних, які не були надані моделі до цього.

Параметр ‘k’ в рамках функції ‘fourier_set’ може приймати значення від 1 до 26, оскільки дані, що використовуються у веб-додатку мають потижневий формат. У Розділі 2.2.2. було описано, що максимальним значенням ‘k’ має бути значення k = m/2, де m – це сезональний період. Тобто при використанні потижневих даних, коли середня кількість тижнів в році = 52.18 [42], ‘k’ має дорівнювати не більш ніж 52.18/2 = 26.09. Оскільки ‘k’ має бути цілим значенням, то округляємо до 26.

Аргументами функції ‘fourier_set’ є такі параметри:

- ‘df’ – датасет, який буде використовуватись при побудові моделі TSLM, та для якого, відповідно, треба порахувати параметр ‘k’;

- ‘test_df’ – датасет, на якому буде відбуватись тестування результатів застосування певного параметру ‘k’. За замовчуванням цей показник встановлений як “0”, що означає, що функція буде використовувати тренувальний набір також і як тестовий. Якщо присвоїти цьому аргументу певний набір даних, то вже він буде використовуватись як тестовий. Цей аргумент дозволяє більш гнучко використати підбір параметру ‘k’;
- ‘y’ – цільова змінна у датасеті ‘df’;
- ‘cols_to_exclude’ – текстовий аргумент, що відповідає за назви, які у разі використання у датасеті ‘df’ будуть виключені і не використовуватимуться для побудови і тестування моделей в рамках функції ‘fourier_set’.

Основними етапами, що виконує функція ‘fourier_set’ під час її виклику є такі:

1) Створення пустої таблиці, в яку пізніше функція запише результати її виконання. Видалення змінних з датасету ‘df’, згідно з аргументом ‘cols_to_exclude’, використовуючи регулярні вирази. Визначення необхідного тестового датасету;

2) Виконання циклу, всередині якого відбувається перебір параметру ‘k’ та побудова моделей TSLM з використанням функції ‘fourier’ та відповідним ‘k’. Після побудови кожної моделі розраховуються похибки прогнозу RMSE та MAPE на обраному тестовому датасеті. Результат кожної побудови функції та розрахунку похибок записується у створену у 1 пункті таблицю із зазначенням поточного значення ‘k’;

3) Після виконання циклу функція зберігає найкраще значення ‘k’ згідно з найменшою розрахованою похибкою RMSE. На виході функція повертає значення найкращого параметру ‘k’ і таблицю результатів, де можна переглянути похибки RMSE та MAPE для кожного параметру ‘k’.

Недоліком використання цієї функції є її відносна повільність (5-7 секунд), оскільки під час її виконання програмний код розраховує 26 моделей TSLM, і по кожній з них розраховує прогноз, похибки і зберігає результат. І все це відбувається

всередині циклу, а не векторним розрахунком, де усе це могло б бути розраховано одночасно.

Але навіть враховуючи цей недолік, розроблена функція ‘fourier_set’ надала можливість використовувати значно менше часу для налаштування моделі TSLM із використанням рядів Фур’є.

3.3 Опис роботи програми. UI та Server частини

Під час запуску описаного надалі веб-додатку першим чином виконується, так звана, підготовча частина, а саме – відбувається підключення бібліотек, назначається робоча директорія з файлами (working directory), виконується програмний код файлу “helpers.R”, де відбувається завантаження необхідних даних з файлу “Sales_Base.xlsx”, обробка цих даних, створення необхідних змінних та функцій для використання у інших файлах із програмним кодом додатку.

Серед основних бібліотек, що використовуються у даному веб-додатку є такі:

- ‘shiny’ – фреймворк для розробки веб-додатків використовуючи програмний код мови R [23];
- ‘shinythemes’, shinyWidgets” – додаткові бібліотеки для роботи із розширеним функціоналом та додатковими можливостями налаштування веб-додатку на базі бібліотеки ‘shiny’ [42];
- ‘readxl’ – бібліотека, що робить зручним використання даних з MS Excel всередині R. Порівнюючи з іншими існуючими бібліотеками, такими як ‘gdata’, ‘xlsx’, ‘xlsReadWrite’, – ‘readxl’ не має зовнішніх залежних елементів, її легко встановити та вона може використовуватись на усіх операційних системах. Ця бібліотека була створена для роботи з табличним даними [43];

- ‘fpp3’ – бібліотека, що включає в себе низку інших важливих бібліотек (‘tibble’, ‘dplyr’, ‘tsibble’, ‘tidyr’, ‘ggplot2’ та ін), що використовуються для аналізу, прогнозування та візуалізації даних. Опис цієї бібліотеки з багатьма підходами щодо прогнозування та роботи із даними описано у книзі авторів бібліотеки – Rob J Hyndman and George Athanasopoulos: “Forecasting: Principles and Practise” [26];
- ‘forecast’ – бібліотека, яка надає методи та інструменти для відображення та аналізу прогнозів однофакторних часових рядів, включаючи експоненційне згладжування за допомогою моделі просторових станів та автоматичне моделювання використовуючи модель ARIMA (Autoregressive integrated moving average) [44];
- ‘plotly’ – графічна бібліотека R, яка робить інтерактивні, готові до публікації графіки [45]. Будь-який графік, що створюється за допомогою цієї бібліотеки, використовує бібліотеку з мови програмування JavaScript – “plotly.js”. Завдяки інтеграції цієї бібліотеки у ‘plotly’, користувач може створювати інтерактивні графіки на базі вже побудованих, наприклад, за допомогою бібліотеки ‘ggplot2’ [46].

Також у файлі “helpers.R” відбувається завантаження та обробка датасету, що використовується у веб-додатку. Цей датасет завантажується спочатку у вигляді двох окремих таблиць: Sales та Penetration. Після цього назви у кожній таблиці змінюються, аби вони були однаковими та зручними у використанні. Наприклад, під час завантаження у таблиці Sales дані ТБ активності мають назви “TRP Dir 10"”, “TRP Dir 5"” і тд. Ці назви містять пропуски і незручні у використанні позначки секунд “"”. Тому назви цих змінних були замінені на більш зручні, вигляду “TRP_dir_10” тощо.

Після зміни назв у таблицях відбувається універсалізація деяких значень шляхом приведення їх до одного реєстру, а саме у змінних Brand та Core_or_Others усі значення замінюються на такі ж, але тільки з використанням верхньому реєстру (великих букв).

Після цих налаштувань відбувається об’єднання двох датасетів в один за типом об’єднання “left join”, де першою таблицею виступає таблиця продажів. Це обумовлено тим, що у таблиці пенетрації можуть міститись дані, що не співпадатимуть значеннями ідентифікаторів (якими в даному випадку є бренд та СКЮ), а головною таблицею все ж є та, що містить дані продажів.

Після об’єднання у загальну таблицю відбувається фінальна очистка даних від пустих значень. Це відбувається шляхом застосування конструкції “select(where(~!(all(is.na(.)) || all(. == 0))))”, завдяки якій будь-які змінні, що містять тільки пусті чи нульові значення будуть видалені. Після цього відбувається заміна усіх пропущених числових значень на нульові, і усіх текстових (категорійних) – на “NA”.

Потім ініціалізуються деякі унікальні та спеціальні значення із завантажених даних для подальшого використання у файлах “ui.R” та “server.R”. Серед таких значень є: значення мінімальних та максимальних років з загального датасету, унікальні значення усіх категорійних змінних, мінімальні та максимальні значення потижневих дат із датасету та послідовність усіх можливих тижнів, та деякі спеціальні змінні, що пізніше будуть описані у детальному описі функціонування програми.

Останньою частиною файлу “helpers.R” є ініціалізація розроблених функцій, які вже були описані у минулих розділах.

Перед тим як переходити до опису кожної складової веб-додатку варто зазначити про ту частину програми, яка встановлюється для усього додатку загалом – це тема, заголовок та загальна структура веб-додатку у частині ui. Тема була встановлена за допомогою функції ‘shinytheme’ і був обраний варіант “flatly”. Також за допомогою вбудованої функції “HTML” були змінені деякі параметри відображенні веб-додатку і був доданий заголовок під назвою “Analysis App”, і заголовок у вікні браузера – “Shiny App”. Загальна ж структура додатку була визначена як ‘fluidPage’, для забезпечення максимальної кількості налаштувань розташування елементів у додатку. І кожна окрема частина веб-додатку, що буде

надалі описана, знаходиться всередині загальної функції ‘navbarPage’, в окремих блоках під назвою ‘tabPanel’. Така структура дозволяє розділити важливі елементи веб-додатку в окремі блоки: “Data Research”, “Features Tuning” та “Modeling”.

3.3.1 Data Research

Перша частина веб-додатку надає користувачу можливість візуалізувати потрібні для нього показники з загального датасету. Вигляд першої сторінки веб-додатку представлений на рис. 3.1.

Рисунок 3.1 – Загальний вигляд частини ‘Data Research’ у розробленому веб-додатку

Джерело: розроблено автором

Загальний опис роботи частини з точки зору користувача. Перш за все розглянемо ліву сторону екрану, де зазначені усі параметри, що користувач може обрати або змінити. Серед них у верхній частині є блок для фільтрації даних (“Data Filterting”), у якому, відповідно, можна відфільтрувати дані за бажаними параметрами. Серед таких фільтрів представлена можливість фільтрації по даті (по

“Year” та “Week”) та по категорійним змінним (“Main or Competitors”, “Core Brand or Other”, “Category”, “Brand”, “SKU”). Нижче фільтрів додана кнопка для завантаження відфільтрованого датасету (“Download Table”). Ще нижче знаходиться блок для взаємодії з графіками, що знаходяться праворуч. В цьому блоку є два основних параметри для налаштування: “Select type of 1 Plot” відповідає за перемикання верхнього графіку між двома різними типами, “Select Values for plot (max = 3)” потрібний для вибору бажаних змінних, що користувач хоче побачити на верхньому графіку праворуч. Отже, верхній графік відповідає за відображення ринкових показників, а нижній – за відображення показників рекламної активності на ТБ. Також у правій частині (output) зверху користувач може обрати варіант що саме відображати – графік (“Plot”), або таблицю із відфільтрованими даними (“Table”). Більш детально кожну складову буде описано нижче.

Реалізація UI частини. Завдяки використанню структури додатку ‘fluidPage’ у веб-додатку була можливість використати більш кастомізовану структуру розташування елементів. Таким чином inputs у лівій частині сторінки мають більш стислий вигляд завдяки використанню функцій ‘column’ всередині ‘sidebarPanel’. Усі inputs в цій частині веб-додатку, як і в усій роботі, реалізовані завдяки вбудованим функціям типу ‘…Input’. Наприклад для ввідного елементу для фільтрації по рокам та тижням був використаний input типу slider, де в налаштуваннях вказано використання двох елементів для створення діапазону значень. Також серед input було використано формат ‘radioButtons’, який потрібен для вибору одного з доступних користувачеві значень, у даному випадку – для вибору типу графіку з ринковими даними. Для інших елементів input у цій частині веб-додатку був використаний input типу selectize, що є розширеною версією ‘selectInput’ та дозволяє гнучкіше налаштовувати сам елемент input, а також дозволяє користувачу більше варіантів для взаємодії із елементом. Для виводу у веб-додаток графіків та таблиці із даними були використані спеціальні елементи виводу із вбудованими можливостями взаємодії з ними. Так, для виводу графіку використовується функція ‘plotlyOutput’ з пакету ‘plotly’, а для виведення таблиці

– функція ‘dataTableOutput’ з пакету ‘DT’. Для перемикання між графічним і табличним виводом даних використовується метод ‘tabsetPanel’, що дозволяє розділяти сторінку на декілька окремих частин.

Реалізація Server частини. Структура частини “Data Filterting” у частині Server складається з декількох послідовних функцій ‘reactive’, ‘observe’ та різних типів output.

Перш за все відбувається фільтрація даних, що отримуються з файлу ‘helpers.R’, у функції ‘reactive’, що просвоєна імені ‘obs_t_f1’, по змінним “Year”, “Week”, “Main_or_C”, “Core_or_Others”, “Category”. Фільтри по “Brand” та “SKU” були винесені в окремі послідовні частини, оскільки була реалізована можливість зміни цих input завдяки використанню поєднання функцій ‘observe’ та ‘updateSelezctizeInput’, що дозволяє оновлювати ці input при зміні зазначених у ‘obs_t_f1’ показників. Як можна побачити на рис. 3.2, при зміні параметру “Category”, можливі для вибору бренди також були автоматично змінені.

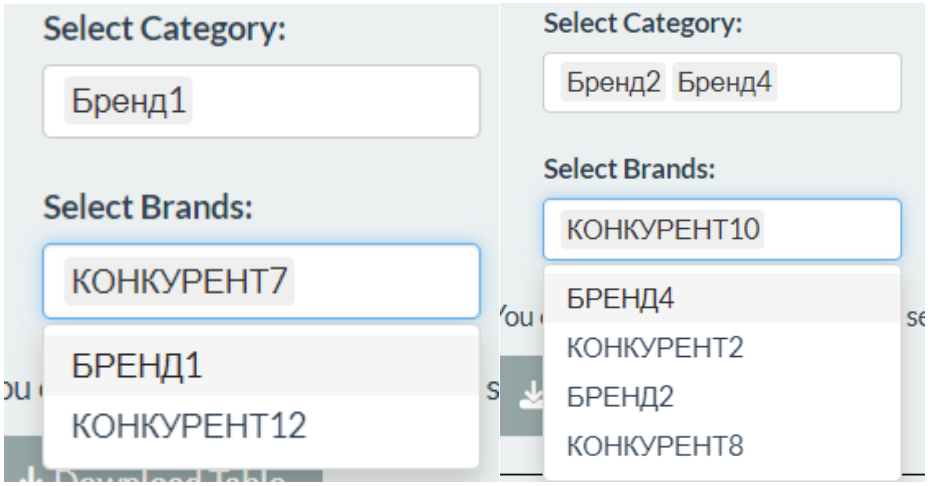


Рисунок 3.2 – Відображення реалізації оновлення даних у input при зміні інших параметрів

Джерело: розроблено автором

Також у ‘obs_t_f1’ була прописана валідація виконання подальшого коду для забезпечення коректності оброблення даних. Для цього у цій функції була

прописана обов'язковість вибору кожного з елементів, по яким відбувається фільтрація даних. Наприклад, якщо користувач не вибере категорію, то замість помилки у обробці даних буде виведено текст із проханням обрати якусь категорію для продовження роботи із веб-додатком (рис. 3.3).

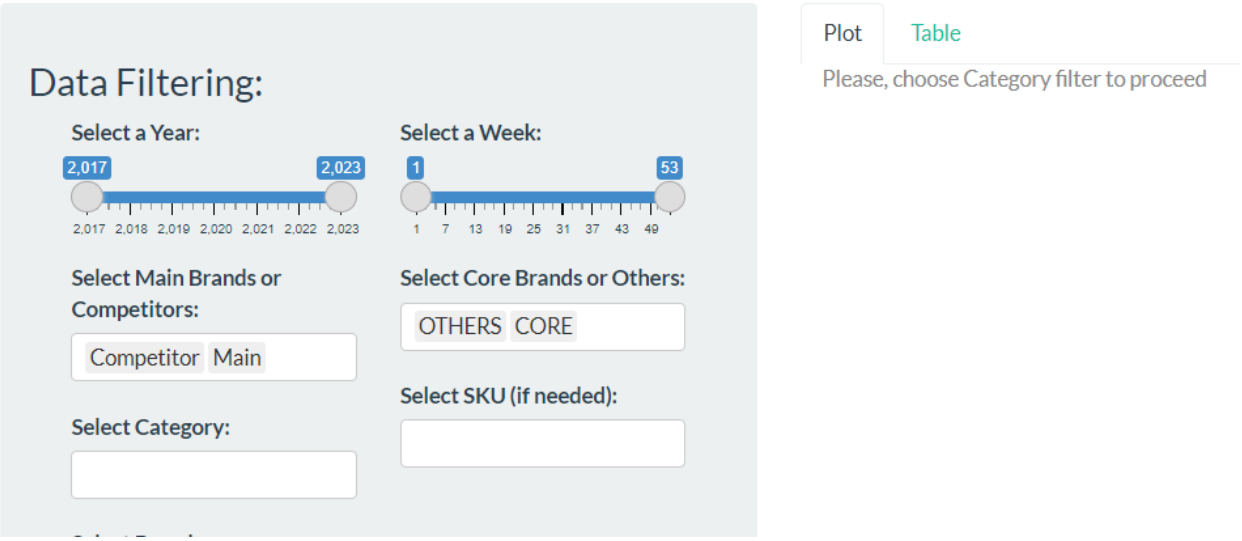


Рисунок 3.3 – Приклад реалізації валідації щодо використання необхідних input

Джерело: розроблено автором

Наступним етапом є функція, де виконується фільтрація даних за змінною "Brand". Тут також відбувається валідація щодо того, аби користувач обов'язково обрав певний бренд, щоб код зміг зробити відповідну фільтрацію. Після цього відбувається оновлення input – для елементу вибору змінної "SKU".

Наступною функцією виконується перевірка умови нульової довжини input, який визначає вибір SKU користувачем. Це потрібно для того, щоб користувач не обов'язково обирав цей параметр, оскільки іноді потрібно візуалізувати весь бренд, а не окремо певний SKU. У тому разі, якщо довжина обраного input є більше 0, що означає, що користувач обрав певний SKU а не залишив його порожнім, то у цій частині відбувається фільтрація по цьому показнику, а потім групування даних за роком, тижнем та усіма категорійними показниками. Якщо довжина дорівнює 0 то

відбувається тільки групування. Потім відбувається розділення даних на частину ринкових показників та пенетрації окремо, оскільки пенетрація представлена у помісячному форматі, і якщо одразу зробити сумування даних по групам, то можна отримати некоректні результати. Тож на цьому етапі відбувається сумування ринкових показників і додається змінна ціни SKU (або ж середньозважена ціна бренду, якщо параметр SKU залишився пустим), також окремо зводяться дані пенетрації, але у якості функції агрегації використовується не сумування, а знаходження максимального значення, оскільки пенетрація – це відносний показник, що може приймати значення від 0 до 100%. Далі розділені дані об'єднуються в єдину таблицю, в залежності від того, чи був обраний SKU у input, – тільки по даті та бренду, або також і по SKU. Таким чином отримуються відфільтровані та згруповані дані по обраним користувачем параметрам, які тепер можуть бути відображені у вигляді таблиці, або у вигляді графіків.

Отриману таблицю тепер можна передати до функції для рендеру таблиці, що була описана вище. Також необхідно реалізувати візуалізацію цих даних. Для цього було використано дві функції 'reactive', одна з яких будує верхній графік (з ринковими показниками), другий – нижній (із графіком рекламної активності на ТБ).

Перша з цих функцій перш за все проводить валідацію щодо того, чи обрав користувач змінні, що треба вивести на графік, та тип графіку для відображення. Оскільки користувач може обрати декілька типів графіку, а також декілька різних брендів або SKU для відображення, то треба було враховувати усі ці нюанси під час створення графіку. Для початку отримані з минулих функцій дані переводяться у довгий формат, використовуючи функцію 'pivot_longer', для більш зручної роботи із легендою та різними лініями на одному графіку. Далі відбувається перевірка двох умов щодо типу використаного графіку. Якщо користувач обирає варіант графіку "Full Period", то будується графік по обраним користувачем змінним, при чому якщо користувач обирав певні SKU, то колір змінюється в

залежності від SKU, а якщо не обирав – то в залежності від обраних брендів. Ця умова потрібна для того, аби дані коректно відображались, оскільки може бути обрано декілька брендів, але при цьому SKU буде обрано тільки для одного бренду, і в такому разі на графіку буде виведено неправильні дані. Також для цього графіку використовується функція ‘facet_wrap’, яка розділяє один графік на декілька, якщо було обрано декілька різних змінних для відображення. На рис. 3.1 було показано як графік відображає лише один обраний бренд і жодного SKU. Тому на рис. 3.4 представлено приклад того, як графік може відобразити одразу декілька обраних SKU по декільком різним змінним.

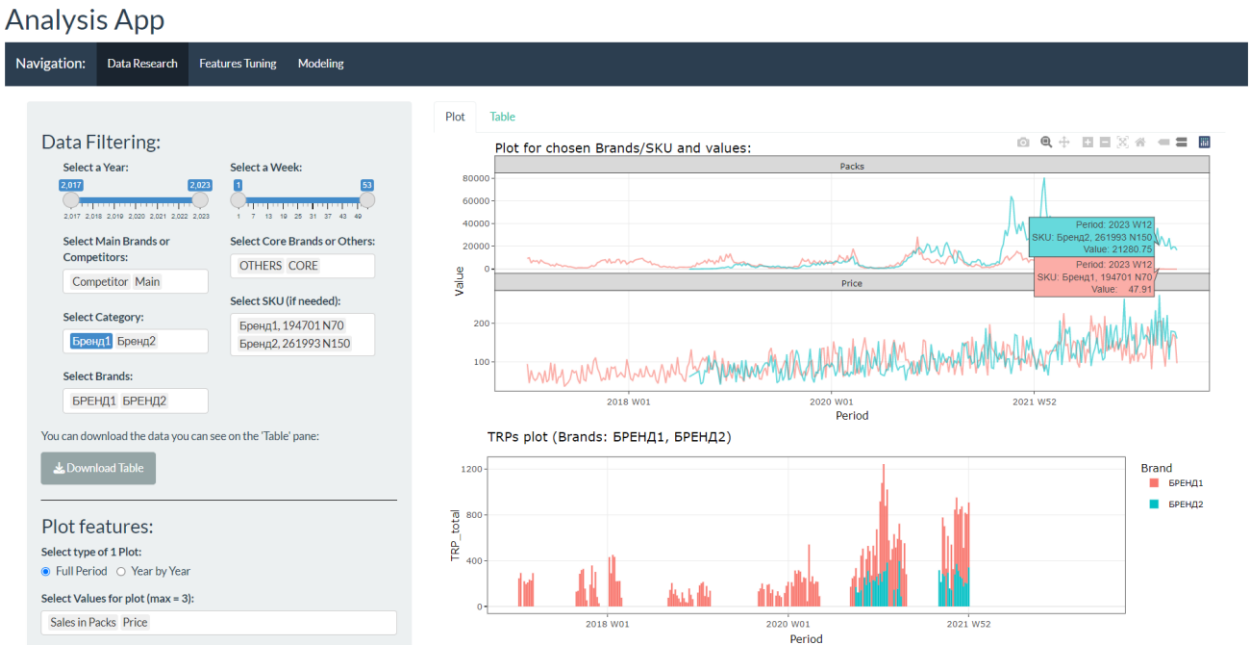


Рисунок 3.4 – Приклад відображення декількох змінних із обраними декількома SKU

Джерело: розроблено автором

Якщо ж користувач обирає варіант графіку “Year by Year”, то дані спочатку переводяться у формат ‘tsibble’ для того, аби побудувати графік за допомогою функції ‘gg_season’. В такому випадку кожна з обраних змінних відображається окремою лінією для кожного окремого року. Це потрібно для більш зручного

порівняння динаміки певної метрики. Кожен рік в такому разі підписується праворуч відповідної лінії на графіку. Приклад використання такого типу графіку вказано на рис. 3.5.

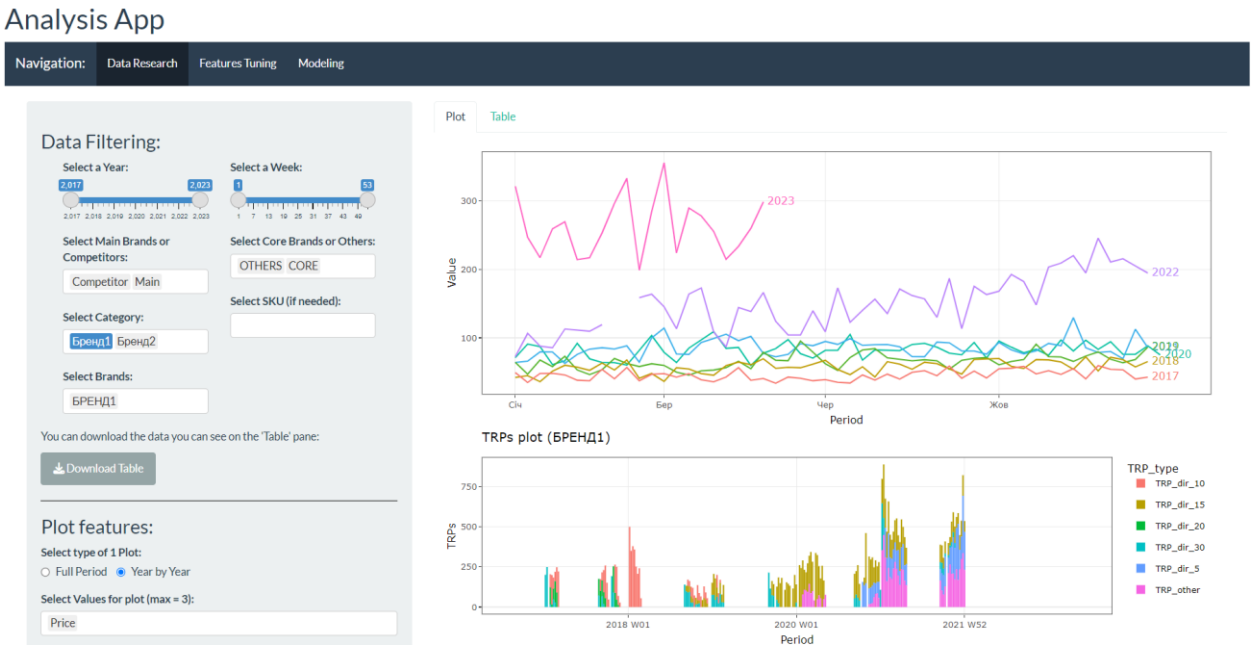


Рисунок 3.5 – Приклад використання графіку типу “Year by Year”

Джерело: розроблено автором

Ще одним елементом цієї частини веб-додатку є графік для відображення TRP – тобто рекламної активності на ТБ. Для побудови цього графіку відповідна функція бере відфільтровані дані, згідно попередньо описаних функцій, перетворює їх у довгий формат із застосуванням регулярних виразів для вибору усіх змінних, що містять “TRP” у назві, та створюють діаграму із відображенням відповідних змінних. У цій частині також було реалізовано можливість відображення кожного типу TRP окремо, та їхню суму загалом, в залежності від кількості обраних брендів. Якщо бренд обраний тільки 1, то графік будується детально по кожному TRP окремо, що видно на рисунку 3.1. Якщо брендів обрано більше одного, то дані ТБ активність відображається по кожному бренду окремо, як на рис. 3.4.

Останнім елементом цієї частини є можливість вивантаження даних у файл формату “.xlsx”. Це дозволяє використовувати дані для швидкої обробки даних із можливістю подальшого використання цих даних у інших цілях. Для реалізації вивантаження даних використовується конструкція ‘downloadHandler’, яка дозволяє обрати назву файлу (яку користувач може змінити) та які саме дані мають бути вивантажені. Під час натискання кнопки “Download Handler” користувачу пропонується перейменувати файл та зберегти його у будь-яке бажане місце на комп’ютері. В результаті у визначеному місці зберігається файл із таблицею, що відображена на вкладці “Table” у веб-додатку.

3.3.2 Features Tuning

Друга частина веб-додатку має на меті проведення проміжного етапу між первинною обробкою даних та їхнім моделюванням, а саме – налаштування та додавання додаткових змінних до даних. Відповідно, це є логічним продовженням першого етапу “Data Research”, де можна провести “exploratory analysis”, розглянути різні метрики різних брендів і різних комбінаціях, переглянути графіки по ним та зробити певні висновки щодо подальшого аналізу та прогнозування даних. А також ця частина є підготовчою для переходу до наступної, третьої частини веб-додатку – “Modeling”.

Загальний опис роботи частини з точки зору користувача. Загальний вигляд цієї частини веб-додатку показаний на рис. 3.6, де можна побачити майже аналогічний з частиною “Data Research” блок “Data Filtering” для фільтрації даних з боку користувача, що знаходиться ліворуч.

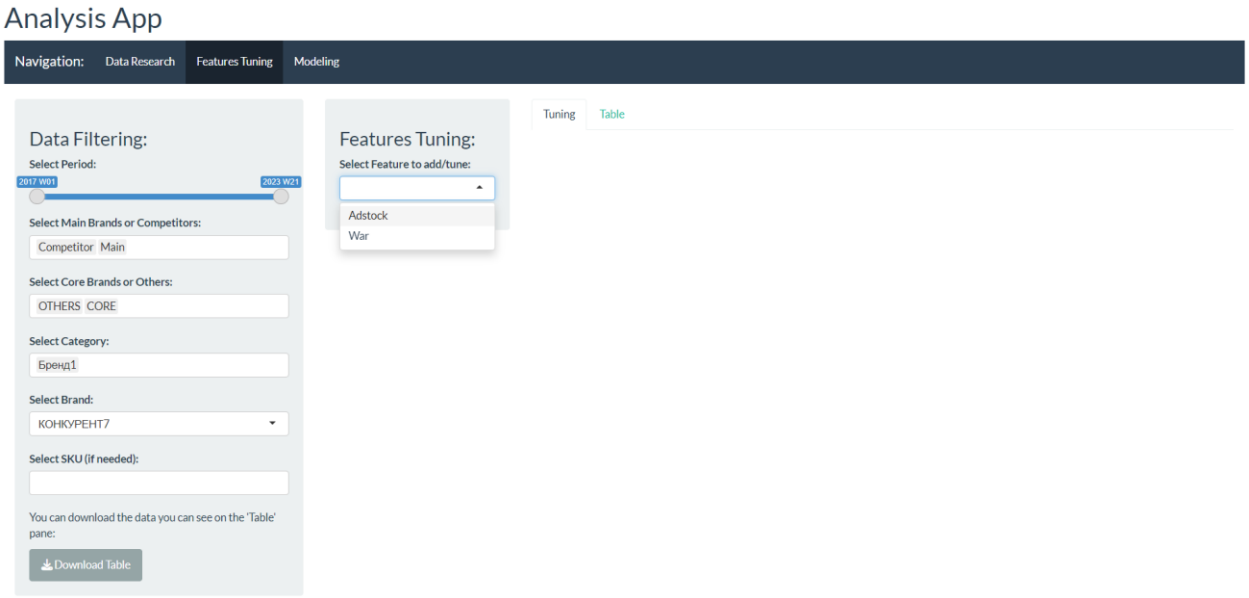


Рисунок 3.6 – Загальний вигляд частини ‘Features Tuning’ у розробленому веб-додатку

Джерело: розроблено автором

Серед змін у порівнянні до першої сторінки веб-додатку можна відмітити можливість фільтрації дати не по рокам і тижням окремо, а загалом за увесь доступний період, у форматі рік-тиждень. Також в цій частині є можливість обрати тільки 1 певний бренд для проведення аналізу та моделювання, але кількість СКЮ залишилась необмеженою. Це пояснюється тим, що для налаштування та додавання певних змінних (як і для моделювання) потрібно використовувати один набір даних, де є одна цільова змінна та декілька залежних, за що і буде відповідати можливість обирати тільки один бренд. Необмежена фільтрація по СКЮ потрібна тому, що користувач може мати необхідність відібрати або прибрати певну кількість СКЮ за певними параметрами, наприклад, є потреба проаналізувати дані бренду по певним смакам продуктів, форматам упаковки або розміру цих упаковок тощо. Також нижче під фільтрами є кнопка для завантаження даних, що знаходяться у вкладці “Table” праворуч. В даному випадку була реалізована можливість запису усіх налаштувань щодо набору даних, таких як: які фільтри були

застосовані, які змінні були додані та які при цьому параметри для кожної змінної були застосовані.

Праворуч від блоку з фільтрацією є інший, в якому користувачу запропоновано обрати один з варіантів змінних, що можна налаштувати/додати, і після цього вибору користувач побачить параметри, що можна налаштувати до відповідної змінної. Ще праворуч знаходиться вивід для користувача, де вкладка “Tuning” відповідає за вивід результатів налаштованих параметрів, а “Table” показує таблицю з відфільтрованими даними, де у разі додавання змінних вони також будуть відображатись. Оскільки доступно два різних варіанта змінних, що можна налаштувати, то кожен з них треба описати більше детально.

Першою змінною, з якою користувач може попрацювати у цій частині веб-додатку є “Adstock”. Про цей метод вже було детально описано у минулих розділах, тому розберемось із реалізацією підбору параметрів для цього методу. На рис. 3.7 можна побачити приклад того, як виконується такий підбір.

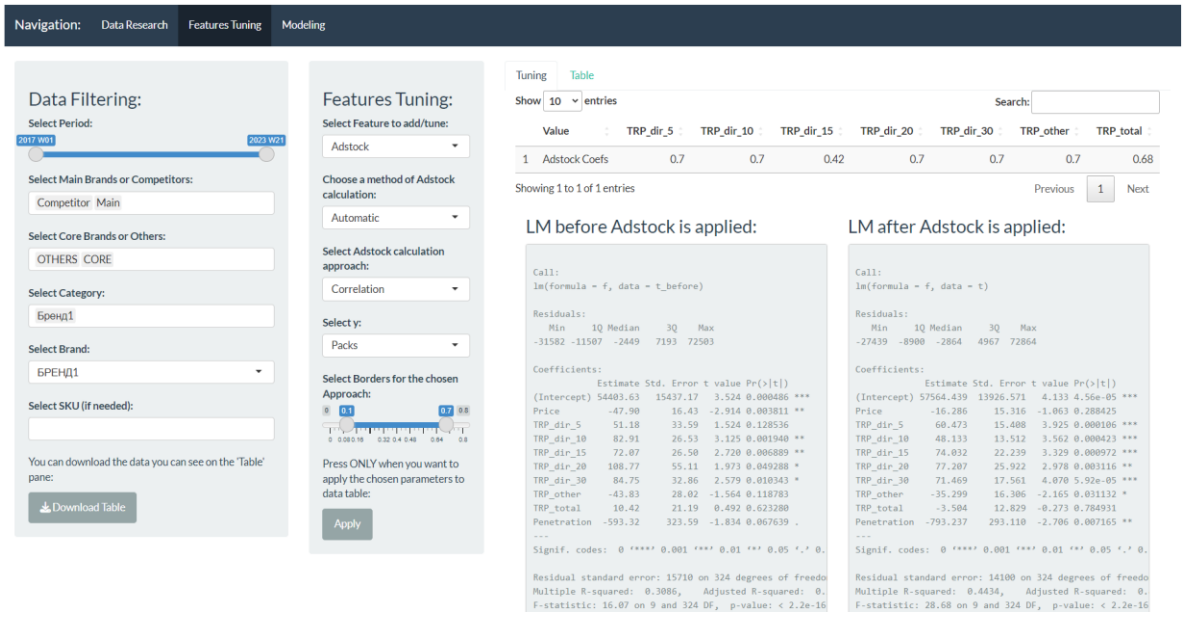


Рисунок 3.7 – Приклад використання підбору коефіцієнту Adstock у розробленому веб-додатку

Джерело: розроблено автором

Якщо користувач бажає підібрати коефіцієнти Adstock для своїх змінних із медіа активністю, і застосувати цей метод для кожної такої змінної, то цей розділ стане в нагоді. Першим, що потрібно обрати користувачу в такому разі – це метод розрахунку коефіцієнтів Adstock. Серед варіантів доступні “Automatic” та “Manual”, перший з яких відповідає за автоматизований підбір коефіцієнтів для кожної потрібної змінної, а другий – за те, аби користувач самостійно обрав коефіцієнт від 0 до 0.8, що варто застосувати до цих змінних. У разі використання першого з цих варіантів користувач отримує також можливість обрати підхід до розрахунку коефіцієнтів – “Adstock calculation approach”. Серед них є "Correlation", "LM + max TRP coef", "Correlation + lag", для кожного з яких було розроблено відповідну функцію, що були описані у минулих розділах. Також користувач може змінити цільову змінну, щодо якої будуть розраховуватись усі головні елементи у функціях. Цей input відповідає за аргумент ‘y’ у кожній з функцій для розрахунку коефіцієнтів. Також серед налаштувань користувач може обрати межі коефіцієнту від 0 до 0.8, аби провести підбір коефіцієнта у бажаному діапазоні. Останнім ввідним елементом у цьому блоці є кнопка, після натискання якої до кожної змінної із медіа активністю буде застосована модель Adstock із відповідними коефіцієнтами, які вказані праворуч, у верхній таблиці.

Після вибору бажаних параметрів користувач може одразу переглянути як саме показує себе той чи інший підхід до обраного набору даних. Цей процес реалізовано через побудову двох лінійних моделей регресії – до та після застосування моделі Adstock. Завдяки цьому користувач одразу може переглянути як кожен із елементів змінює свої коефіцієнти, значимість та загалом як змінюється та показує себе регресійна модель при застосуванні різних підходів до розрахунку коефіцієнтів Adstock.

Другою змінною для налаштування є “War” – додаткова змінна, що може бути використана для більш якісного опису аномального (у ринковому сенсі) періоду з початку повномасштабного вторгнення Росії до України 24 лютого 2022 року. В основі підходу до розробки можливості такої оцінки є порівняння середніх

ринкових показників певних минулих років із цифрами 2022-2023 років. На рис. 3.8 можна побачити приклад налаштування цієї змінної.

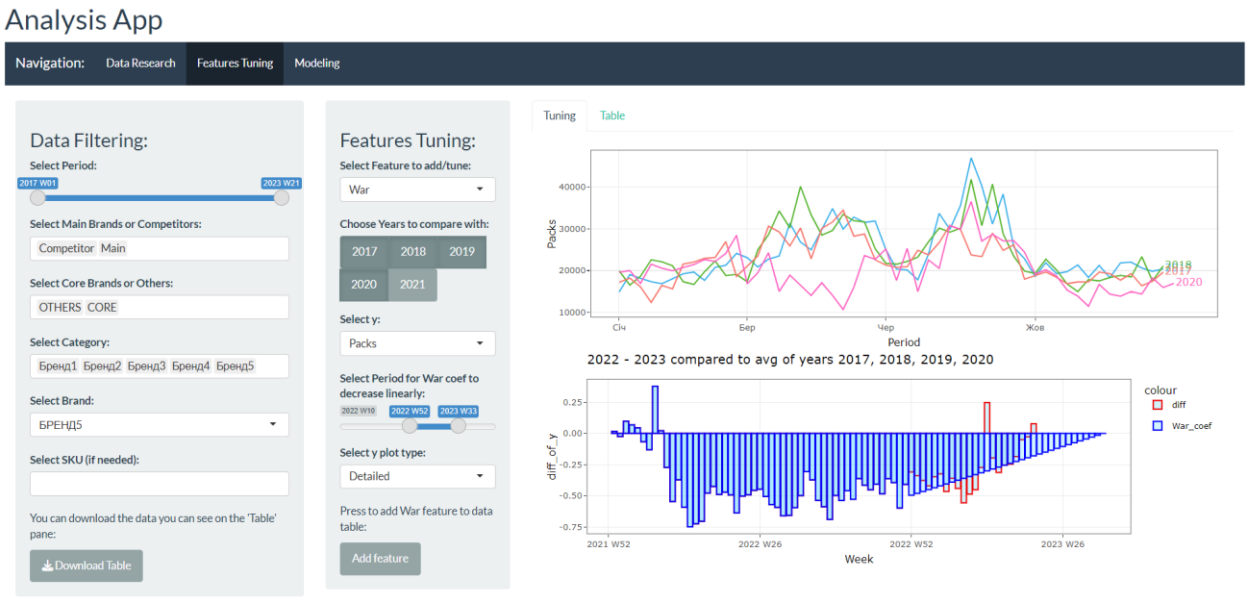


Рисунок 3.8 – Приклад налаштування змінної War у розробленому веб-додатку

Джерело: розроблено автором

Серед налаштувань у даному випадку користувачу надається можливість обрати бажані роки до 2022 року, по яким будуть розраховані середні значення, які будуть порівняні з поточними значеннями та виведені у якості коефіцієнтів, які потім можна додати до створеного набору даних (у вигляді змінної впливу війни). Серед додаткових параметрів для налаштування такого підходу є також вибір ринкового показника (у випадку описаного раніше датасету ними є продажі в упаковках, продажі в грн, ціна та пенетрація бренду), по якому проводяться усі розрахунки. Також можливо обрати період, протягом якого користувач бажає отримати лінійне зменшення коефіцієнту впливу війни до 0. Це може бути корисним у випадках, коли спостерігається зменшення різниці значень між середнім по минулих роках та поточними. В такому разі можна описати тільки певний фактичний період, або ж навпаки, зробити певну оцінку на майбутні періоди щодо можливої динаміки такого показника.

Також користувач може обрати який саме графік відобразити у веб-додатку – детальний по кожному обраному вище року (рис. 3.8), або ж графік із середнім значенням по обраним роках та значеннями за 2022 рік (рис. 3.9). Це дозволяє користувачу отримати більше інформації про динаміку певного показника протягом років та мати більше можливостей для якісного вибору усіх попередньо зазначених параметрів. Внизу блоку знаходиться кнопка, що дозволяє додати налаштований параметр до раніше створеного набору даних.

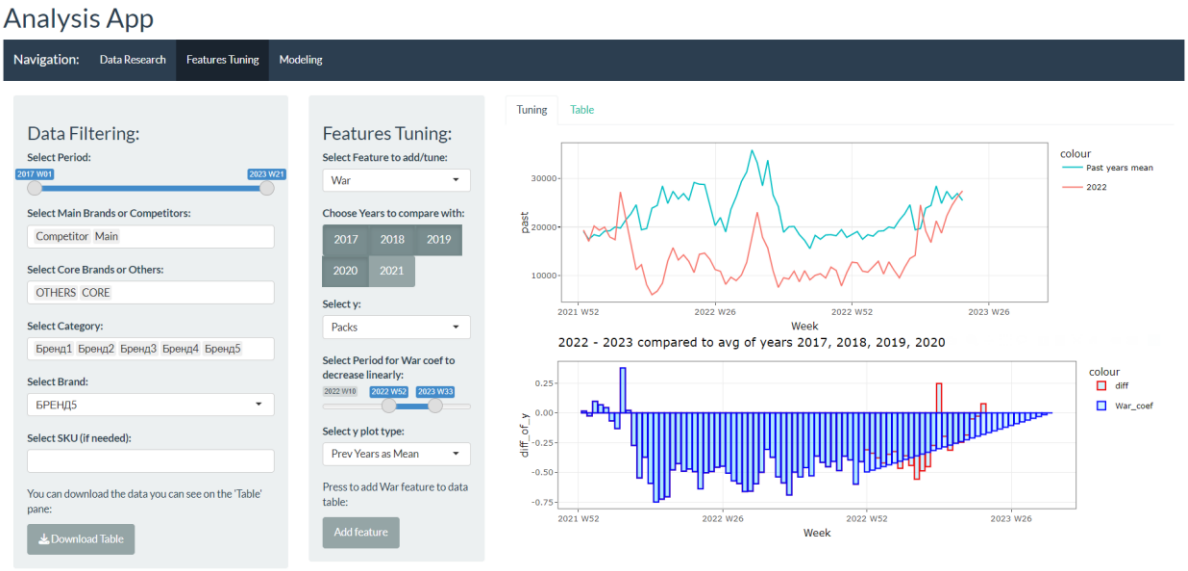
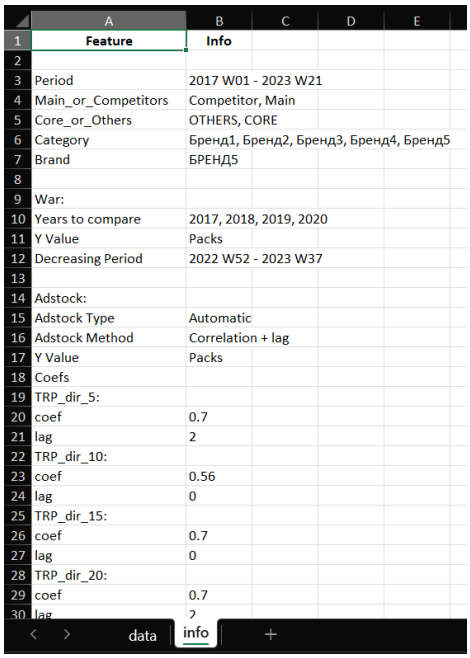


Рисунок 3.9 – Приклад використання графіку “Prev Years vs Mean”

Джерело: розроблено автором

Як було зазначено, завантаження даних в цій частині веб-додатку також супроводжується додаванням мета-інформації до вивантаженого файлу. На рис. 3.10 зображений приклад того, як може виглядати ця частина вивантаженого файлу, де видно кожен елемент, що був використаний та застосований в цій частині веб-додатку.



	Feature	Info
2		
3	Period	2017 W01 - 2023 W21
4	Main_or_Competitors	Competitor, Main
5	Core_or_Others	OTHERS, CORE
6	Category	Бренд1, Бренд2, Бренд3, Бренд4, Бренд5
7	Brand	БРЕНД5
8		
9	War:	
10	Years to compare	2017, 2018, 2019, 2020
11	Y Value	Packs
12	Decreasing Period	2022 W52 - 2023 W37
13		
14	Adstock:	
15	Adstock Type	Automatic
16	Adstock Method	Correlation + lag
17	Y Value	Packs
18	Coefs	
19	TRP_dir_5:	
20	coef	0.7
21	lag	2
22	TRP_dir_10:	
23	coef	0.56
24	lag	0
25	TRP_dir_15:	
26	coef	0.7
27	lag	0
28	TRP_dir_20:	
29	coef	0.7
30	lag	2

data info +

Рисунок 3.10 – Приклад завантажених даних, використовуючи розроблений веб-додаток

Джерело: розроблено автором

Реалізація UI частини. Input-елементи, що використовуються для фільтрації даних є дуже схожими до тих, що були описані щодо 1 частини веб-додатку. Єдиною суттєвою відмінністю в цій частині був input для фільтрації по періоду, що використовуватиметься у наборі даних. Він був реалізований за допомогою функції ‘sliderTextInput’, оскільки в ідеї було відображення цього фільтру саме у форматі слайдеру із зазначеними тижнями року, а не датами, що могло б бути зроблено завдяки функції ‘dateRangeInput’.

Другий блок цієї частини був виконаний у схожій структурі до першого, але оскільки потрібно було додати реалізацію перемикання між різними змінними для їхнього налаштування, то було імплементовано велику кількість умовних блоків, які відкриваються тільки у разі вибору певних інших елементів input. За це відповідала функція ‘conditionalPanel’, яка, використовуючи синтаксис мови JavaScript, дозволяє створювати такі умовні елементи взаємодії з додатком.

Третій блок у “Features Tuning”, який представляє собою панель для виводу інформації, була реалізована також із застосування умовних елементів інтерфейсу,

по аналогічним причинам. Більшість варіантів для виводу інформації вже були описані для першої частини веб-додатку, і серед нових можна зазначити ‘verbatimTextOutput’, який дозволяє виводити відображати інформацію у схожому до вбудованої консолі вигляді. Цей елемент був реалізований у частині виводу результатів лінійних моделей при налаштування параметра “Adstock”.

Реалізація Server частини. Структура серверної частини, що відповідає за фільтрацію даних має схожий вигляд до описаної для 1 частини веб-додатку, але серед важливих елементів в даному випадку є додавання змінної ‘ft_df’, що має формат реактивної змінної (‘reactiveValues’), який відповідає за дані, що мають оновлюватись при настанні певної умови, і з якими можна взаємодіяти у різних частинах додатку. У цьому випадку ця змінна має дві складові – ‘ft_data’, яка відповідає за зберігання набору даних, що налаштовується у цій частині веб-додатку, та ‘dt_info_table’, яка зберігає усю мета-інформацію щодо змін у наборі даних для подальшого її виводу у збережений файл. Таким чином, на початку опису роботи цієї частини до змінної ‘ft_df’ присвоюються пусті значення, а після проведення певних маніпуляцій із даними ця змінна оновлюється за допомогою функції ‘observeEvent’, який першим аргументом отримує інформацію щодо якогось елементу з веб-додатку (наприклад дані про натискання кнопки про додавання змінної “War”), а другим – інформацію щодо того, яку дію потрібно виконати, у даному випадку – перезаписати існуючі змінні в середині ‘ft_df’. Для більшої зручності усі важливі функції та змінні, що використовуються у цій частині веб-додатку мають префікс ‘ft’, що означає “Features Tuning”.

Програмна реалізація того, як відбувається процес розрахунку та виводу даних для налаштування змінної “Adstock”. Після вибору користувачем потрібних параметрів для налаштування цієї змінної в серверній частині коду відбувається перевірка отриманих ввідних даних та розраховуються моделі Adstock відповідно до них. Так, в залежності від обраного методу та підходу щодо підбору коефіцієнту Adstock, прописані в коді умови дозволяють розраховувати тільки необхідний для користувача варіант, при цьому одразу зберігаючи усі коефіцієнти та мета-дані для

подальшого виводу на екран, і одразу застосовуючи ці коефіцієнти до копії створеного набору даних, але не замінюючи їх, оскільки заміна відбувається тільки після натискання відповідної кнопки в інтерфейсі. Для побудови лінійних регресійних моделей для даних із застосованою та без застосованої моделі Adstock, створені дані копіюються та створюється окремо дві моделі на відповідних даних, які потім зберігаються під різними змінними та викликаються як два окремі елементи в інтерфейсі.

Програмна реалізація того, як відбувається процес розрахунку та виводу даних для налаштування змінної “War”. Після вибору усіх необхідних параметрів в серверній частині відбувається розділення датасету, що був створений в рамках цієї частини веб-додатку, на частину 2022 року та минулих обраних років. Також додатково прописана умова того, що у разі виявлення у даних періоду, що пізніше за 2022 рік, то це буде використовуватись як окрема частина цього датасету. Після отримання бажаного датасету виконується розрахунок коефіцієнту “War” шляхом розділення значень за 2022 та подальші роки на значення середнього по минулим рокам, та віднявши одиницю, для отримання значень, що відображають різницю у динаміці. Після цього застосовується input, що відповідає за період для лінійного зниження цього коефіцієнту. Для цього знаходиться середнє по усім значенням коефіцієнту до початку цього періоду, після чого це середнє ділиться на кількість тижнів у цьому періоді лінійного спадання, і застосовується почергове віднімання такого значення від розрахованого середнього для кожного тижня, що користувач зазначив у додатку. Далі до цих даних додається змінна по даті, усі пусті значення замінюються на 0 і відбувається об’єднання із набором даних, створеним у цій частині веб-додатку. При натисканні кнопки в інтерфейсі, готовий датасет також замінюється для подальшого його використання.

Також при налаштуванні змінної “War” користувачу відображаються два типи графіків, один з яких змінюється в залежності від обраного варіанту. За це відповідає окрема функція в серверній частині, в якій будується три різних графіки, один з яких фактично просто візуалізує значення усіх років по бажаному

ринковому показнику за роки до 2022 року, другий розраховує середнє по значенням метрики до 2022 року та візуалізує це середнє разом із значеннями самого 2022 року, і третій – отримує дані із частини, що була описана вище, із коефіцієнтом “War”, при чому на графіку можна побачити подвійну діаграму, де одна з них є саме відображенням коефіцієнту, що можна додати до даних з урахуванням усіх налаштувань, а другий відображає дані, з яких цей коефіцієнт і розраховується, тобто без застосування лінійного зниження коефіцієнту протягом певного періоду.

3.3.3 Modeling

Третя частиною веб-додатку створена для проведення моделювання даних, що були налаштовані у другій частині. Загальний вигляд цієї частини веб-додатку представлений на рис. 3.11.

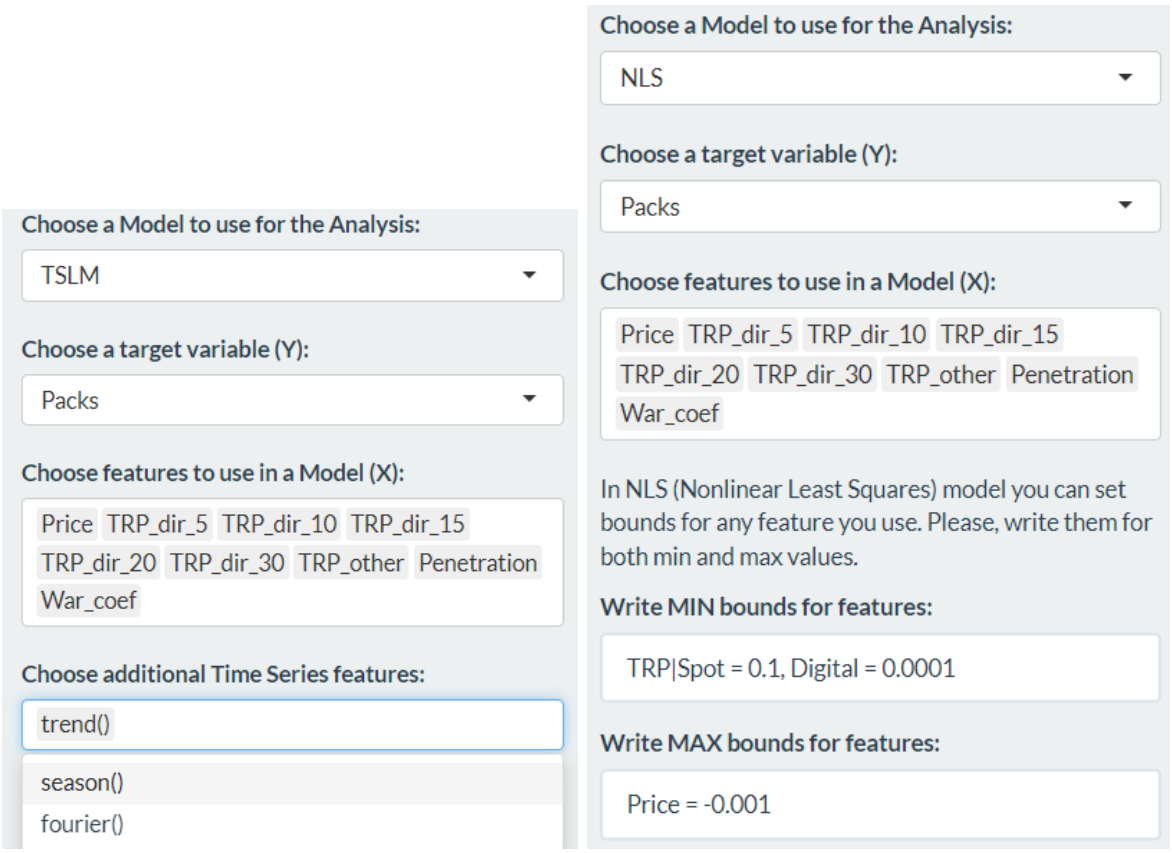
Рисунок 3.11 – Загальний вигляд частини “Modeling” у розробленому веб-додатку

Джерело: розроблено автором

На цій сторінці користувач має можливість обрати яку модель він бажає побудувати та налаштувати деякі параметри кожної моделі. На даний момент серед перших двох ввідних параметрів доступні лише зазначені на рисунку 19. Наприклад, для варіанту “Choose a data to use” можна обрати варіант “From Features Tuning”, який дозволяє використовувати дані, що були налаштовані у другій частині веб-додатку з відповідною назвою. Надалі планується додати можливість завантаження датасетів з різних джерел напряму до цієї та інших частин веб-додатку, аби додати більшої гнучкості для роботи із даними. Другим фільтром користувачу пропонується обрати тип аналізу для виконання, де на даний час можна провести лише регресійний аналіз, хоча надалі планується додати такі типи аналізу, як аналіз часових рядів, кластерний аналіз тощо.

Наступним елементом користувач може обрати яку саме модель використовувати для обраного типу аналізу. В даному випадку для регресійного аналізу доступні варіанти моделей “LM”, “TSLM” та “NLS”, кожна з яких були детально описані у минулих розділах. Наступним елементом можна обрати яка зі змінних має бути цільовою, тобто щодо якої і проводиться регресійний аналіз, а наступним, відповідно, можна обрати які саме змінні використовувати для опису цільової, тобто обираються “Y” та “X”-и моделей. Ще одним input, що є спільним для усіх типів регресійного аналізу в цьому веб-додатку є повзунок для вибору співвідношення між тренувальною та тестовою частинами даних. Такий підхід до розділення вибірки є дуже зручним та ефективним, оскільки дозволяє інтерактивно та швидко подивитись як саме модель буде себе поводити при різних періодах для тренування та тестування, які проблеми можуть виникнути при цьому тощо. Останнім спільним елементом для усіх типів моделей є кнопка, яка відповідає за побудову обраної специфікації моделі. Використання кнопки замість автоматичного перерахування кожного разу, коли користувач змінює якийсь input зумовлено тим, що це економить ресурси та час для виконання цієї задачі, зменшує можливість виникнення помилки.

На рис. 3.11 представлено приклад використання звичайної лінійної моделі регресії, але при виборі інших моделей користувач має можливість налаштувати більшу кількість змінних. Так, при виборі варіанту “TSLM” з’являється можливість обрати додаткові змінні, які відповідають за складові часових рядів у цільовій змінній (рис. 3.12). А при виборі моделі “NLS” можливо також налаштувати мінімальні та максимальні значення коефіцієнтів регресії, що будуть застосовані до відповідних змінних (рис. 3.13). В цьому випадку користувач має ввести з клавіатури бажані обмеження у форматі “Змінна” = “число”. Програмна частина додатку автоматично розпізнає такий паттерн та застосує необхідні параметри до даних. У випадку некоректного вводу, користувачу буде надано помилку про потребу правильного вводу цих обмежень (рис. 3.14).



Рисунки 3.12-3.13 – Зміна вигляду інтерфейсу в залежності від обраної користувачем моделі регресії

Джерело: розроблено автором

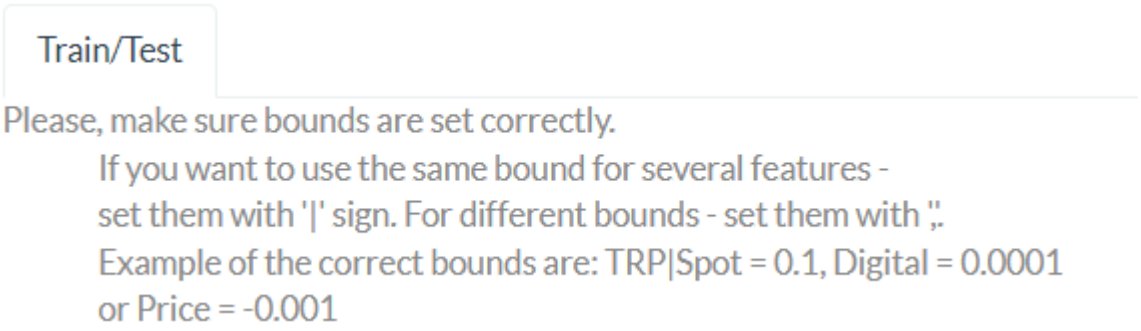


Рисунок 3.14 – Вивід для користувача у разі запуску некоректної моделі NLS
Джерело: розроблено автором

Після налаштування обраної моделі та натискання кнопки для запуску програмної обробки даних, користувачу буде надано результати у вигляді трьох складових – графіку тренувальної та тестової частин даних із відповідними прогнозами, таблиці із результатами моделі на тренувальній та тестовій частинах, консольного виводу результатів обраної моделі (рис. 23). На верхньому графіку можна побачити результат побудови моделі на відповідних частинах даних та порівняння змодельованих значень із фактичними. На таблиці із похибками прогнозу можна порівняти тренувальну та тестову частини використовуючи три види похибок – MAPE, RMSE, MAE, кожна з яких дає розуміння щодо точності моделі з різних сторін. Тому що MAPE – це відносна похибка, що показує абсолютне відхилення у відсотках, RMSE – дуже чутливий показник до значних відхиленнях прогнозу, оскільки використовує квадратичне відхилення від фактичних значень, а MAE показує похибку у абсолютних значеннях, що також надає додаткової інформації для розуміння якості побудованої моделі. Зрештою, консольний вивід результатів моделі дає ще більше інформації для оцінки якості моделі, оскільки тут можна побачити коефіцієнти кожного фактору моделі, їхню значущість, оцінити R2 моделі на тренувальній частині тощо.

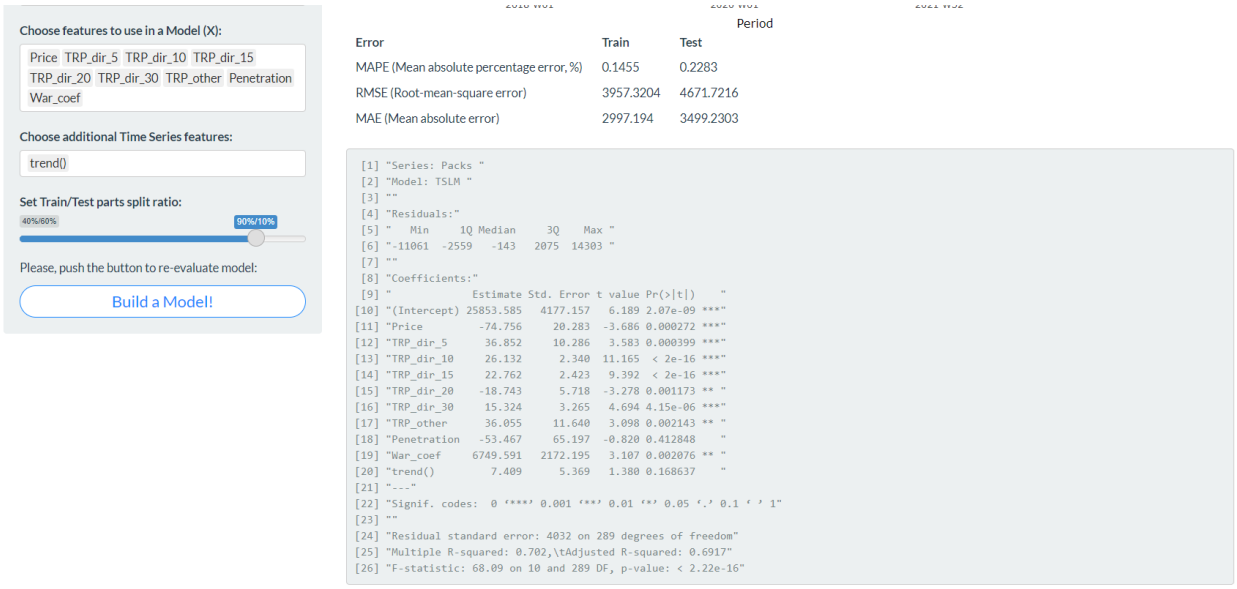


Рисунок 3.15 – Консольний вивід результатів моделі

Джерело: розроблено автором

Реалізація UI частини. Структура веб-додатку з точки зору розробки інтерфейсу для третьої частини програми є схожою до попередніх частин і більшість елементів вже була описана раніше. Серед унікальних для цієї частини методів можна відмітити функцію ‘textInput’, що використовується для опису обмежень змінних в моделі “NLS”. Цей підхід був використаний для полегшення розробки додатку, замість створення декількох елементів, кожен з яких окремо обирав би змінну для обмеження, і окремо елементи, в яких зазначались би значення для обмеження. Така структура додатку могла б стати занадто громіздкою та неочевидною для користувача, на відміну від запропонованого варіанту. Також вартим уваги є елемент для вибору співвідношення між тренувальною та тестовою частинами, оскільки він реалізований шляхом використання функції ‘sliderTextInput’, і фактично на виході програма отримує просто текстове значення, наприклад “80%/20%”, що не може бути напряму використане для розділення даних, тому в описі серверної частини буде більш детально описаний програмний шлях реалізації такого підходу. Цей підхід також був використаний для більшої зручності з боку користувача, оскільки такий вигляд цього елементу є очевидним у

використанні, на відміну від декількох різних повзунків або текстових елементів, де можна було б прописати бажане співвідношення тощо.

Реалізація Server частини. Перш за все у серверній частині було створено функцію для ініціалізації функції для регресійної моделі, що використовується у кожній моделі. Ця формула передається наступній функції, де відбувається розділення даних на тренувальну та тестову вибірки та оновлення формули у разі використання факторів для аналізу часових рядів. Текстове значення із інформацією щодо співвідношення для розділення даних використане тут шляхом комбінації декількох формул, одна з яких виконує розділення текстового значення за обраними символами, а інша спрощує отриману змінну формату списку у векторний формат для подальшого використання першого значення у якості коефіцієнту. Таким чином, наприклад, із текстового значення “80%/20%” отримується значення 80, яке ділиться на 100 і використовується у функції ‘initial_time_split’, яка виконує розділення часового набору даних у послідовному порядку. У цій частині також присутні валідаційні складові, для забезпечення коректного функціонування моделей. Ці складові попереджують користувача про необхідність вибору деяких складових вхідних елементів, а також, наприклад, про те, що цільова змінна зазначена у факторах впливу, якщо таке станеться. А у разі вибору складових часових рядів для моделі “TSLM”, програмний код перебирає усі ці складові, і якщо однією з них є використання рядів Фур’є, то виконується функція ‘fourier_set’, де тестовою частиною вказується тільки що розділена частина даних. Для складових сезонності та тренду – вони просто додаються до формули, оскільки не потребують додаткового опрацювання (в рамках веб-додатку).

Наступною частиною у серверній частині є функція, яка будує обрану модель. Для того, аби була можливість використовувати кнопку в інтерфейсі, яка б запускала цю частину коду, було обрано використовувати функцію ‘isolate’, який дозволяє буквально ізолювати певну частину коду від перевиконання у разі зміни якихось input всередині цієї функції. Таким чином, за межами цієї ізольованої частини знаходиться лише виклик input, який прив'язаний до описаної кнопки у

інтерфейсі. Сама побудова моделей виконується шляхом початкової перевірки обраної моделі користувачем, побудовою відповідної моделі та побудовою прогнозів на тренувальній та тестовій частинах, збереження цих результатів і також виконання описаної у попередніх розділах функції ‘errors_func’ для подальшого відображення таблиці похибок в інтерфейсі. Також важливо відмітити, що у разі побудови моделі “NLS” спочатку відбувається перевірка ненульової умови щодо мінімальних та максимальних обмежень параметрів моделі. Якщо певні обмеження були задані, то для кожного з таких обмежень виконується частина, де текстовий параметр переводиться у векторний шляхом розділення по комі (якщо для якогось обмеження було обрано декілька змінних). Після цього виконується валідація, яка дозволяє впевнитись, що користувач правильно ввів потрібні обмеження. Якщо цей етап пройдено, то за допомогою циклічного перебору кожного ввідного елементу та використання регулярних функцій виконується призначення вектору, в якому і знаходяться відповідні обмеження. Цей фактор далі передається у функцію ‘nls_set’, яка була описана у попередніх розділах. Вже ці результати передаються у функцію ‘nls’, звідки беруться результати для прорахунку прогнозів за відповідні періоди. Додатковим обмеженням nls є те, що у результатах не можна отримати показник R2, тому він розраховується за допомогою функції ‘r.sq_func’ та додається до виводу моделі.

Остання функція у цій частині виконує побудову графіку по заданим параметрам. Ця функція також включає в себе ізольовану частину для того, аби графік не перебудовувався кожного разу, коли елементи моделі налаштовуються заново, оскільки це видавало б некоректні результати. Бо графік має будуватись тільки після того, коли сама модель будується, а це відбувається тільки при натисканні кнопки в інтерфейсі.

3.4 Перспективи та плани щодо подальшої розробки веб-додатку для аналізу даних

Розроблений на даний момент веб-додаток вже може використовуватись для оптимізації значної кількості задач у разі роботи із описаним датасетом. Але для досягнення кращої якості моделювання та проведення більшої кількості різних маніпуляцій із даними цей веб-додаток може бути значно розширений.

Однією з бажаних модифікацій веб-додатку є дати можливість користувачу завантажувати власні дані, відмінні від тих, що використовуються наразі як за замовченням. Обрані завантажені дані повинні мати певні обмеження, з якими користувач може ознайомитись всередині самого додатку, наприклад, дані мають бути у потижневому форматі, і повинні мати певну структуру для забезпечення безпомилкового виконання програмного коду. Таким чином користувач буде мати змогу використовувати різні набори даних для виконання їх обробки, розширення та моделювання на їхній основі. Це дозволить працювати із даними різних клієнтів значно швидше та ефективніше виконувати різноманітні задачі щодо оцінки впливу медіа на ринкові показники, прогнозувати ці ринкові показники тощо.

Також серед майбутніх доопрацювань можна зазначити розширення другої частини веб-додатку, де відбувається налаштування та додавання змінних. Для розширення цієї частини вже є деякі ідеї, які допоможуть проводити подальше моделювання ефективніше. Серед них, наприклад, можливість додавання змінних, що налаштовуються шляхом аналізу ряду даних як часового ряду. В якості самих змінних у такому випадку можуть виступати значення сезонності та тренду, які можуть будуватись на цільовій для моделювання змінній, або ж можна реалізувати варіант, при якому ці показники будуть розраховуватись по даним відповідної категорії, а не самого бренду, що дозволило б отримувати більш стійкі значення сезонності та показувати загальний тренд всієї категорії. Самі змінні в такому разі можуть бути взяти із декомпозиції відповідного часового ряду, наприклад, використовуючи метод STL (Seasonal-Trend decomposition using LOESS, де LOESS

– це locally estimated scatterplot smoothing) [47]. Цей метод є одним із найкращих підходів до декомпозиції часових рядів, і має велику кількість переваг перед класичними методами, такими як SEATS, X-11 тощо [48]. Серед головних переваг можна зазначити можливість використовувати будь-який тип сезональності, а не тільки місячну або квартальну, також цей метод дозволяє використовувати додаткові параметри налаштування, які можуть бути зазначені користувачем, і ця модель може бути стійкою до викидів у даних, що дозволяє будувати стійкіші прогнози показників сезонності та тренду [49].

Також варто розглянути можливість додавання певних змінних, що користувач може завантажити з стороннього файлу. Адже інколи варто провести точкові зміни у даних, додати коефіцієнти для опису декількох аномальних точок, або просто використати додаткові предиктори для якіснішого опису даних. При цьому може бути корисним розробити комплексний підхід до цього питання, який дозволятиме використовувати різні варіанти вигляду самих даних, у різних розрізах періодів, з різними змінними, що можуть ідентифікувати ці періоди. І при цьому будь-які такі дані мають зрештою зводитись до потижневого формату, що використовується у веб-додатку. Варто також звернути увагу на роздільне маніпулювання із числовими та категорійними даними, додати можливість вибору типу агрегації або розділення даних тощо.

Ще одним важливим аспектом для розширення веб-додатку є збільшення кількості моделей, із якими може взаємодіяти користувач. Це включає в себе додаткові регресійні моделі, частина з яких вже була реалізована у додатку, а також можливість використання іншого типу аналізу даних, такого як аналіз часових рядів, класифікація та кластеризація даних.

Серед додаткових регресійних моделей можна розглянути регресію із регуляризацією (наприлад, модель Elastic Net), модель MMM (Mix Marketing Modeling), регресійні дерева рішень тощо. Elastic Net використовує штрафи з обох Lasso та Ridge регресій для отримання найбільш оптимального варіанту з статистичної точки зору [50]. Метою використання MMM є розуміння того, наскільки кожен маркетинговий фактор впливає на продажі, і скільки потрібно

витрачати бюджету на кожен з них. MMM допомагає визначити ефективність кожного маркетингового заходу з точки зору повернення інвестицій (ROI – Return on Investments) [51]. В R ця модель може бути реалізована шляхом використання бібліотеки ‘robyn’ від Facebook, який дозволяє отримати програмну реалізацію методу знаходження інкрементальних продажів та обирати найбільш ефективні повернення інвестицій [52]. Регресійні дерева рішень - це, дерева рішень, які використовуються для задачі регресії, яка може бути використана для прогнозування неперервних значень вихідних даних замість дискретних. Вони можуть бути корисними в багатьох областях, де зв'язок між змінними є нелінійним. Важливо пам'ятати, що алгоритм схильний до перенавчання, тому треба приділяти особливу увагу налаштуванню параметрів цієї моделі [53].

Якщо розглядати аналіз часових рядів, то можна розглянути такі моделі, як ARIMA (Autoregressive Integrated Moving Average Model), ETS (Error Trend Season model), NNetAR (Neural Network Time Series Forecasts using Autoregressive model) та інші. Експоненційне згладжування та моделі ARIMA є двома найбільш розповсюдженими підходами до прогнозування часових рядів, які забезпечують взаємодоповнюючі підходи до проблеми. У той час як моделі експоненційного згладжування ґрунтуються на описі тренду та сезонності даних, моделі ARIMA спрямовані на опис автокореляцій у даних [54]. Усі з зазначених моделей для прогнозування часових рядів є комплексними підходами та потребують якісного налаштування параметрів, де не завжди автоматизовані підходи можуть показати якісні результати.

Зазначені удосконалення, що можуть бути імплементовані до вже розробленої частини веб-додатку, дозволять аналізувати дані в більшій кількості задач, використовувати різноманітні підходи та методи для отримання необхідних результатів моделювання.

ВИСНОВКИ

Описаний у роботі веб-додаток може бути застосований у різних галузях та має значні перспективи щодо розбудови та імплементації у різні сфери науки та бізнесу.

В роботі розглянуто різноманітні методи для роботи із даними, деякі з яких включають в себе автоматизовані підходи для обробки та моделювання даних. Кожен з таких інструментів має свої недоліки та переваги та може використовуватись для вирішення різних задач, в залежності від об'єму та специфіки даних, поставлених безпосередніх задач до набору даних, комплексності аналізу, що має бути виконаний тощо. Важливим висновком є те, що наразі не існує абсолютно універсального підходу до роботи із даними, оскільки кожен із запропонованих варіантів має свою структуру і свої обмеження щодо певного аналізу. Так, в деяких інструментах досить легко можна виконати точкові маніпуляції із даними, а в деяких навпаки, це може бути не зручно, проте зробити певні загальні зміни бути значно швидше і простіше. Так само з моделюванням, де одні програмні засоби дозволяють розраховувати складні моделі з використанням машинного навчання, але при цьому не дають такого контролю над даними, і до того ж можуть бути платними застосунками, в той час як інші дають повну свободу рішень, але є дуже обмеженими у своєму функціоналі з точки зору моделювання.

Також в роботі описано розроблений веб-додаток для вирішення багатьох задач при роботі зі специфічними даними. Цей додаток був розроблений виключно на мові програмування R, що дозволяло спростити процес розробки завдяки вбудованим функціям для програмного опису інтерфейсу та серверної частини веб-додатку. Розроблена програма включає в себе три основні частини:

- "Data Research" – частина, де користувач має змогу взаємодіяти із завантаженими даними шляхом їхньої фільтрації та побудови різних видів

візуалізації цих даних. Ця сторінка може бути дуже корисним для користувача у випадках, коли треба швидко розібратись із датасетом, оглянути певні частини даних окремо від інших, проаналізувати структуру та дослідити певні закономірності в змінних;

- “Features Tuning” – друга сторінка веб-додатку, де користувач може провести розширення датасету шляхом додавання чи налаштування необхідних змінних. На даному етапі розробки користувач має два варіанти щодо ваємодії зі змінними датасету, кожна з яких має значно покращити пояснення бажаної цільової змінної. Після виконання фільтрації та обрання бажаного бренду, з яким буде виконуватись подальша взаємодія, користувач може налаштувати параметри, що відповідають за рекламну активність відповідного бренд шляхом використання моделі Adstock. Також користувач може додати змінну “War”, яка може допомогти описати період із початку повномасштабного вторгнення Росії в Україну 24 лютого 2022 року, оскільки в цей період споживча поведінка на ринку сильно змінилась, і були періоди, коли перепади ринкових показників були абсолютно аномальними у порівнянні із будь-яким іншим роком спостереження. Ці взаємодії із даними допоможуть створити більш якісні моделі у третьому розділі веб-додатку, або ж дадуть змогу користувачу вивантажити відредаговані дані та використати їх у інших цілях;
- “Modeling” – остання, на даний момент, сторінка веб-додатку, яка дозволяє користувачу проводити моделювання даних, використовуючи зазначені у роботі моделі регресії. Кожен з методів має свої відмінності та може бути використаний для певного набору даних. Завдяки інтерактивній взаємодії користувача із процесом моделювання, можна швидко оцінити кожну модель, в залежності від обраних специфікацій, що дозволяє швидко та якісно обрати найкращий можливий метод для моделювання підготовлених даних.

Представлений у роботі веб-додаток на даному етапі надає багато можливостей для роботи із описаним датасетом, і відповідно, може бути використаний для інших даних із подібною структурою. Як було зазначено,

планується продовжувати розбудову цього веб-додатку із додаванням додаткових можливостей для взаємодії із даними, такими як додавання специфічних описових параметрів цільової змінної, використання більшої кількості математичних моделей для покриття ширшого спектру задач, що можуть виникнути у користувача під час взаємодії із додатком. Також планується додати можливість використовувати дані, що будуть завантажуватись із локального файлу, який користувач буде самостійно обирати. Ці дані зрештою будуть зводитись до потижневого розподілу по даті, оскільки в роботі використовується саме цей розріз даних, який є найбільш розповсюдженим при роботі із ринковими показниками, такими як продажі, ціна, пенетрація товару тощо.

Зазначені у роботі методи та підходи дозволяють вирішити значну кількість питань та проблем, що можуть виникнути у користувача під час роботи із датасетом ринкових даних, що описаний у роботі. А в перспективі цей веб-додаток допомагатиме вирішувати ще більшу кількість задач завдяки інтеграції додаткових методів та можливості використання додаткових наборів даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Moore G. E. Cramming more components onto integrated circuits. *In Electronics*. 1965. Vol. 38, no. 8. P. 53–59.

2. What is Moore's Law? // Our World in Data. – 2023. – URL: https://ourworldindata.org/moores-law.

3. An interactive R-Shiny app for quickly visualizing a tidy, long dataset with multiple dimensions with an application in clinical trial simulations for platform trials / E. L. Meyer et al. SoftwareX. 2023. Vol. 22. P. 101347. URL: https://doi.org/10.1016/j.softx.2023.101347.

4. shinyChromosome: An R/Shiny Application for Interactive Creation of Non-circular Plots of Whole Genomes / Y. Yu et al. Genomics, Proteomics & Bioinformatics. 2019. Vol. 17, no. 5. P. 535–539. URL: https://doi.org/10.1016/j.gpb.2019.07.003.

5. How to Build a Shiny Application from Scratch. bookdown.org, 2019. URL: https://bookdown.org/hadrien/how_to_build_a_shiny_app_from_scratch/.

6. Окселенко, Н. О. (2013). Особливості застосування моделей лонгітюдних даних для сільськогосподарських підприємств. Scientific Progress & Innovations, (4), 117-120. URL: https://doi.org/10.31210/visnyk2013.04.30.

7. Higgins, Hannah (2009). The Grid Book. MIT Press. ISBN 9780262512404.

8. Collins English Dictionary – Complete and Unabridged (12th ed.). HarperCollins Publishers. 2014.

9. LINEST function. support.microsoft.com. URL: https://support.microsoft.com/en-us/office/linest-function-84d7d0d9-6e50-4101-977a-fa7abf772b6d.

10. Стівен Булен, Роб Боуві, Джон Грін. Професіональна розробка додатків Microsoft Office Excel: «Вільямс», 2007. — ISBN 0-321-26250-6.

11. Янковий А., Радзіховська Л. Особливості використання пакету statistica та ms excel для обробки статистичних даних. Інформаційні технології і

автоматизація – 2022 : матеріали xv міжнар. науково-практ. конф., м. Одеса, 21 жовт. 2022 р. Одеса, 2022. С. 62–63.

12. Hill, Ian. "18 New Languages for Drive, Docs, Sheets, and Slides". Google Drive Blog. 2013.

13. Google Trends. URL: https://trends.google.com/home?hl=uk.

14. Tableau. URL: https://www.tableau.com/products/desktop.

15. Data Visualisation | Microsoft Power BI. Power BI - Data Visualization | Microsoft Power Platform. URL: https://powerbi.microsoft.com/en-au/.

16. GitHub Copilot. GitHub. URL: https://github.com/features/copilot/.

17. Abandy R. The History of Microsoft Azure. TECHCOMMUNITY MICROSOFT.COM. URL: https://techcommunity.microsoft.com/t5/educator-developer-blog/the-history-of-microsoft-azure/ba-p/3574204.

18. Cloud Computing Services - Amazon Web Services (AWS). Amazon Web Services, Inc. URL: https://aws.amazon.com/?nc1=h_ls.

19. What is Amazon Machine Learning? - Amazon Machine Learning. Wayback Machine. URL: https://web.archive.org/web/20170328002330/http:/docs.aws.amazon.com/machine-learning/latest/dg/what-is-amazon-machine-learning.html.

20. Amazon Machine Learning. Wayback Machine. URL: https://web.archive.org/web/20170407103735/https:/aws.amazon.com/machine-learning/.

21. Product Overview. Akkio. URL: https://www.akkio.com/product.

22. What is Shiny (in R)? | Domino Data Science Dictionary. Domino Data Lab | Unleash Data Science at Scale. URL: https://domino.ai/data-science-dictionary/shiny-in-r.

23. Media O. Mastering Shiny. 2020. URL: https://mastering-shiny.org/.

24. Solymos P. The Anatomy of a Shiny Application. Hosting. URL: https://hosting.analythium.io/the-anatomy-of-a-shiny-application/.

25. Haase, Richard; Haase, Richard F. (2011). Multivariate General Linear Models. SAGE. ISBN 978-1-4129-7249-9.

26. Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. URL: https://OTexts.com/fpp3.

27. Норік Л. О., Бріль М. С. Використання рядів Фур'є в задачі прогнозування індексу промислової продукції. *Сучасні проблеми управління підприємствами: теорія та практика*. 2019. Вип. 2. С. 27–29.

28. Derbentsev V., Ovcharenko A., Bezkorovainyi V. Monitoring the state of times seris of exchange rates by using the Fourier analysis. Modeling and Information Systems in Economics. 2019. No. 97. P. 117–128. URL: https://doi.org/10.33111/mise.97.12.

29. The Nonlinear Least Squares (NLS) Regression Model. Time Series Analysis, Regression, and Forecasting. URL: https://timeseriesreasoning.com/contents/nonlinear-least-squares-nls-regression/.

30. Kommenda, M., Burlacu, B., Kronberger, G. et al. Parameter identification for symbolic regression using nonlinear least squares. Genet Program Evolvable Mach 21, 471–501 (2020). https://doi.org/10.1007/s10710-019-09371-3.

31. stats package. RDocumentation. URL: https://www.rdocumentation.org/packages/stats/versions/3.6.2.

32. Broadbent, S. (1979) "One Way TV Advertisements Work", Journal of the Market Research Society Vol. 23 no.3. P. 107–108.

33. Powell, Guy R., Marketing Calculator: Measure and manage your return on marketing investment (2008) John Wiley and Sons. ISBN 978-0-470-82395-8.

34. What are target rating points (TRPs)?. Adjust. URL: https://www.adjust.com/glossary/target-rating-points/.

35. DATE function. Microsoft Support. URL: https://support.microsoft.com/en-us/office/date-function-e36c0c8c-4104-49da-ab83-82328b832349.

36. Tidy Temporal Data Frames and Tools. tsibble. URL: https://tsibble.tidyverts.org/.

37. R: The R Stats Package. SfS – Seminar for Statistics | ETH Zurich. URL: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html#N.

38. apply(), lapply(), sapply(), and tapply() in R - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/apply-lapply-sapply-and-tapply-in-r/.

39. Numeracy, Maths and Statistics - Academic Skills Kit. The things we do here make a difference out there | Newcastle University. URL: https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html.

40. Mathematics of the ISO calendar. science.uu.nl project csg. URL: https://webspace.science.uu.nl/~gent0113/calendar/isocalendar.htm.

41. CRAN - Package fabletools. The Comprehensive R Archive Network. URL: https://cran.r-project.org/web/packages/fabletools/index.html.

42. ShinyWidgets : Extend widgets available in shiny. GitHub. URL: https://github.com/dreamRs/shinyWidgets.

43. Read Excel Files. readxl. URL: https://readxl.tidyverse.org/.

44. forecast package. RDocumentation. URL: https://www.rdocumentation.org/packages/forecast/versions/8.21.1.

45. Plotly. Low-Code Data App Development. URL: https://plotly.com/r/.

46. Interactive web-based data visualization with R, plotly, and shiny. plotly-r. URL: https://plotly-r.com/preface.

47. Seasonal-Trend decomposition using LOESS (STL). URL: https://www.statsmodels.org/devel/examples/notebooks/generated/stl_decomposition.html.

48. Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. J. (1990). STL: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics, 6(1), 3–33. URL: http://bit.ly/stl1990.

49. Seasonal-Trend decomposition using LOESS. ArcGIS. URL: https://doc.arcgis.com/en/insights/latest/analyze/stl.htm.

50. Elastic Net. Corporate Finance Institute. URL: https://corporatefinanceinstitute.com/resources/data-science/elastic-net/.

51. Kumar R. Market Mix Modeling (MMM). Medium. URL: https://towardsdatascience.com/market-mix-modeling-mmm-101-3d094df976f9.

52. An Analyst's Guide to MMM | Robyn. URL: https://facebookexperimental.github.io/Robyn/docs/analysts-guide-to-MMM/.

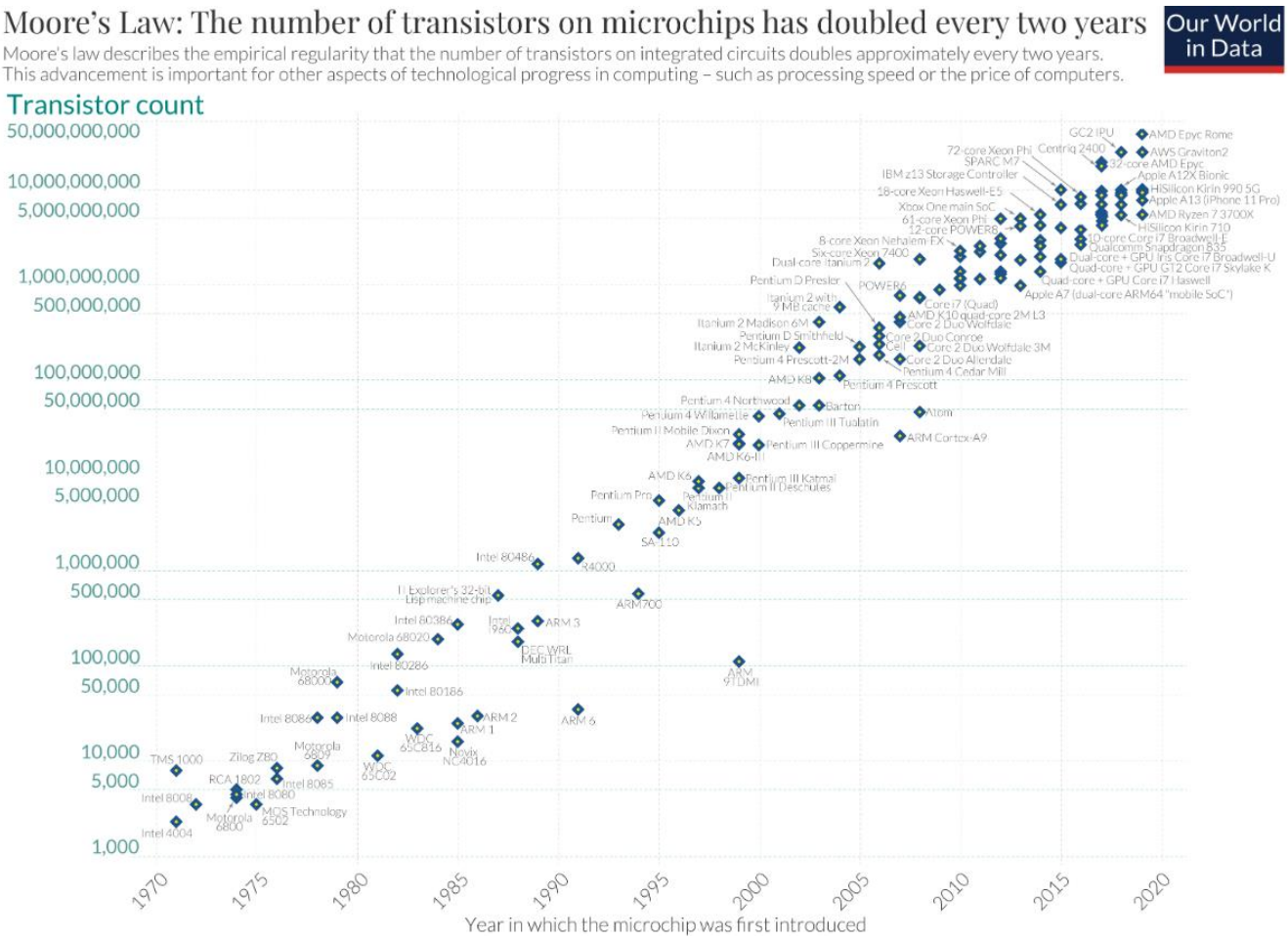
53. Prasad A. Regression Trees | Decision Tree for Regression | Machine Learning. Medium. URL: https://medium.com/analytics-vidhya/regression-trees-decision-tree-for-regression-machine-learning-e4d7525d8047.

54. Kotu V., Deshpande B. Autoregressive Integrated Moving Average Model. ScienceDirect. 2023. URL: https://www.sciencedirect.com/topics/mathematics/autoregressive-integrated-moving-average-model.

ДОДАТКИ

Додаток А.1

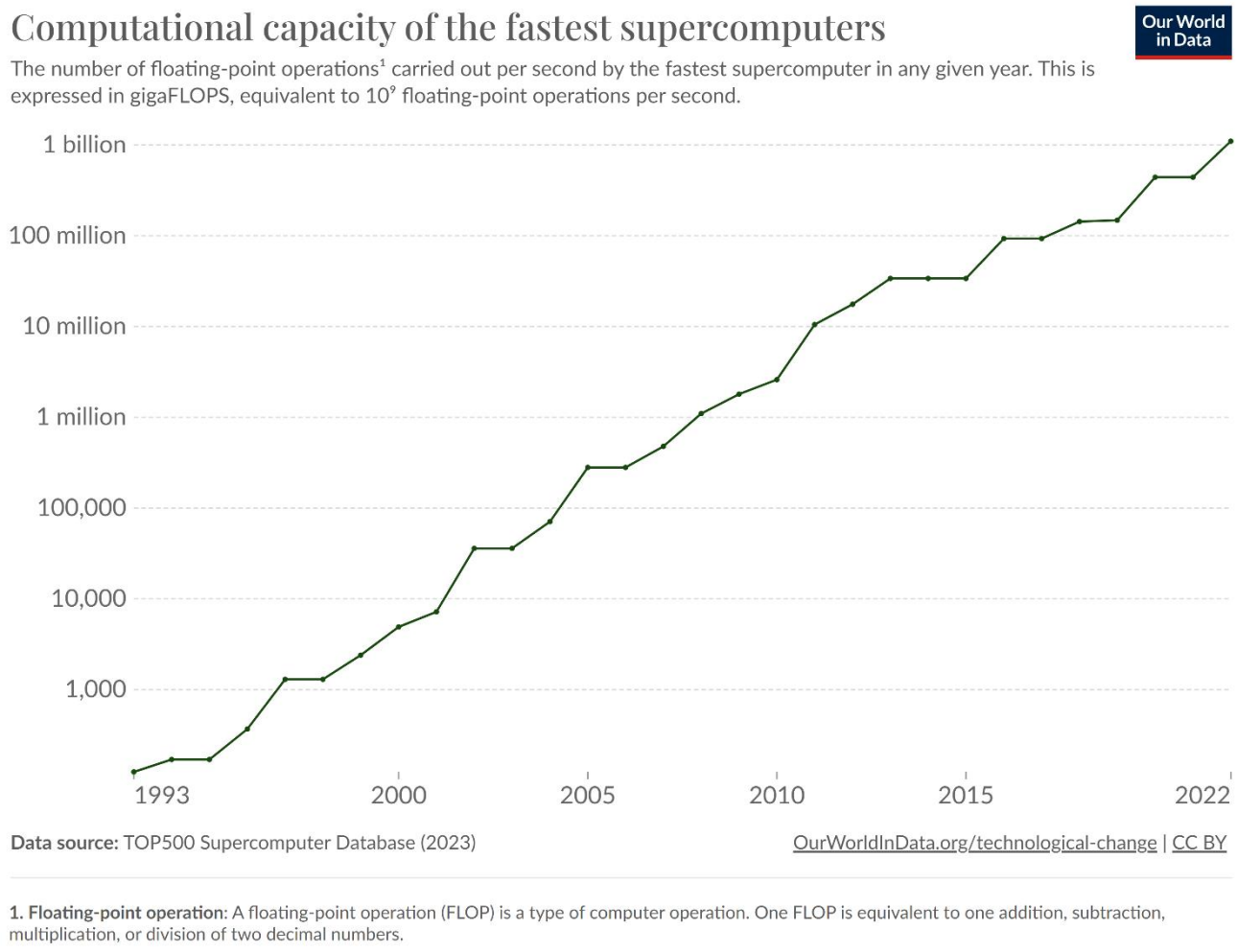
Динаміка кількості транзисторів, що можуть бути розміщені на одному мікрочіпі, за роками



Джерело: [2]

Додаток А.2

Динаміка обчислювальної спроможності найкращих суперкомп'ютерів, за роками



Джерело: [2]

UNICHECK
by Turnitin

Ім'я користувача:
Математичного моделювання та статистики Даценк...

ID перевірки:
1015968154

Дата перевірки:
04.12.2023 15:57:22 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
04.12.2023 16:33:19 EET

ID користувача:
100005744

Назва документа: Єгупов_Осипова_плагіат

Кількість сторінок: 94 Кількість слів: 20957 Кількість символів: 158489 Розмір файлу: 2.07 MB ID файлу: 1015646918

4.47%
Схожість

Найбільша схожість: 0.74% з Інтернет-джерелом (https://ela.kpi.ua/bitstream/123456789/26688/1/Timenko_magistr.pdf)

3.78% Джерела з Інтернету	286	Сторінка 96
3.93% Джерела з Бібліотеки	441	Сторінка 98

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	10