Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

“Інформаційна технологія прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки”

Виконав: студент 2 курсу, групи 2ІСТ-21м
спеціальності 126 – «Інформаційні системи та технології»
_____ Лопухов Б. Л.

Керівник: к.т.н., доц. каф. САІТ
_____ Козачко О. М.
«01» _____12_____ 2022 р.

Опонент: д.т.н., проф. каф. КСУ
_____ Ковтун В.В.
«16» _____12_____ 2022 р.

Допущено до захисту

Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«05» _____12_____ 2022 р.

Вінниця ВНТУ – 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – ІІ-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 126 Інформаційні системи та технології
Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«16» 09 2022 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Лопухову Богдану Леонідовичу

1. Тема роботи: "Інформаційна технологія прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки ",
керівник роботи: Козачко О.М. к.т.н., доцент каф. САІТ,
затверджені наказом закладу вищої освіти від «14» 09 2022 року №205
2. Строк подання студентом роботи «01» 12 2022 року
3. Вихідні дані до роботи:
Датасет Kaggle «COVID-19: Holidays of countries» з даними про хворих на коронавірус в США.
4. Зміст текстової частини:
   - аналіз предметної області;
   - вибір оптимальної інформаційної технології прогнозування кількості нових хворих;
   - реалізація інформаційної технології прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки;
   - економічна частина.
5. Перелік ілюстративного матеріалу:
   - блок-схема алгоритму роботи інформаційної технології;
   - графік критерія Оксфордської лабораторії Stringence index;
   - графік залежностей опадів і середньої температури;
   - графік кількість нових хворих у США з накладеними аномальними датами;
   - графік впливу свят на прогноз;
   - прогноз кількості нових хворих на коронавірус в проміжок довгого періоду часу.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Бурєннікова Н.В, д.е.н., проф. каф. ЕПВМ	05.11.22р.	25.11.22р.

7. Дата видачі завдання «16» 09 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МКР	Строк виконання етапів роботи	Примі
1	Аналіз предметної області	09.2022	
2	Вибір оптимальної інформаційної технології прогнозування кількості нових хворих	10.2022	
3	Розроблення інтелектуальної інформаційної технології	11.2022	
4	Економічна частина	11.2022	
5	Оформлення матеріалів до захисту МКР	11.2022	

Студент Лопухов Б. Л

Керівник роботи Козачко О. М.

АНОТАЦІЯ

УДК 004.8

Лопухов Б.Л. Інформаційна технологія прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2022. 94 с.

На укр. мові. Бібліогр.: 21 назв; рис.: 51; табл.: 5.

В магістерській кваліфікаційній роботі розроблено інформаційну технологію, яка прогнозує кількість нових хворих на коронавірус в Сполучених Штатах Америки. Вибрано оптимальні технології та за їх допомогою побудовано довгострокову модель прогнозу.

Ілюстративна частина складається з 6 плакатів.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження інформаційної технології прогнозування кількості нових хворих на коронавірус в Сполучених Штатах Америки.

Ключові слова: прогнозування, інформаційна технологія, коронавірус, Facebook Prophet, машинне навчання.

ABSTRACT

Lopukhov B.L. Information technology for forecasting the number of new coronavirus patients in the United States of America. Master's qualification thesis on specialty 126 - information systems and technologies, educational and professional program - information technologies of data and image analysis. Vinnytsia: VNTU, 2022. 94 p.

In Ukrainian speech Bibliography: 21 titles; Fig.: 51; tab.: 5.

The master's thesis developed the concept of information technology that predicts the number of new coronavirus patients in the United States of America. Optimal technologies were selected and a pilot version of the program was created with their help.

The illustrative part consists of 6 posters with the results of forecasting the number of new coronavirus patients in the United States of America.

The section of the economic part discusses the feasibility of developing and implementing information technology for forecasting the number of new coronavirus patients in the United States of America.

Keywords: forecasting, information technology, coronavirus, Facebook Prophet, machine learning.

ЗМІСТ

ВСТУП

Актуальність теми. В сучасному світі, коли інформаційні технології оточують нас в повсякденному житті з'являються великі масиви даних, на основі яких можна виявляти різні закономірності. Задача прогнозування стало досить вагомим у виявлені цих закономірностей.

Проблематика прогнозування і передбачення ймовірності є досить обговорюваною зараз. Актуальність цього процесу обґрунтована наявністю широкого кола завдань, для вирішення яких використовують наявну різноманітність способів прогнозування. Особливо в сьогоднішніх реаліях, коли весь світ уже кілька років знаходиться в умовах пандемії і щоб слідкувати за її розвитком і вводити заходи щоб зупинити розповсюдження коронавірусної інфекції величезну роль відіграє прогнозування. Тому актуальність теми роботи не викликає жодних питань.

Мета і завдання роботи. Підвищення точності прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки за рахунок використання методів машинного навчання.

Розробка інформаційної технології передбачає виконання наступних задач:

– аналіз предметної області;
– вибір оптимальних технологій;
– побудова довгострокової моделі прогнозу.

Об'єктом дослідження магістерської кваліфікаційної роботи є процес прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки, на основі датасету, що взяті з сервісу Kaggle.

Предметом дослідження магістерської кваліфікаційної роботи є технології та програмні засоби аналізу та прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки у 2022 р.

Новизна одержаних результатів. Подальшого розвитку набула інформаційна технологія прогнозування кількості нових хворих на

коронавірус у Сполучених Штатах, за рахунок використання моделі Facebook Prophet з урахуванням та регуляризацією аномальних даних, що дозволяє підвищити точність цього прогнозування

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи доповідались на всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (МН-2023).

Публікації результатів магістерської кваліфікаційної роботи. Опубліковано тези на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (МН-2023) [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні типи моделей поширення захворюваності

В сучасному світі неодноразово вчені знаходять чимало нових вірусів, з якими потім бореться людство та економіка. І цього разу знайдено вірус, який забрав життя мільйонів та наніс великий удар по економіці всіх країн світу. І тому вірусологам та епідеміологам було дане завдання прогнозувати поширення, розвиток та наслідок вірусних захворювань.

Цифрова епідеміологія – це галузь науки, яка показувала високі темпи розвитку у XX столітті через безперервне зростання кількості даних і обчислювальних потужностей, а також прориви в методах аналізу даних і мовах програмування та інформаційних технологіях. На сьогоднішній день технології, які використовується для прогнозування епідеміологічних процесів, є досить різноманітними. Враховуючи природу захворювання та довжину прогностичного вікна, це дозволяє підібрати широкий асортимент моделей для кожного конкретного випадку і, за необхідності, звузити до тих захворювань, які виявилися найбільш підходящими для цього типу. моделі.

1.2 Особливості захворювання та методи дослідження

Інфекція коронавірусу (COVID-19) – це захворювання, що викликається SARS-CoV-2, новим типом вірусу в родині коронавірусів. У грудні 2019 року в Ухані, Китай, було виявлено перший відомий випадок. Згідно з повідомленнями, вірус походить від тварин, можливо, від кажанів. Одна з імовірних причин поширення вірусу з впливом ринку морепродуктів, проте це джерело не доведене, передачі вірусу від людини до людини ще не встановлено. 11 березня Всесвітня організація охорони здоров’я (ВООЗ) підтвердила спалах коронавірусу як пандемію, назвавши його «глобальним поширенням нової хвороби» (рис. 1.1) [2].

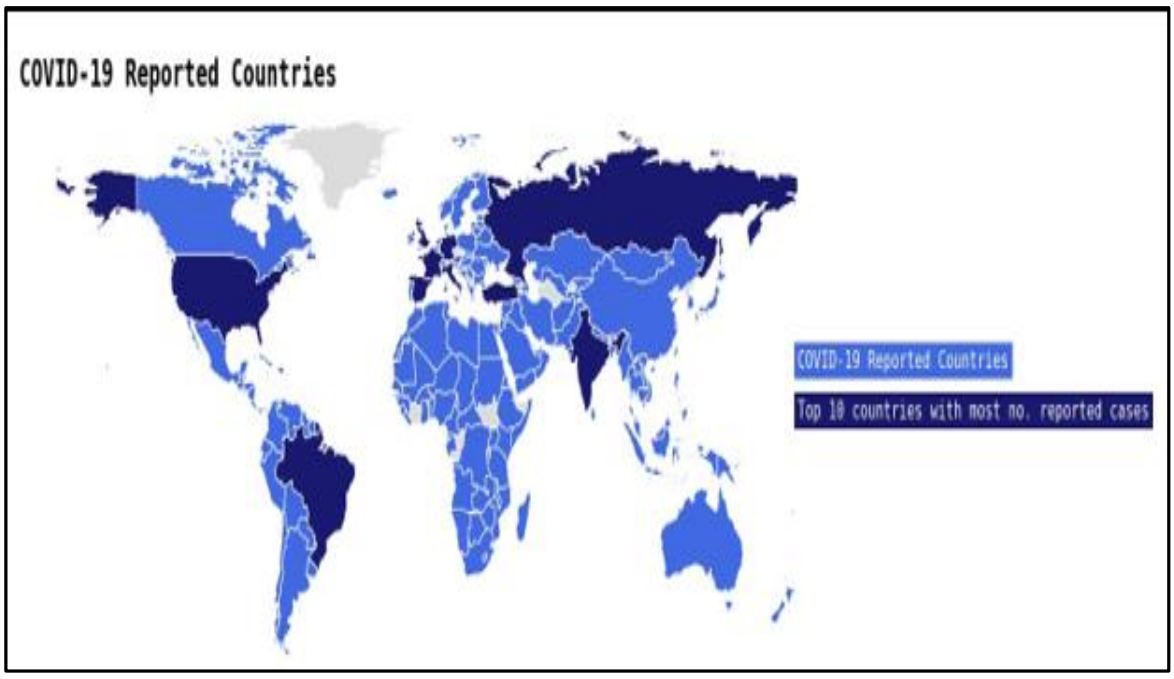

Рисунок 1.1 – Розповсюдження COVID-19 у світі

1.2.1 Огляд захворювання та його основних епідеміологічних ознак

Головний біль, втрата смаку або нюху, висипання або зміна кольору пальців рук або ніг. При складному процесі захворювання можуть виникати утруднення дихання або задишка, біль у грудях, афазія і навіть рух. У цьому випадку інфекція, викликана вірусом SARS-CoV-2, може викликати дихальну недостатність, септичний шок, органну недостатність і навіть смерть [3].

Важливими характеристиками захворювання є тривалість інкубаційного періоду, середній час зараження, відсоток смертей та відсоток людей, які одужали. В США смертність оцінюється в 1,1%, тоді як рівень смертності у світі в якийсь момент досяг 22%. Однак, рівень смертності є різним у різні хвилі захворюваності, залежно від типу штаму, який превалює.

Тривалість інкубаційного періоду – це період від моменту потрапляння інфекції в організм до появи перших симптомів інфекції. За оцінками, цей пробіл є суттєвим для ефективного впровадження карантинних обмежень. Дослідження показали, що його тривалість в основному становить від 2 до 14 днів, але також можуть бути випадки, коли перші симптоми з’являються від 0

до 27 днів після зараження. Тому цей параметр може відрізнятися від пацієнта до пацієнта. В середньому оцінюється 5,2 дня.

Інфекційний період – це час, протягом якого пацієнт може поширювати інфекцію. Цей параметр істотно впливає на правила ізоляції та виявлення. Середня тривалість COVID-19 становить 9,5 днів.

1.3 Статистика по коронавірусу в Сполучених Штатах Америки

Перший випадок появи коронавірусної хвороби було виявлено на території США 15 січня 2020 року. Станом на 20 березня 2020 року, виявлено понад 19000 випадків зараження, понад 100 хворих одужало, понад 200 людей померло. Але вже 26 березня Сполучені Штати Америки обійшли Китай і вийшли на перше місце за кількістю хворих , а 30 березня кількість захворілих на коронавірус зросла до 150 000.

– 28 квітня кількість випадків інфікування зросла до 1 млн;
– 8 червня кількість випадків інфікування зросла до 2 млн;
– 9 серпня кількість випадків інфікування перевищила 5 млн;
– 5 листопада в США вперше кількість нових інфікованих перевищила 100 тис осіб за добу, всього 9,4 млн.

На рисунку 1.2 зображено графік померлих, одужавших і хворіючих людей за два роки в США [3].

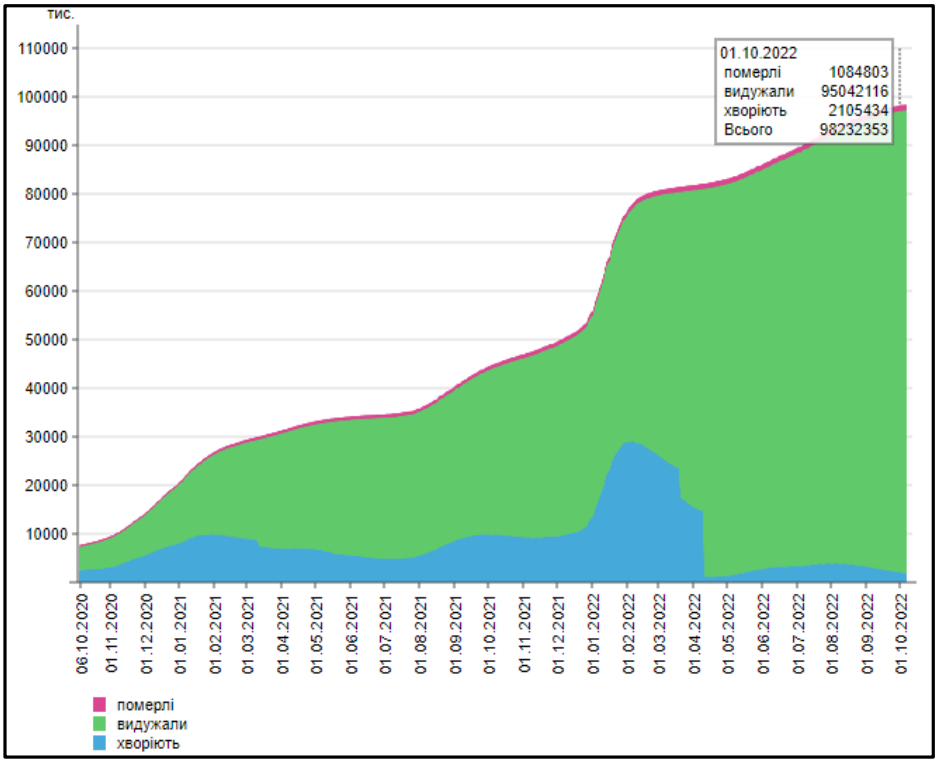

Рисунок 1.2 – Статистика по кількості померлих, одужавших та хворих людей за 2 роки в США

На рисунку 1.3 зображено графік кількості інфікованих людей за два роки в США.

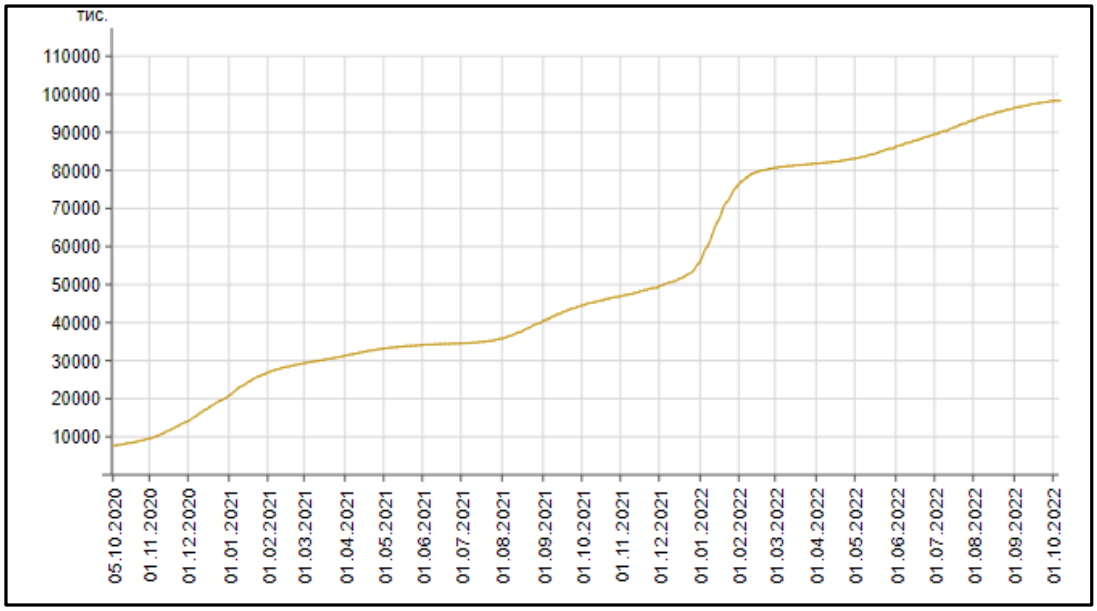

Рисунок 1.3 – Графік кількості інфікованих людей за 2 роки в США [3]

На рисунку 1.4 зображено графік кількості померлих людей за два роки в США.

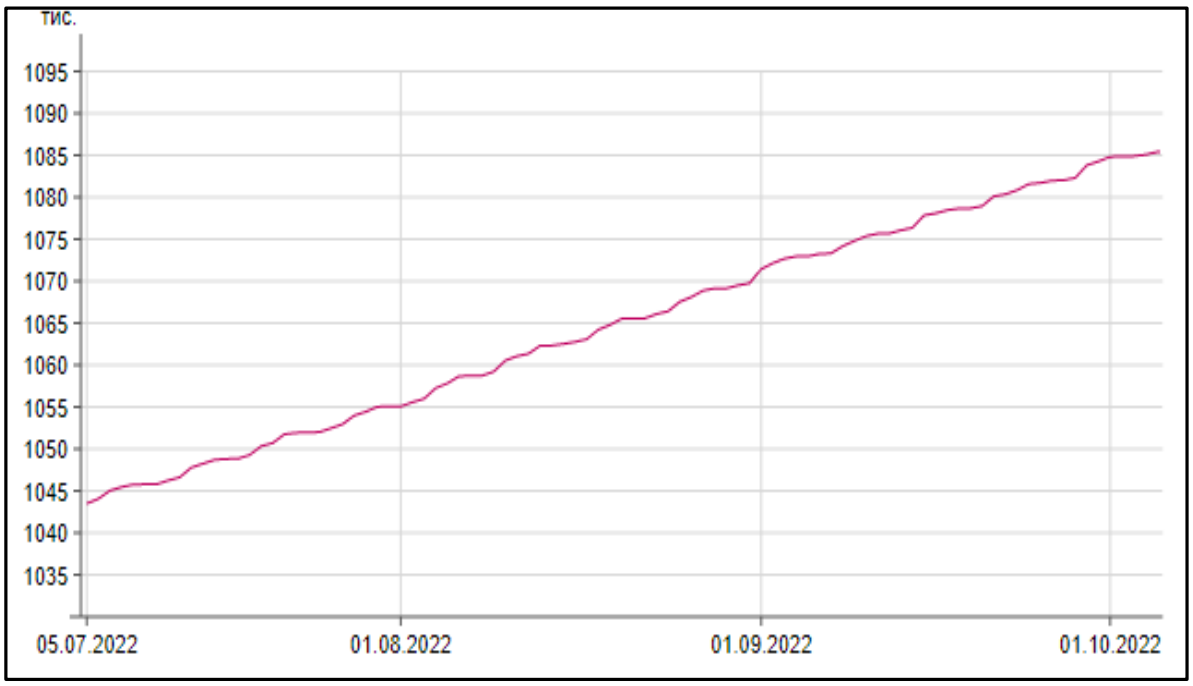

Рисунок 1.4 – Графік кількості померлих людей за 2 роки у США [3]

На рисунку 1.5 зображено графік кількості хворіючих людей за два роки у США.

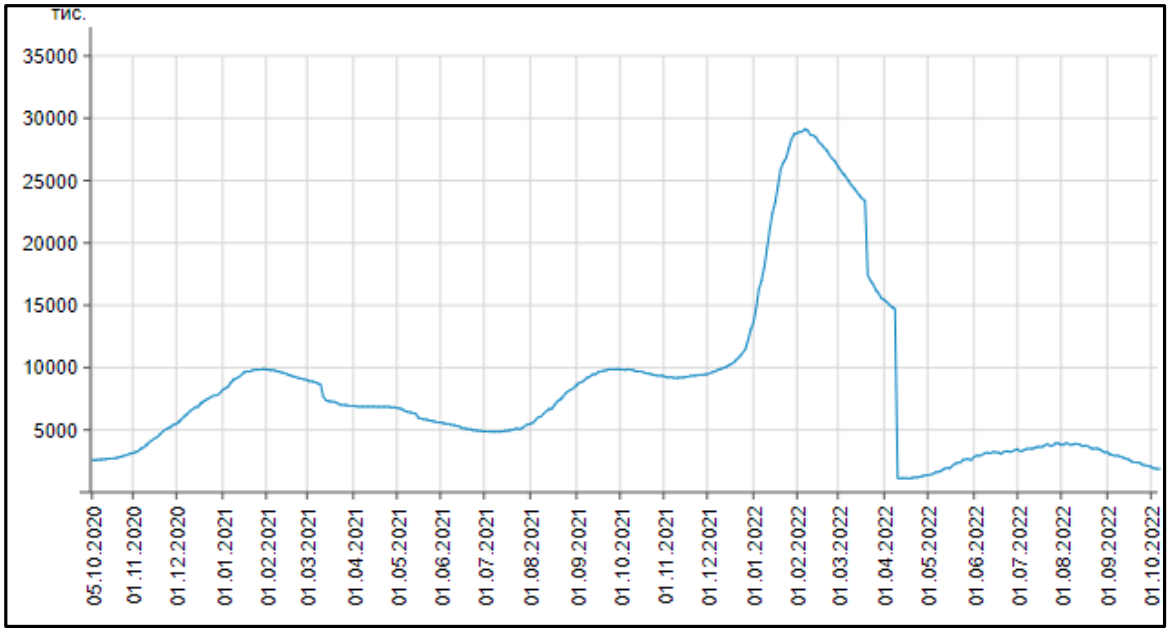

Рисунок 1.5 – Графік кількості хворіючих людей за 2 роки в США [3]

На рисунку 1.6 зображено графік кількості видужавших людей за два роки в США.

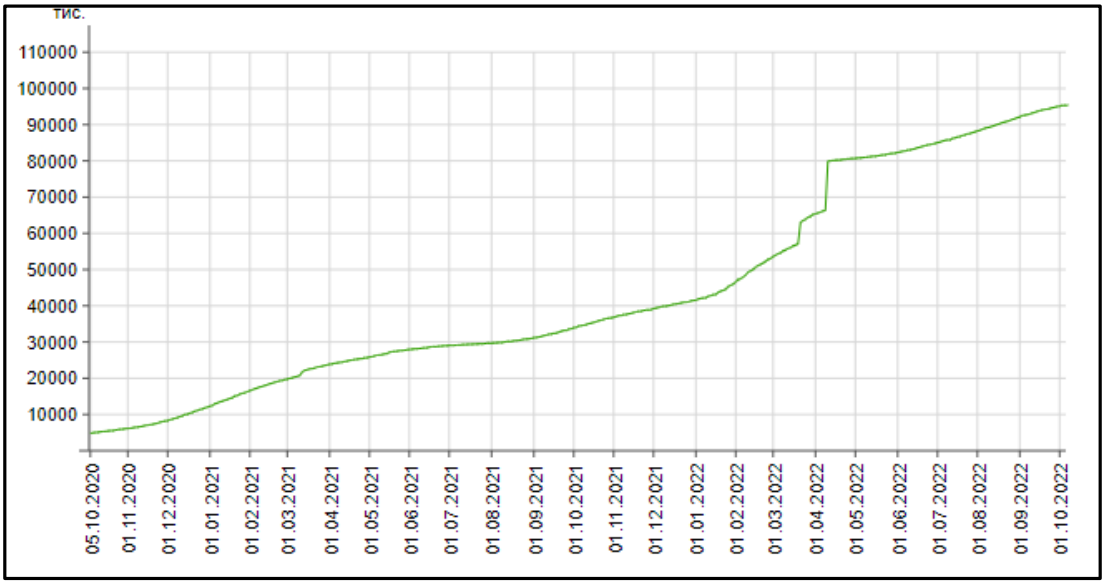

Рисунок 1.6 – Графік кількості одужавших людей за 2 роки в США [3]

На рисунку 1.7 зображено графік порівняння кількості підтверджених випадків захворювання на коронавірус із кількістю госпіталізованих людей США.

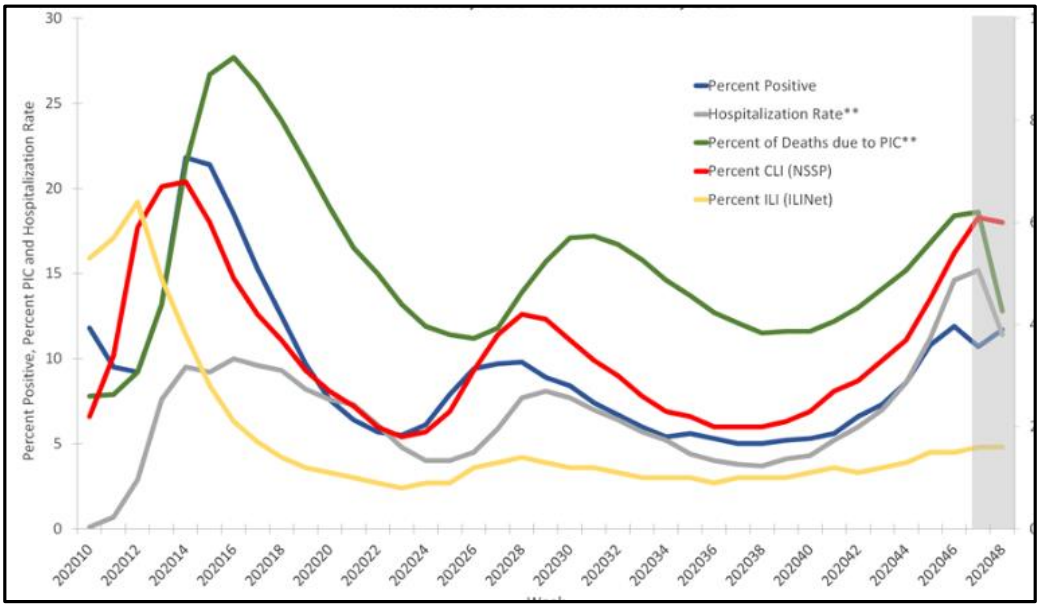

Рисунок 1.7 – Графік порівняння кількості госпіталізованих і підтверджених випадків [3]

На рисунку 1.8 зображено графік кількості підтверджених випадків захворювання на коронавірус по регіонам США.

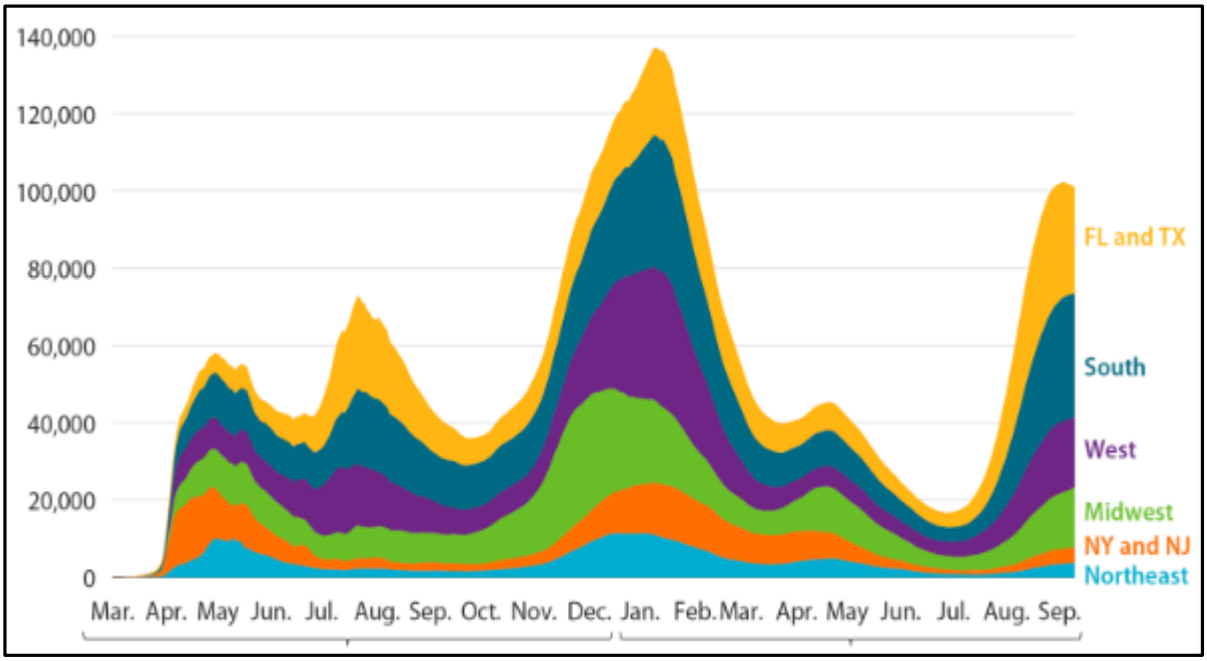

Рисунок 1.8 – Кількість хворих по регіонам [3]

На рисунках 1.9 і 1.10 показано кількість захворівших і кількість померлих по штатах відповідно. Як можем помітити ці показники майже прямо пропорціональні [3].

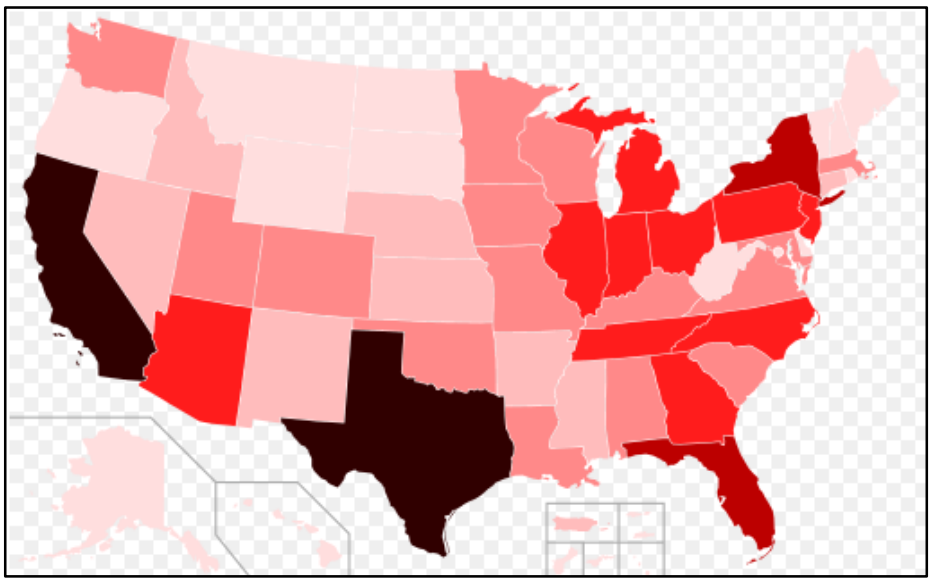

Рисунок 1.9 – Кількість хворих по штатах [3]

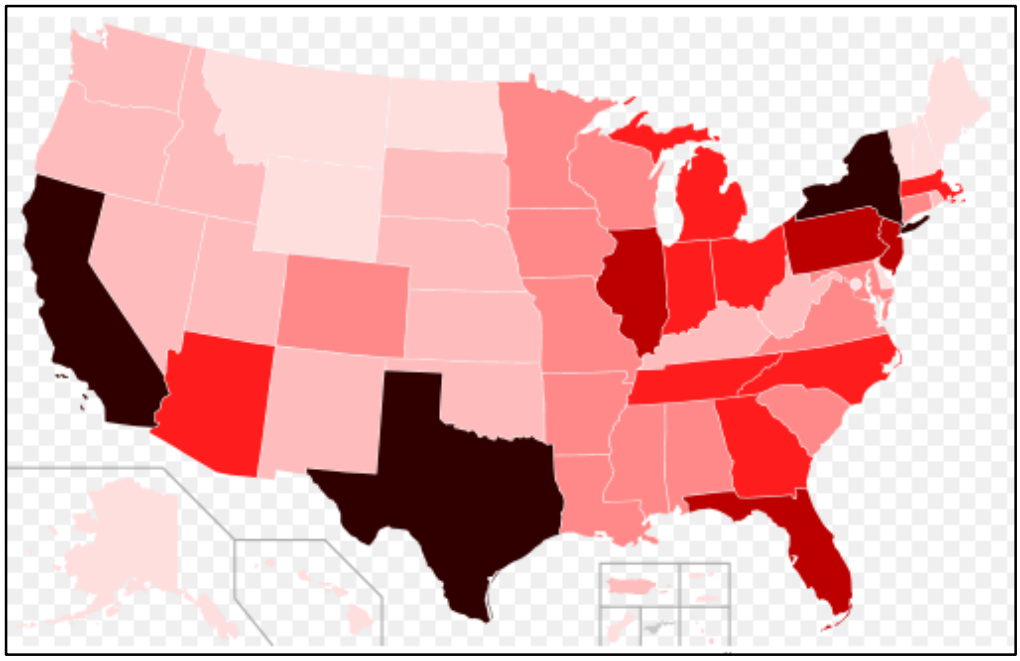

Рисунок 1.10 – Кількість померлих по штатах [3]

На рисунку 1.11 зображено графік порівняння кількості хворих на коронавірус із кількістю смертей в США. Кількість хворих (синя лінія) і кількість смертей (червона).

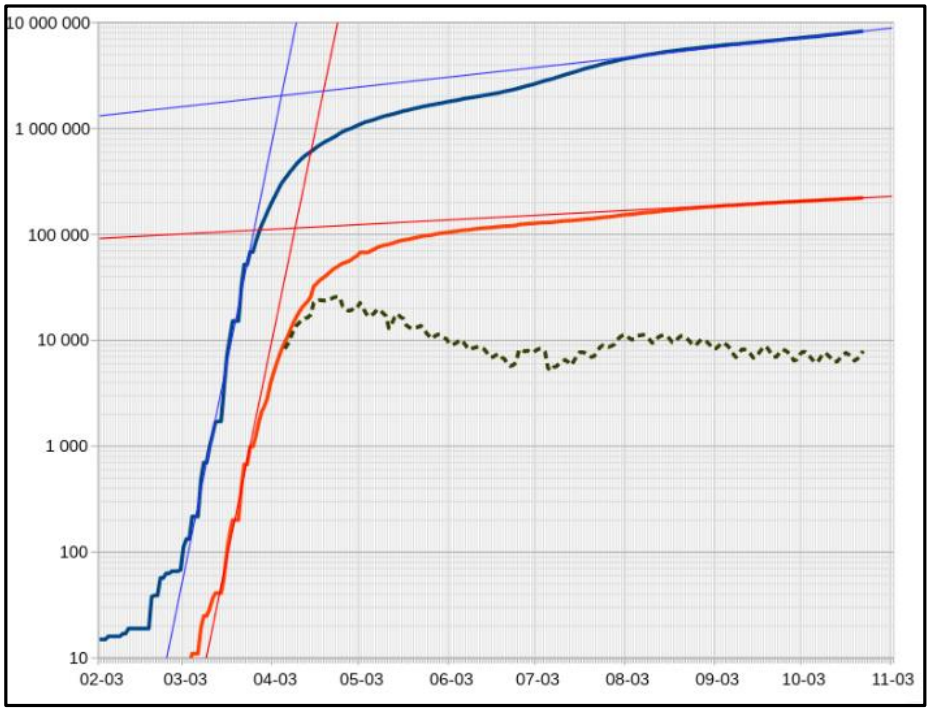

Рисунок 1.11 – Графік порівняння кількості смертей від кількості хворих на коронавірус

1.4 Огляд існуючих методів вирішення технічної проблеми

Головним виконавчим органом України, який контролює ситуацію з коронавірусної пандемією являється Національна служба здоров'я України. Вона проводить заходи по запобіганню розповсюдження коронавірусної інфекції.

У квітні 2020 р. міжвідомча робоча група враховуючи міжнародний досвід математичного моделювання розвитку епідемії коронавірусу, взявши за основу статистичні дані про динаміку епідемії в Україні та Європи – створювала і протестувала математичну модель SEIR–U [4].

Викладачі Вінницького національного технічного університету розробили інформаційну технологію аналізу та прогнозування кількості нових підтверджених випадків захворювань на коронавірус викликану інфекцією SARS-CoV-2, та випробували її на прикладі щодобових сумарних по Україні даних щодо однієї найбільшої «хвилі» з урахуванням аномальних дат (свят і псевдо свят), які можуть мати вагомий вплив [5].

Моделі побудовані даною технологією показали чудові результати як для США, так і для інших країн світу. Однак через те що ділянка сталого зростання закінчилась. Надалі спостереження показали неможливість даної моделі до прогнозування, крива досягла локальний максимум, відбувся спад хворих. Модель перестала бути ефективною, оскільки будувалась тільки на ділянці приросту. Також були ще дві хвилі з невеликим локальним максимумом проте, впродовж року дана модель не могла моделювати весь ряд у ретроспективі [6]. Вирішенням було використати моделі, в яких враховується багато хвильова природа процесу. Однак точність моделювання виявилась низька, в першу чергу, через не дуже високу адекватність і точність базової моделі, для описування основного тренду, який накладається на ідентифіковані хвилі [7].

В більшості моделей для опису хвиль використовували ряди Фур'є — цими рядами моделює різні сезонні складові і модель Фейсбуку —Prophet[8].

Отже, перед нами постає задача удосконалити створену раніше інформаційну технологію, на основі моделі Facebook Prophet описану у статті і описати багато хвильову природу часового ряду, для підвищення точності прогнозування кількості нових підтверджених випадків захворювань на коронавірус COVID-19 у Сполучених Штатах Америки.

1.5 Висновки

В даному розділі розглянуто ситуацію з коронавірусом у світі. Проаналізовано статистичні дані по кількості нових хворих та інших показниках. Крім того розглянуто деякі з найбільш поширених методів прогнозування інфекційних захворювань, проаналізовано їх переваги та недоліки. Вирішено удосконалити створену раніше інформаційну технологію, на основі моделі Facebook Prophet, щоб забезпечити прийнятний рівень точності.

2 ВИБІР ОПТИМАЛЬНОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ КІЛЬКОСТІ НОВИХ ХВОРИХ

2.1 Огляд вхідних даних

Для оптимального вибору засобів реалізації технології необхідно зробити огляд вхідних даних. Вхідними даними для виконання магістерської роботи є набір даних зі статистикою по кількості нових хворих у Сполучених Штатах Америки. Вони були завантажені з “Github” (рис. 2.1).

```
# Thanks https://github.com/CSSEGISandData/COVID-19
myfile = requests.get('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_
open('data', 'wb').write(myfile.content)
confirmed_global_df = pd.read_csv('data')
confirmed_global_df
```

Рисунок 2.1 – Завантаження даних

Також в цей же ноутбук будуть додані аномальні дати для покращення точності нашої моделі. Приклади деяких з них можна побачити на рисунку 2.2.

	ds holidays	holiday	ds	country	code	country official name	lower window	upper window	prior scale	source	ds dt
1	2020-05-25	Memorial Day	2020-06-01	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2020-06-01
34	2020-06-15	the weakening of quarantine	2020-06-22	United States	US	United States of America	-1	1	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-06-22
3	2020-07-03	Independence Day (Observed)	2020-07-10	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2020-07-10
2	2020-07-04	Independence Day	2020-07-11	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2020-07-11
22	2020-07-20	the weakening of quarantine	2020-07-27	United States	US	United States of America	-1	1	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-07-27
4	2020-09-07	Labor Day	2020-09-14	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2020-09-14
24	2020-09-12	the weakening of quarantine	2020-09-19	United States	US	United States of America	-1	1	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-09-19
5	2020-10-12	Columbus Day	2020-10-19	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2020-10-19
25	2020-10-26	the weakening of quarantine	2020-11-02	United States	US	United States of America	-1	1	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-11-02
6	2020-11-11	Veterans Day	2020-11-18	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2020-11-18
7	2020-11-26	Thanksgiving	2020-12-03	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2020-12-03
23	2020-12-02	the weakening of quarantine	2020-12-09	United States	US	United States of America	-1	1	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-12-09
8	2020-12-25	Christmas Day	2021-01-01	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2021-01-01
9	2021-01-01	New Year's Day	2021-01-08	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2021-01-08
11	2021-01-18	Martin Luther King Jr. Day	2021-01-25	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2021-01-25
30	2021-02-03	the weakening of quarantine	2021-02-10	United States	US	United States of America	-1	1	10	https://www.bsg.ox.ac.uk/research/research-pro...	2021-02-10
12	2021-02-15	Washington's Birthday	2021-02-22	United States	US	United States of America	-1	1	10	https://github.com/dr-prodigy/python-holidays	2021-02-22

Рисунок 2.2 – Приклади аномальних дат

2.2 Огляд методу вирішення поставленої задачі

Машинне навчання (ML) — це тип штучного інтелекту (ШІ), який дозволяє програмним програмам точніше прогнозувати результати без явного програмування для цього. Алгоритми машинного навчання використовують історичні дані як вхідні дані для прогнозування нових вихідних значень. Завданням машинного навчання це навчити машини виконувати завдання надавши їм пару прикладів (як виконувати завдання чи не виконувати його) [9].

Механізми рекомендацій є поширеним випадком використання машинного навчання. Інші популярні способи використання включають протидія шахраям, фільтрування спаму електронної пошти, виявлення загроз зловмисного програмного забезпечення, автоматизацію бізнес-процесів (BPA) і прогнозне обслуговування.

Машинне навчання спрямоване на розробку комп'ютерних програм, що можуть отримувати доступ до даних та використовувати їх для самостійного навчання.

Історія розвитку машинного навчання розвивалась в такій послідовності:

– 1642 — Блез Паскаль винайшов механічну машину, яка може складати, віднімати, множити і ділити;

– 1679 — Готфрід Вільгельм Лейбніц винайшов систему двійкового коду;

– 1834 — Чарльз Беббідж придумав ідею загального універсального пристрою, який можна було б програмувати за допомогою перфокарт;

– 1842 — Ада Лавлейс описує послідовність операцій для вирішення математичних задач за допомогою теоретичної перфокартової машини Чарльза Беббіджа та стає першим програмістом;

– 1847 – Джордж Буль створює булеву логіку, форму алгебри, у якій усі значення можна звести до двійкових значень істини чи хибності;

– 1936 — англійський логік і криптоаналітик Алан Тьюрінг запропонував універсальну машину, яка могла б розшифровувати та виконувати набір інструкцій. Його опублікований доказ вважається основою інформатики;

– 1952 — Артур Самуель створює програму, яка допомагає комп'ютеру IBM ставати кращим у шашках, чим більше він грає;

– 1959 — MADALINE стає першою штучною нейронною мережею, застосованою до реальної проблеми: видалення відлуння від телефонних ліній;

– 1985 – Штучна нейронна мережа Террі Сейновскі та Чарльза Розенберга навчилася правильно вимовляти 20 000 слів за один тиждень.

– 1997 — Deep Blue від IBM переміг шахового гросмейстера Гаррі Каспарова;

– 1999 рік – прототип інтелектуальної робочої станції CAD перевірив 22 000 мамографів і виявив рак на 52% точніше, ніж це зробили рентгенологи;

– 2006 — комп'ютерний науковець Джеффрі Хінтон винаходить термін глибоке навчання для опису дослідження нейронної мережі;

– 2012 — Неконтрольована нейронна мережа, створена Google, навчилася розпізнавати котів у відео YouTube з точністю 74,8%;

– 2014 — Чат-бот пройшов тест Тьюрінга, переконавши 33% суддів, що це був український підліток на ім'я Євген Густман;

– 2014 — AlphaGo від Google перемагає чемпіона-людина в Go, найскладнішій настільній грі у світі;

– 2016 — LipNet, система штучного інтелекту DeepMind, визначає у відео слова, які читаються з губ, із точністю 93,4%;

– 2019 – Amazon контролює 70% частки ринку віртуальних помічників у США.

Машинне навчання важливе, оскільки воно дає підприємствам уявлення про тенденції в поведінці клієнтів і бізнес-моделях, а також підтримує

розробку нових продуктів. Багато провідних сучасних компаній, таких як Facebook, Google і Uber, роблять машинне навчання центральною частиною своєї діяльності. Машинне навчання стало значним конкурентоспроможним фактором для багатьох компаній.

За останні кілька десятиліть технологічний прогрес у сховищах і обчислювальних потужностях дозволив створити деякі інноваційні продукти на основі машинного навчання, такі як система рекомендацій Netflix і безпілотні автомобілі.

Машинне навчання є невід'ємною частино стрімко розвиваючої галузі науки про дані. Через використання статистичних методів алгоритми, розвиваються. навчаються робити прогнози, а також виявляти основні ідеї в проектах інтелектуального аналізу даних. Ця інформація з часом керує прийняттям рішень у програмах чи бізнесі, ідеально впливаючи на ключові показники зростання. Оскільки великі дані продовжують збільшуватися та зростати, ринковий попит на науковців з обробки даних зростатиме. Вони повинні будуть допомогти визначити найбільш релевантні бізнес-питання та дані для відповіді на них [10].

Види машинного навчання зображені на рисунку 2.3 [10].

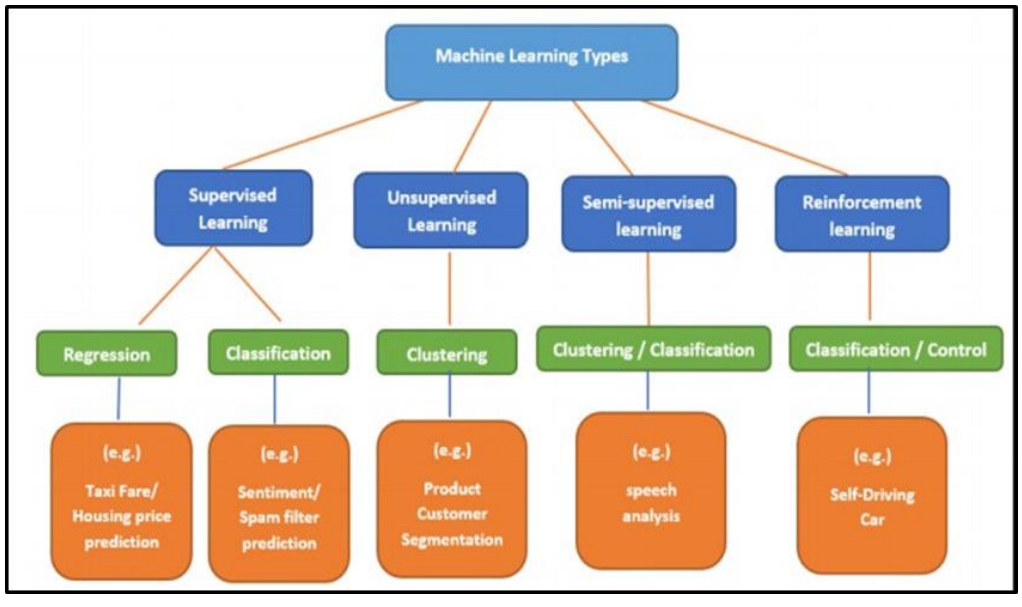

Рисунок 2.3 – Види машинного навчання

Машинне навчання розділяють за тим, як алгоритм вчиться ставати точнішим у своїх прогнозах. Існує чотири основні підходи: навчання під контролем, навчання без контролю, навчання з напівконтролем і навчання з підкріпленням. Тип алгоритму, слід вибирати на основі даних які потрібно передбачити.

Сьогодні машинне навчання використовується в широкому діапазоні програм. Мабуть, одним із найвідоміших прикладів машинного навчання в дії є механізм рекомендацій, який підтримує стрічку новин Facebook.

Facebook використовує машинне навчання, щоб персоналізувати спосіб доставки каналу кожного учасника. Якщо учасник часто зупиняється, щоб прочитати дописи певної групи, система рекомендацій почне відображати більше активності цієї групи раніше в стрічці.

За лаштунками механізм намагається посилити відомі шаблони поведінки учасників в Інтернеті. Якщо учасник змінить шаблони та не зможе прочитати дописи з цієї групи протягом наступних тижнів, стрічка новин буде відповідним чином скоригована.

Крім механізмів рекомендацій, машинне навчання можна використовувати в наступному:

– управління взаємовідносинами з клієнтами. Програмне забезпечення CRM може використовувати моделі машинного навчання для аналізу електронної пошти та спонукати членів відділу продажів першими відповідати на найважливіші повідомлення. Досконаліші системи можуть навіть рекомендувати потенційно ефективні відповіді;

– бізнес-аналітика. Постачальники BI та аналітики використовують машинне навчання у своєму програмному забезпеченні для виявлення потенційно важливих точок даних, шаблонів точок даних і аномалій;

– інформаційні системи кадрового забезпечення. Системи HRIS можуть використовувати моделі машинного навчання для фільтрації програм і визначення найкращих кандидатів на відкриту вакансію;

– безпілотні автомобілі. Алгоритми машинного навчання можуть

навіть дозволити напівавтономному автомобілю розпізнати частково видимий об’єкт і попередити водія;

– віртуальні помічники. Розумні помічники зазвичай поєднують контрольовані та неконтрольовані моделі машинного навчання для інтерпретації природного мовлення та надання контексту.

Машинне навчання бачило випадки використання, починаючи від прогнозування поведінки клієнтів і закінчуючи формуванням операційної системи для самокерованих автомобілів.

Коли справа доходить до переваг, машинне навчання може допомогти підприємствам глибше зрозуміти своїх клієнтів. Збираючи дані про клієнтів і співвідносячи їх із поведінкою з часом, алгоритми машинного навчання можуть вивчати асоціації та допомагати командам пристосовувати розробку продукту та маркетингові ініціативи до попиту клієнтів.

Деякі компанії використовують машинне навчання як основний двигун у своїх бізнес-моделях. Uber, наприклад, використовує алгоритми для підбору водіїв і пасажирів. Google використовує машинне навчання, щоб показувати рекламу поїздок у пошукових запитах.

Але машинне навчання має недоліки. По-перше, це може бути дорого. Проектами машинного навчання зазвичай керують спеціалісти з обробки даних, які отримують високу зарплату. Ці проекти також потребують програмної інфраструктури, яка може бути дорогою.

Існує також проблема упередженості машинного навчання. Алгоритми, навчені на наборах даних, які виключають певні популяції або містять помилки, можуть призвести до неточних моделей світу, які, в кращому випадку, будуть невдалими, а в гіршому – дискримінаційними. Коли підприємство базує основні бізнес-процеси на упереджених моделях, воно може завдати шкоди регуляторним органам і репутації.

Як вибрати правильну модель машинного навчання?

Процес вибору правильної моделі машинного навчання для вирішення проблеми може зайняти багато часу, якщо до нього не підходити стратегічно.

Крок 1. Узгодимо проблему з потенційними вхідними даними, які слід враховувати для вирішення.

Крок 2. Зберемо дані, відформатуємо їх, якщо необхідно.

Крок 3. Оберемо, який алгоритм використовувати та перевіримо, щоб побачити, наскільки добре вони працюють.

Крок 4. Продовжимо точно налаштовувати вихідні дані, поки вони не досягнуть прийнятного рівня точності.

Як правило машинне навчання дозволяє проаналізувати гігантську кількість даних. І це забезпечує швидші та точніші результати для виявлення вигідних можливостей або ризиків, однак для хорошої підготовки може знадобитися додатковий час та ресурси. Поєднання машинного навчання з когнітивними технологіями може зробити його ще більш ефективним при обробці великих масивів даних [11].

Для прогнозування кількості нових хворих на коронавірус доцільно застосовувати часові ряди.

Часовий ряд — це послідовність точок даних, які виникають у послідовному порядку протягом певного періоду часу. Це можна порівняти з перехресними даними, які фіксують момент часу.

Під час інвестування часовий ряд відстежує рух вибраних точок даних, таких як ціна цінного паперу, протягом визначеного періоду часу з точками даних, що записуються через регулярні проміжки часу. Немає мінімального чи максимального часу, який необхідно включити, щоб дані могли бути зібрані таким чином, щоб надати інформацію, яку шукає інвестор або аналітик, який вивчає діяльність.

Часовий ряд можна взяти для будь-якої змінної, яка змінюється з часом. В інвестуванні зазвичай використовують часові ряди для відстеження ціни цінних паперів у часі. Це можна відстежувати в короткостроковій перспективі, як-от ціна цінного паперу за годину протягом робочого дня, або в довгостроковій перспективі, наприклад ціна цінного паперу на момент закриття в останній день кожного місяця протягом курс п'ять років.

Аналіз часових рядів може бути корисним, щоб побачити, як певний актив, цінний папір або економічна змінна змінюється з часом. Його також можна використовувати, щоб перевірити, як зміни, пов’язані з вибраною точкою даних, порівнюються зі змінами в інших змінних за той самий період часу [12].

Часові ряди також використовуються в кількох нефінансових контекстах, таких як вимірювання зміни чисельності населення з часом. На малюнку нижче зображено такий часовий ряд для зростання населення США протягом століття з 1900 по 2000 рік (рис. 2.4) [12].

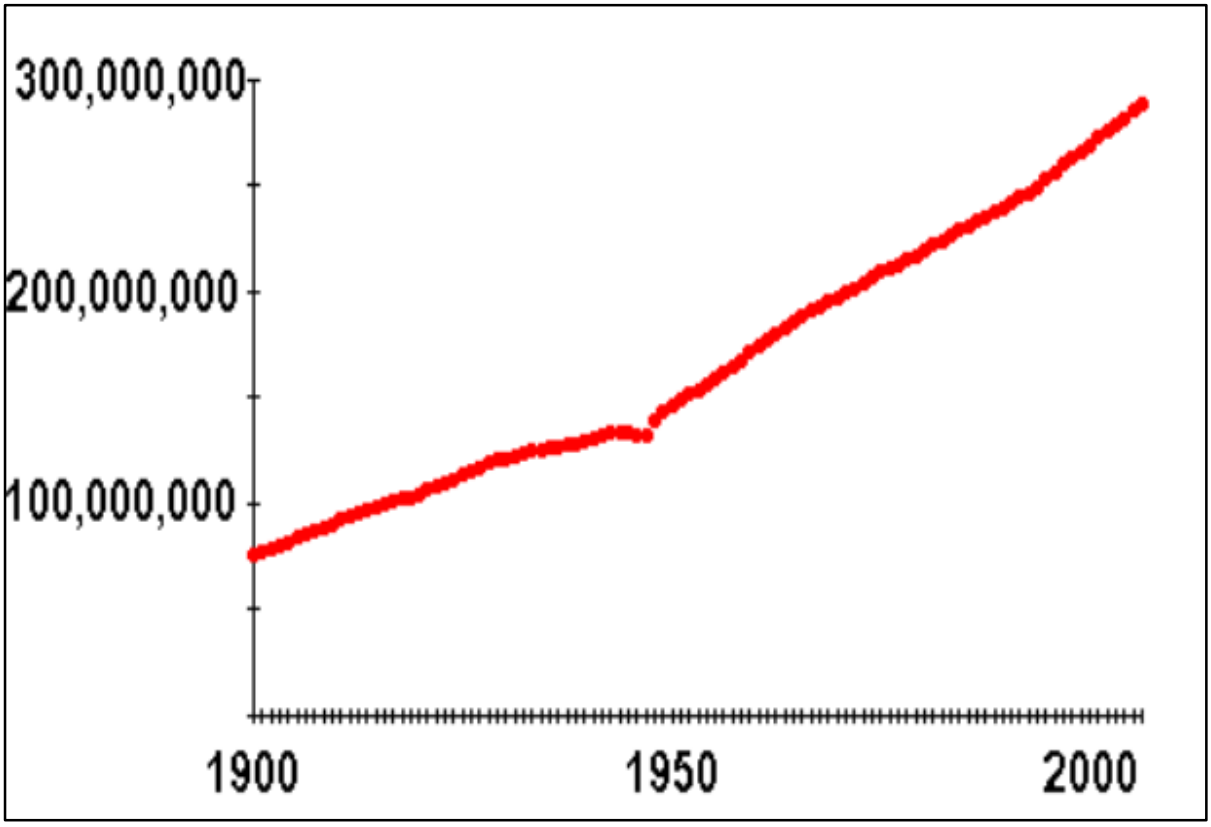

Рисунок 2.4 – Часовий ряд для зростання населення

Припустімо, що ви хочете проаналізувати часовий ряд щоденних цін на акцію закриття певної акції протягом одного року. Ви отримаєте список усіх цін закриття акцій кожного дня за минулий рік і перерахуєте їх у хронологічному порядку. Це однорічний щоденний часовий ряд ціни закриття для акцій.

Заглибившись трохи глибше, ви можете проаналізувати дані часових рядів за допомогою інструментів технічного аналізу, щоб дізнатися, чи показують часові ряди акцій будь-яку сезонність. Це допоможе визначити, чи запаси проходять через піки та спади в звичайний час кожного року. Аналіз у цій сфері вимагав би взяти спостережувані ціни та спів віднести їх із обраним сезоном. Це може включати традиційні календарні сезони, як-от літо та зима, або сезони роздрібної торгівлі, як-от святкові сезони.

Крім того, ви можете записати зміни ціни акцій, оскільки вони пов’язані з економічною змінною, такою як рівень безробіття. Співвідносячи точки даних з інформацією, що стосується вибраної економічної змінної, ви можете спостерігати закономірності в ситуаціях, демонструючи залежність між точками даних і вибраною змінною.

Прогнозування часових рядів використовує інформацію про історичні значення та пов’язані закономірності для прогнозування майбутньої діяльності. Найчастіше це стосується аналізу трендів, аналізу циклічних коливань і питань сезонності. Як і з усіма методами прогнозування, успіх не гарантований.

Модель Бокса-Дженкінса, наприклад, є технікою, розробленою для прогнозування діапазонів даних на основі вхідних даних із визначеного часового ряду. Він прогнозує дані за трьома принципами: авторегресії, диференціації та ковзних середніх. Ці три принципи відомі як p, d і q відповідно. Кожен принцип використовується в аналізі Бокса-Дженкінса, і разом вони представлені як авторегресійне інтегроване ковзне середнє або ARIMA (p, d, q). ARIMA можна використовувати, наприклад, для прогнозування цін на акції або зростання прибутків.

Інший метод, відомий як аналіз зміненого діапазону, можна використовувати для виявлення та оцінки ступеня стійкості, випадковості або повернення середнього значення в даних часових рядів. Перемасштабований діапазон можна використовувати для екстраполяції майбутнього значення або середнього значення для даних, щоб побачити, чи тенденція є стабільною чи

ймовірно зміниться.

Перехресний аналіз є одним із двох основних методів порівняння для аналізу запасів. Перехресний аналіз розглядає дані, зібрані в один момент часу, а не протягом певного періоду часу. Аналіз починається з встановлення цілей дослідження та визначення змінних, які аналітик хоче виміряти. Наступним кроком є визначення перерізу, наприклад, групи однолітків або галузі, і встановлення конкретного моменту часу, який оцінюється. Останній крок полягає в проведенні аналізу на основі перерізу та змінних і приході до висновку щодо ефективності компанії чи організації. По суті, міжсекційний аналіз показує інвестору, яка компанія найкраще підійде з урахуванням показників, які їх цікавлять.

Аналіз часових рядів, відомий як аналіз тенденцій, коли він стосується технічної торгівлі, зосереджується на одному цінному папері протягом певного часу. У цьому випадку ціна оцінюється в контексті його минулих показників. Аналіз часових рядів показує інвестору, чи працює компанія краще чи гірше, ніж раніше, за показниками, які їх цікавлять. Часто це будуть класичні показники, такі як прибуток на акцію (EPS), борг до власного капіталу, вільний грошовий потік (FCF) тощо. На практиці інвестори зазвичай використовують комбінацію аналізу часових рядів і перехресного аналізу, перш ніж приймати рішення, наприклад, переглядаючи прибуток на акцію за певний час, а потім перевіряючи порівняльний прибуток на акцію в галузі

Часові ряди можна використовувати для:

– відстеження даних про погоду;
– відстеження змін у продуктивності програми;
– медичні вироби для візуалізації життєво важливих елементів у реальному часі;
– відстеження журналів мережі.

Погодні записи, економічні показники та показники розвитку здоров’я пацієнтів - все це дані часових рядів. Дані часових рядів також можуть бути метриками сервера, моніторингом продуктивності додатків, мережевими

даними, датчиками даних, подіями, кліками та багатьма іншими типами аналітичних даних.

Зверніть увагу, як час - зображений на рисунку 2.5 - є віссю [13].

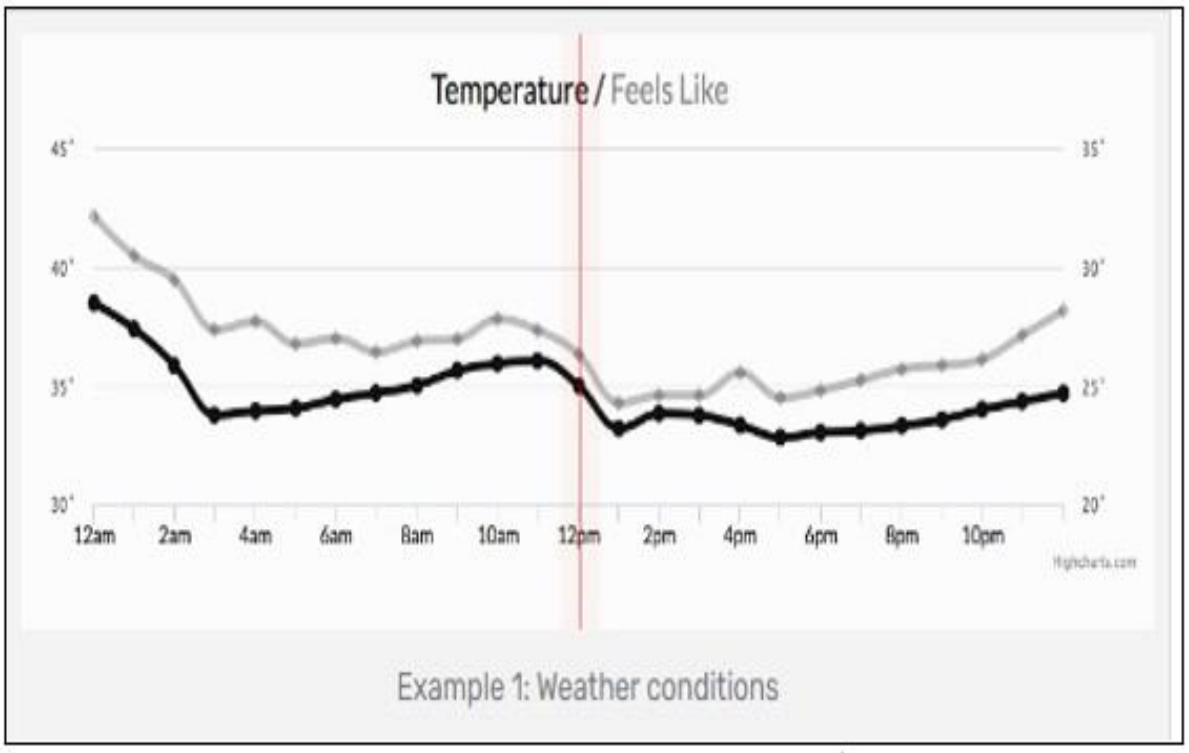

Рисунок 2.5 – Часовий ряд

Дані часових рядів можна класифікувати на два типи:

– вимірювання, зібрані через регулярні інтервали часу (показники);
– вимірювання, зібрані через нерегулярні інтервали часу (події).

На рисунку 2.6 зображено приклад показників та події. Приклади з синьою лінією відображають показники. Приклади з червоною лінією зображують події. Через те що вони відбуваються через нерегулярні відрізки часу, події є досить непередбачувані, і їх неможливо змоделювати чи спрогнозувати, адже прогнозування передбачає, що все, що сталося в минулому, є хорошим показником того, що буде в майбутньому.

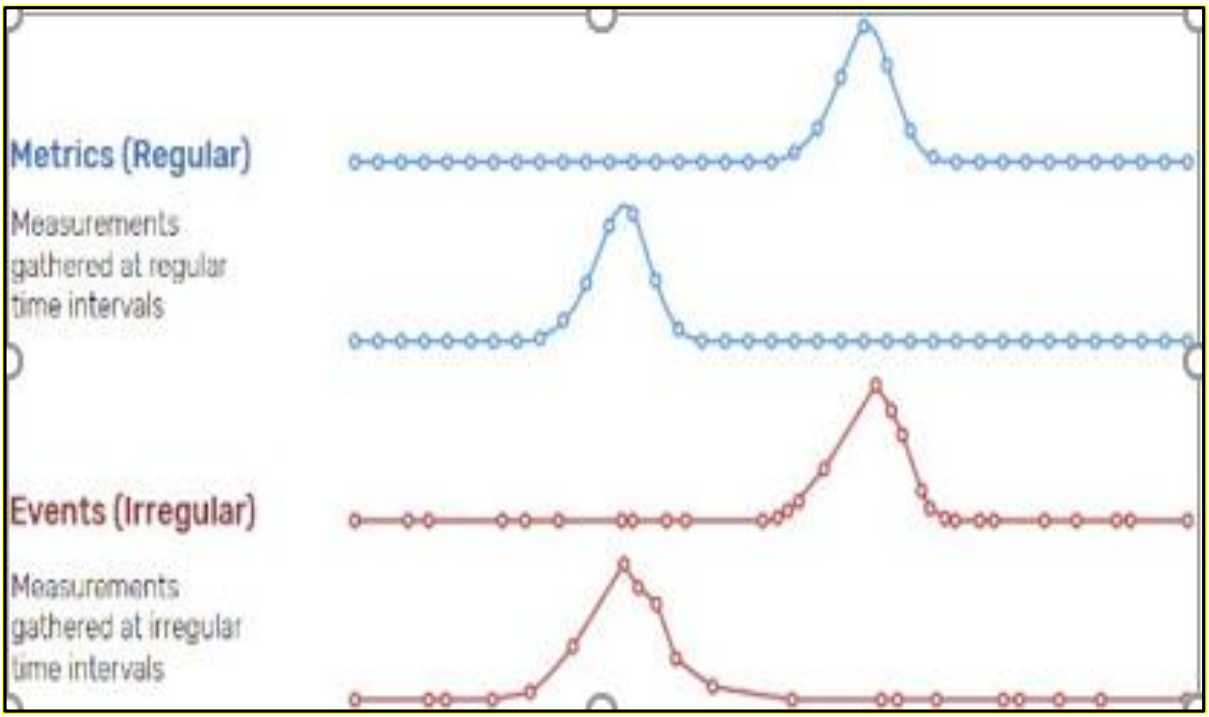

Рисунок 2.6 – Приклади типів часових рядів

Основними компонентами часових рядів є [13]:

– тенденція;
– сезонність;
– нерегулярність;
– циклічність.

Існує чимало методів які при роботі з часовими рядами спрямовані на точності та на допущенні найменшої кількості помилок. На даний момент є кілька нових методів машинного навчання, які є досить точними і маютьнепогану обчислювальну актуальність

Найпопулярнішими моделями для прогнозування часових рядів є:

– наївна модель;
– експоненціальна модель згладжування;
– ARIMA/SARIMA.

Для вирішення задач з прогнозування використовується модель Фейсбуку Prophet. Prophet – це пакет прогнозів на R і Python, розроблений дослідницькою групою Facebook із вивчення даних. Мета пакету — надати

бізнес-користувачам потужний і простий у використанні інструмент, який допоможе прогнозувати результати бізнесу без необхідності бути експертом з аналізу часових рядів. На веб-сайті Facebook Prophet, а також на інших сторонніх сайтах, таких як GitHub і Kaggle, є повна документація пакета, способи його використання та приклади. Після того, як дані очищено та налаштовано у належній схемі, фактичний пакет стає надзвичайно простим у використанні та може виконуватися у 4 рядках коду.

Основним алгоритмом є узагальнена адитивна модель, яка розкладається на три основні компоненти: тренд, сезонність і свята. Як я вже згадував вище, сезонність і тренд є двома важливими, але складними для кількісної оцінки компонентами аналізу часових рядів, і Facebook Prophet чудово враховує обидва.

Оскільки це розкладна модель, відносно легко отримати коефіцієнти моделі, щоб зрозуміти вплив сезонності, тенденції, свят та інших змінних регресора. Наприклад, якщо бізнес-команда намагається спрогнозувати продажі, вони можуть отримати коефіцієнт ціни, щоб побачити, наскільки ціна впливає на прогноз продажів. Декомпозиція допомагає командам зрозуміти рушії бізнесу. Це також допомагає визначити причини відхилення прогнозу. Наприклад, якщо є підвищення ціни в останню хвилину, яке не було враховано під час прогнозування, його можна визначити відносно швидко при оцінці моделі порівняно з фактичною ефективністю.

Одне застереження щодо Facebook Prophet полягає в тому, що він чудово підходить для стаціонарних даних. Стаціонарні дані — це дані часових рядів, які мають однакову поведінку та однакові статистичні властивості протягом усього часу.

На рисунку 2.7 наведено візуальні зображення того, як виглядають стаціонарні та нестаціонарні дані. Стаціонарні дані виділені синім кольором. Легко побачити закономірності протягом багатьох років, і все знаходиться в однаковому діапазоні. Помаранчеві дані – це нестаціонарні дані. На початку нестаціонарні дані можуть виглядати так, ніби вони слідують тенденції, але

в кінці періоду ви можете побачити, що дані досить швидко змінюють шаблони [14].

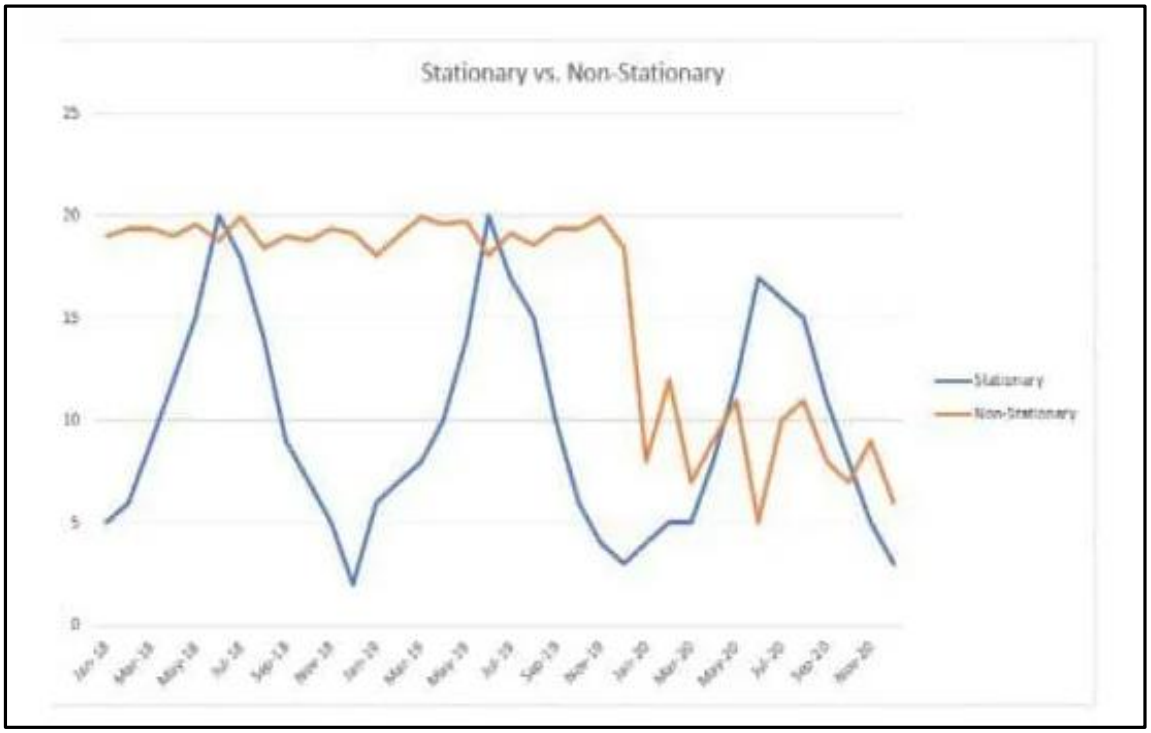

Рисунок 2.7 – Стаціонарні та нестаціонарні дані

Facebook Prophet погано працює з нестаціонарними даними, оскільки важко знайти фактичну сезонність і тренд даних, якщо закономірності суперечливі. Найкраще це працює з часовими рядами, що мають сильний сезонний ефект та кількома сезонами історичних даних. Facebook Prophet стійкий до відсутніх даних та змін у тренді, і, як правило, добре обробляє викиди. Головними перевагами даної моделі є:

– підтримка роботи з періодичними і неперіодичними рядами;

– можливість коректної роботи навіть за умови великої кількості відсутніх (або пропущених) даних;

– можливість виконувати прогнозування для вказаного інтервалу;

– можливість побудови зон невизначеності;

– використання як лінійної, так і логістичної моделі тренду, які можна обмежити точками зміни тренду;

– можливість додавати складові сезонності (як правило, рядами Фур’є заданого порядку);

– застосовування набору даних типу «holidays», що дає змогу вказувати значення і дати відомих подій, які тим чи іншим чином могли впливати на значення у відповідній точці.

На рисунку 2.8 зображено приклад прогнозу, який дозволяє будувати модель Facebook Prophet [14].

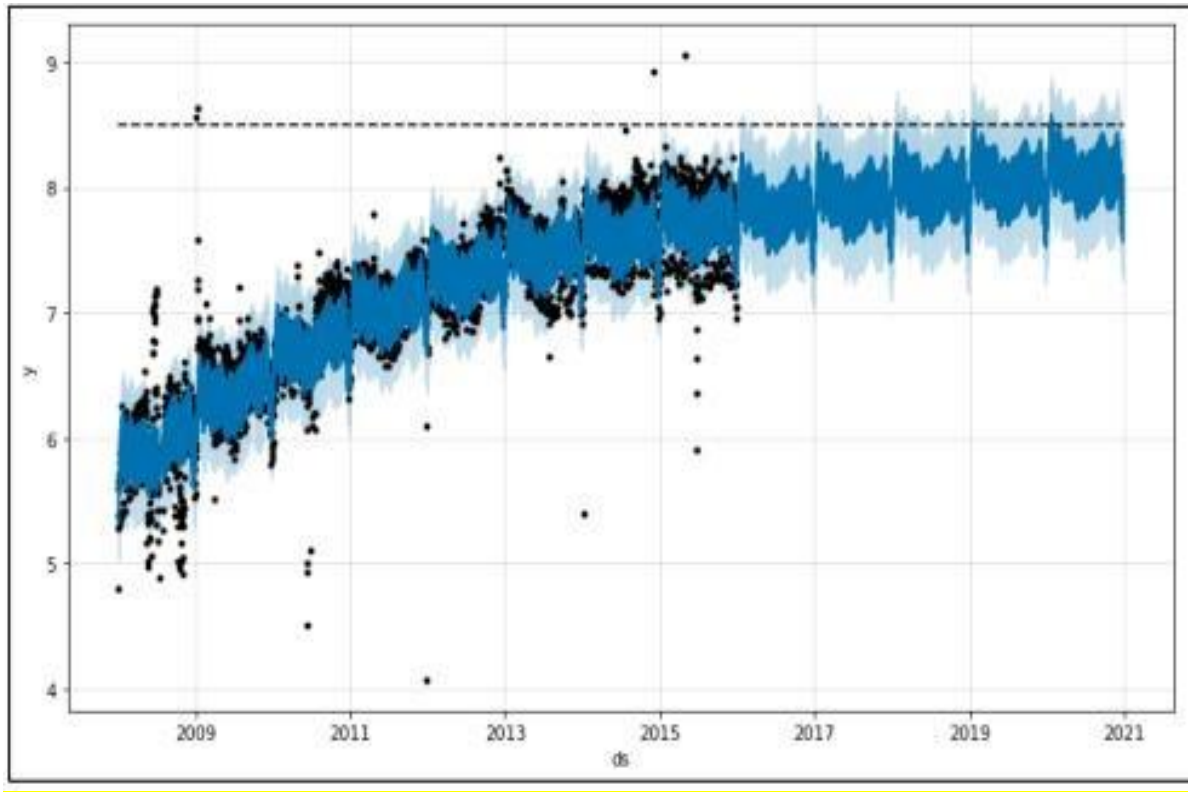

Рисунок 2.8 – Графік передбачення проведеного за допомогою Facebook Prophet

Окрім всього сказаного, Facebook Prophet дозволяє переглянути передбачення, розбите на окремі сезонні компоненти: тренд, річну та тижневу сезонність. Приклад даного графіку зображено на рисунку 2.9.

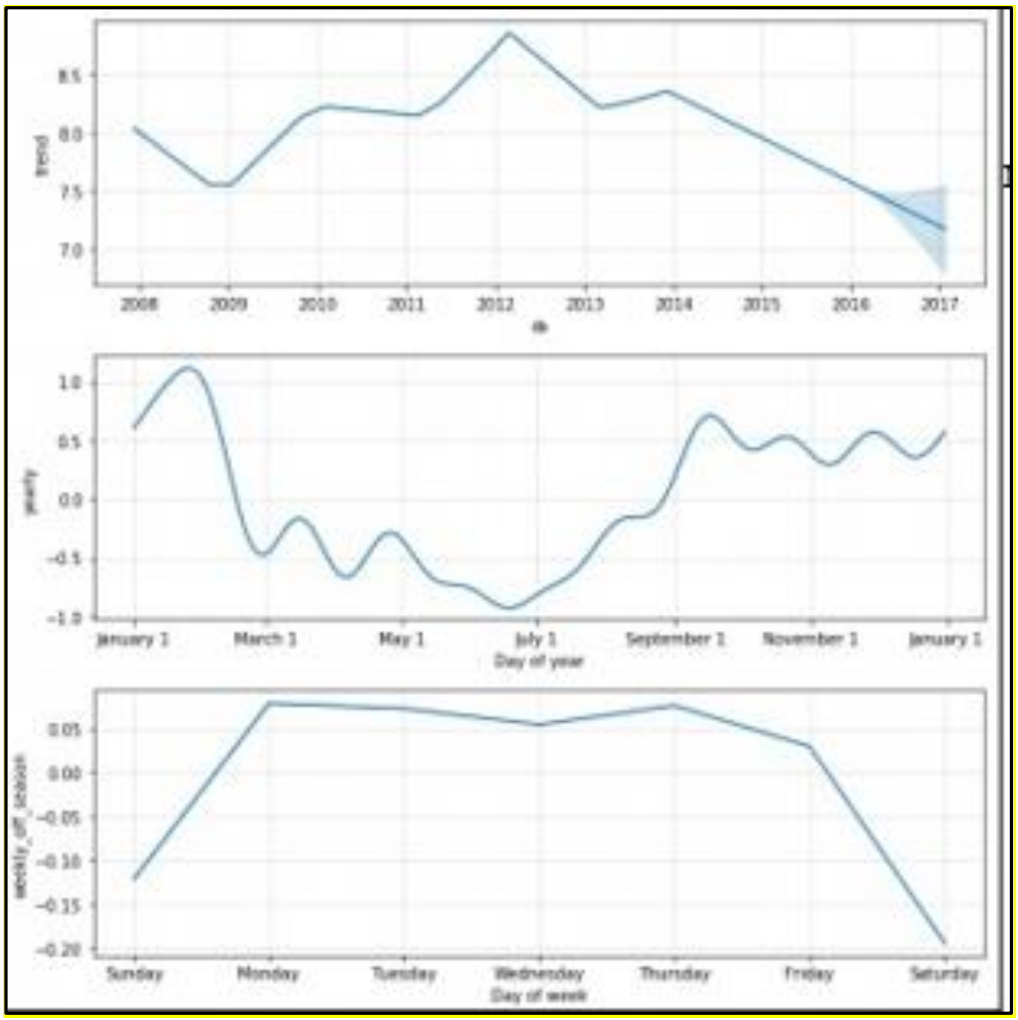

Рисунок 2.9 – Графік окремих сезонних компонентів передбачення

На графіку часових рядів спостережувані значення відображаються на осі ординат проти приросту часу на осі абсцис. Ці графіки візуально підкреслюють поведінку та шаблони даних і можуть закласти основу для створення надійної моделі.

Зокрема, візуалізація даних часових рядів надає попередній інструмент для визначення того, чи дані:

– злочинний або має вибухову поведінку;
– має часовий тренд;
– виявляє сезонність;
– демонструє структурні розриви.

Це, у свою чергу, може допомогти керувати методами тестування, діагностики та оцінки, які використовуються під час моделювання та аналізу

часових рядів.

Повернення даних середнього значення з часом повертається до незмінного середнього значення. Важливо знати, чи включає модель ненульове середнє, оскільки це є необхідною умовою для визначення відповідних методів тестування та моделювання.

Наприклад, тести на одиничний корінь використовують різні регресії, статистичні дані та розподіли, коли в модель включено ненульову константу.

Окрім ненульового середнього значення, дані часових рядів можуть також мати детермінований компонент, пропорційний періоду часу. Коли це відбувається, кажуть, що дані часових рядів мають часовий тренд.

Часові тенденції в даних часових рядів також мають значення для тестування та моделювання. Надійність моделі часових рядів залежить від правильного визначення та обліку часових тенденцій.

Діаграма часового ряду, яка виглядає так, ніби вона зосереджена навколо зростаючої чи спадної лінії, як на графіку вище, свідчить про наявність часової тенденції.

Сезонність — ще одна характеристика даних часових рядів, яку можна візуально визначити на графіках часових рядів. Сезонність виникає, коли дані часових рядів демонструють регулярні та передбачувані закономірності на часових інтервалах, менших за рік.

Прикладом часових рядів із сезонністю є роздрібні продажі, які часто збільшуються з вересня по грудень і зменшуються з січня по лютий. Приклад даного графіка зображено на рисунку 2.10 [14].

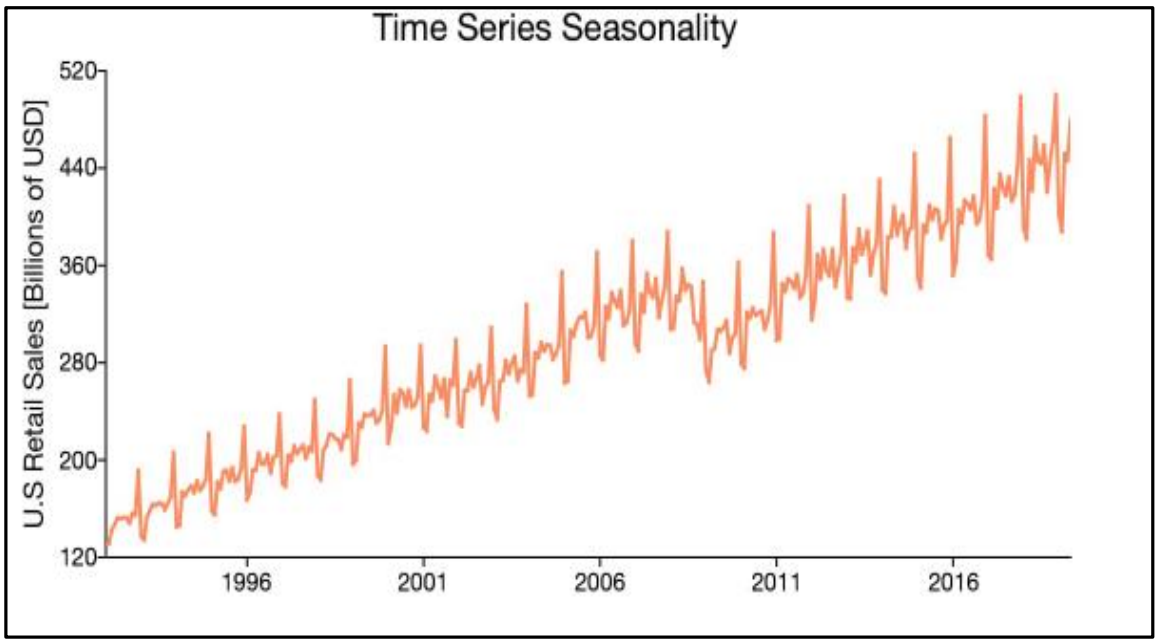

Рисунок 2.10 – Приклад часових рядів із сезонністю

Іноді дані часових рядів показують раптову зміну поведінки в певний момент часу. Наприклад, багато макроекономічних показників різко змінилися в 2008 році після початку світової фінансової кризи. Ці раптові зміни часто називають структурними розривами або нелінійністю. Приклад даних змін зображено на рисунку 2.11 [14].

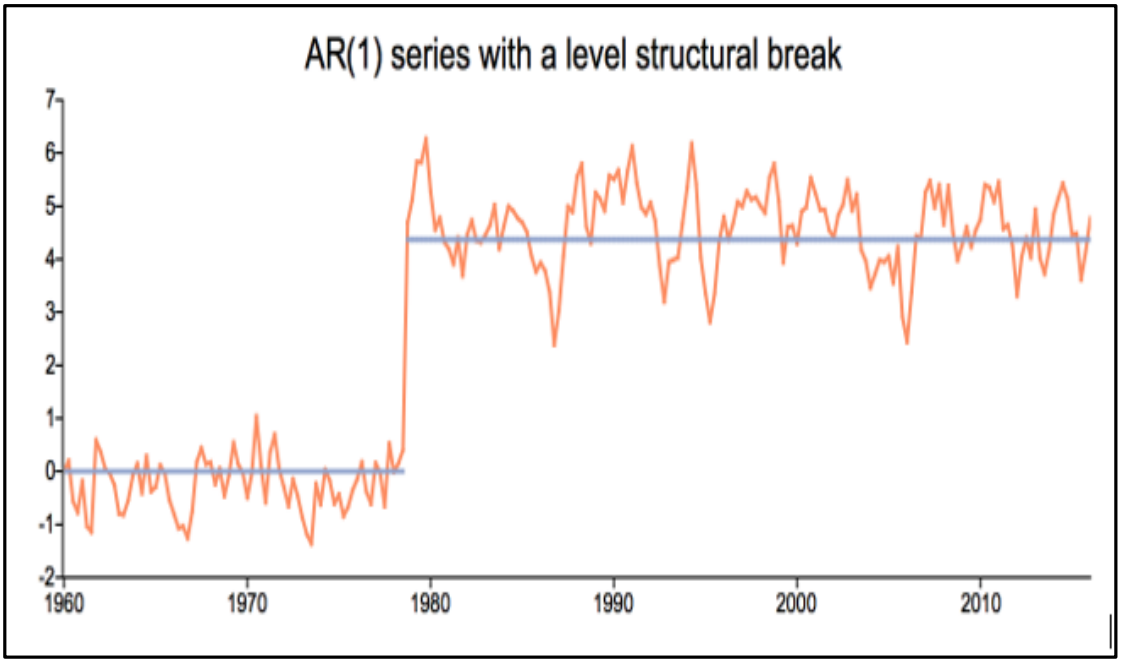

Рисунок 2.11 – Приклад структурного розриву

Іноді дані часових рядів показують раптову зміну поведінки в певний момент часу. Наприклад, багато макроекономічних показників різко змінилися в 2008 році після початку світової фінансової кризи. Ці раптові зміни часто називають структурними розривами або нелінійністю.

2.3 Вибір мови програмування

При виборі мови програмування для технології передбачення кількості нових хворих у Сполучених Штатах Америки було використано рейтинговий сервіс GitHub найпопулярніших мов програмування.

Лідером рейтинга виявився Python. Досить довгий час серед Data science розробників найбільш популярною мовою програмуванняявляється Python, який набув популярності через великий набір інструментів та методів реалізації машинного навчання і великої бази бібліотек [15].

Python — це широко поширена мова програмування загального призначення високого рівня. Він спочатку був розроблений Гвідо ван Россумом у 1991 році та розроблений Python Software Foundation. Його в основному було розроблено для акценту на читабельності коду, а його синтаксис дозволяє програмістам виражати концепції в меншій кількості рядків коду.

Наприкінці 1980-х історія мала бути написана. Це був час, коли почалася робота над Python. Невдовзі після цього Гвідо Ван Россум у грудні 1989 року почав працювати над додатками в Centrum Wiskunde & Informatica (CWI), який розташований у Нідерландах. Спочатку це було започатковано як проект для хобі, тому що він шукав цікавий проект, яким би зайняти себе під час Різдва. Кажуть, що мова програмування, на якій Python досяг успіху, — це мова програмування ABC, яка взаємодіяла з операційною системою Amoeba та мала функцію обробки винятків. Він уже брав участь у створенні ABC на початку своєї кар’єри, і він бачив деякі проблеми з ABC, але більшість функцій сподобалися. Після цього те, що він зробив, було дійсно дуже розумним. Він

узяв синтаксис ABC і деякі з його хороших рис. Він також викликав багато скарг, тому він повністю вирішив ці проблеми та створив хорошу мову сценаріїв, яка усунула всі недоліки. Натхнення для назви прийшло з телевізійного шоу BBC – «Летючий цирк Монті Пайтона», оскільки він був великим шанувальником телевізійного шоу, а також хотів коротку, унікальну та трохи загадкову назву для свого винаходу, тому він назвав його Python! Він був «доброзичливим диктатором на все життя» (BDFL), поки не залишив посаду лідера 12 липня 2018 року. Досить довго він працював у Google, але зараз він працює в Dropbox [16].

Мова нарешті була випущена в 1991 році. Коли вона була випущена, вона використовувала набагато менше кодів для вираження понять, якщо порівнювати її з Java, C++ і C. Її філософія дизайну також була досить хорошою. Його основна мета — забезпечити читабельність коду та покращити продуктивність розробника. Коли він був випущений, він мав більш ніж достатньо можливостей для забезпечення класів із успадкуванням, обробки винятків кількох основних типів даних і функцій.

На рисунку наведено ілюстрації різних версій Python разом із часовою шкалою (рис. 2.12) [15].

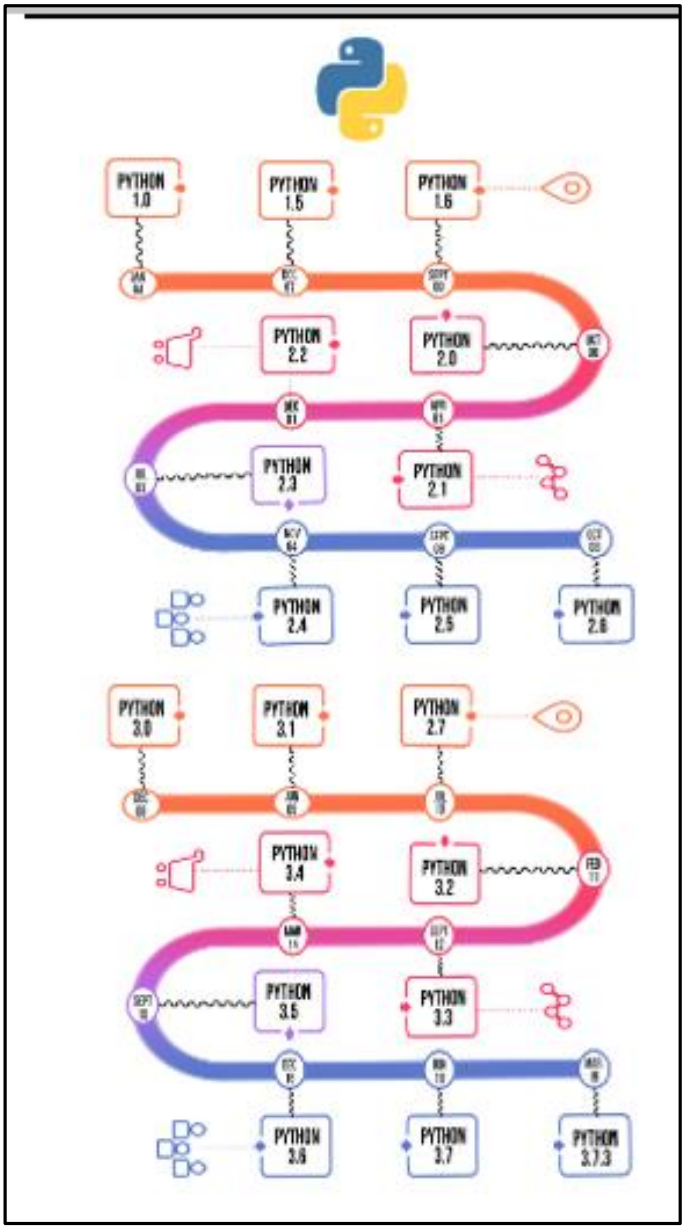

Рисунок 2.12 – Ілюстрації різних версій Python разом із часовою шкалою

У світі програмування Python відомий як одна з найпопулярніших і швидкозростаючих мов програмування. Її можна визначити як високорівневу, інтерпретовану, об’єктно-орієнтовану мову сценаріїв і мову загального призначення. У сучасному IT-ринку існують різні мови програмування, але тільки Python набув досить широкого поширення, чому так? Це факт і питання, яке піднімає багато людей у всьому світі.

Python — це мова комп’ютерного програмування високого рівня з динамічною семантикою. Це високо інтерпретована та об’єктно-орієнтована

мова з вбудованими структурами даних у поєднанні з динамічною типізацією та динамічним зв’язуванням, що робить її дуже привабливою для швидкої розробки додатків.

Пайтон досяг певних висот у аналітиці великих даних та машинному аналізі. тому розглянимо його переваги щоб бути впевненим у виборі мови програмування (рис. 2.13).

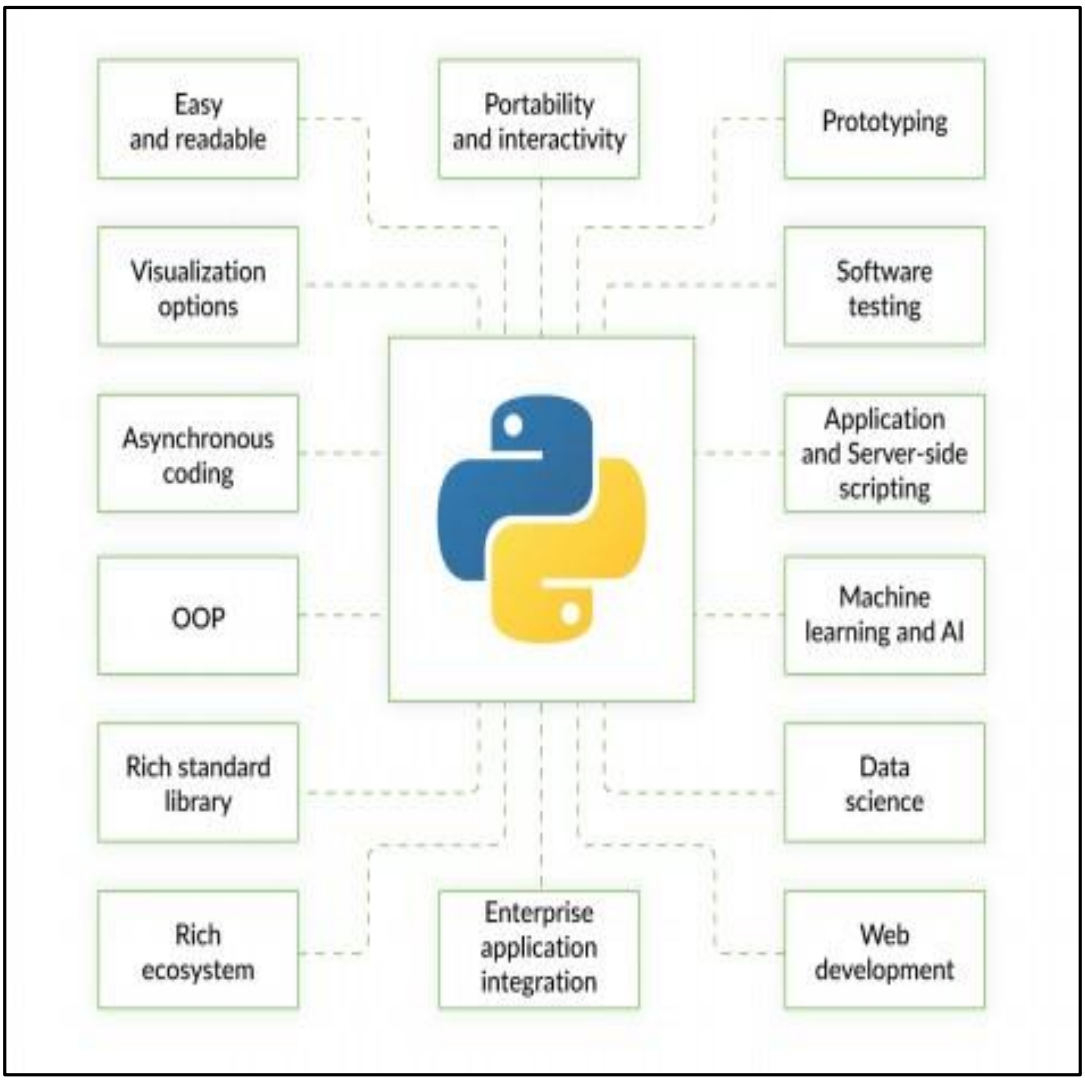

Рисунок 2.13 – Переваги використання мови Python [15]

Python — це мова, яка використовується великою спільнотою програмістів завдяки численним перевагам. Багато компаній обирають Python як основну мову програмування. Давайте дізнаємось про переваги Python і причини, чому більшість розробників люблять Python?

– Це просто;
– Це безкоштовно;
– Це легко використовувати;
– Це дуже сумісно;
– Він об'єктно-орієнтований;
– Він має багато бібліотек;
– Він має вбудовані структури даних;
– Це широко застосовне;
– Це підвищує швидкість і продуктивність;
– Це легко навчитися.

Багато основних бібліотек та служб на основі API мають прив'язки або обгортки Python, дозволяючи Python вільно взаємодіяти з цими службами або безпосередньо використовувати ці бібліотеки [17].

Також Python став найбільш затребуваною мовою програмування серед вакансій для Data Science спеціалістів обігнавши R та SQL (рис. 2.14) [17].

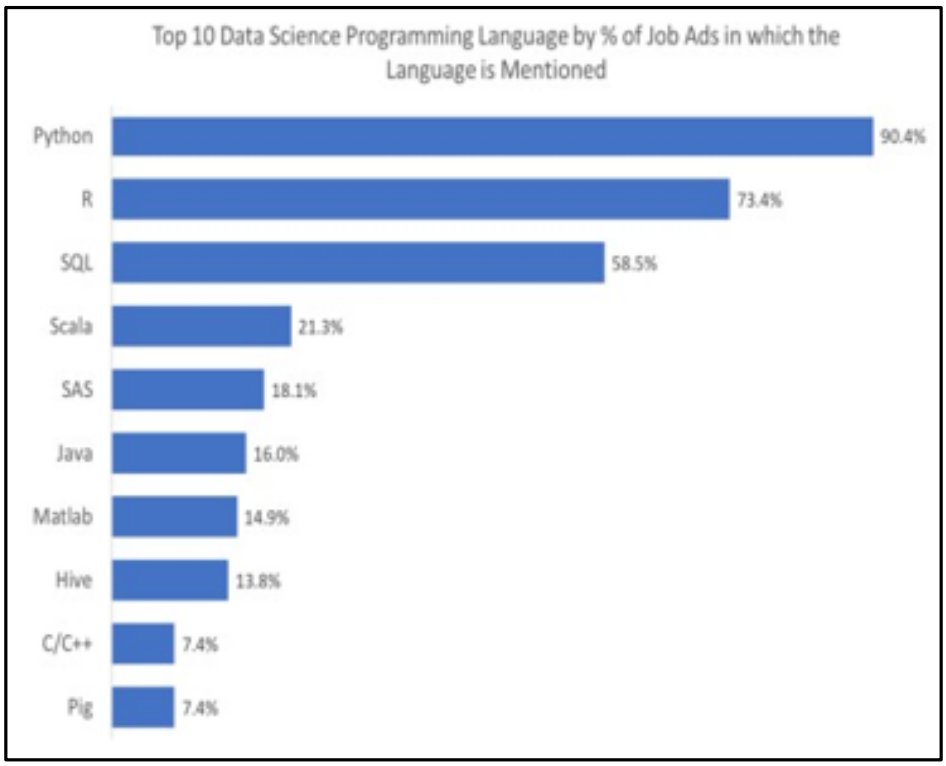

Рисунок 2.14 – Рейтинг мов програмування вакансій Data Science серед роботодавців

Python — найкраща мова комп'ютерного програмування, яка використовується для створення веб-сайтів і програмного забезпечення, автоматизації завдань і аналізу даних. Python використовується для створення різноманітних програм і не спеціалізується на конкретних проблемах, оскільки це мова загального призначення. Його універсальні функції разом із зручністю для початківців роблять Python однією з найбільш використовуваних мов програмування сьогодні.

Python використовується у всьому, від машинного навчання до створення веб-сайтів і тестування програмного забезпечення. Python використовується в багатьох галузях. Звичайно, мова Python найчастіше використовується для створення мобільних, веб- та настільних програм. Під час тестування цих програм також використовується Python. Він використовується для створення алгоритмів рекомендацій, розробки програмного забезпечення, яке керує безпілотними автомобілями, автоматизує наші повсякденні завдання, від веб-розробки до сценаріїв, тестування програмного забезпечення та створення прототипів. Нижче ми перерахували різні варіанти використання Python [18].

Мета мов програмування полягає в тому, щоб допомогти вам створити добре закодований комп'ютерний код для згладження ваших бізнес-операцій. Python може бути мовою, яка створить гарний комп'ютерний код. У Python є всі параметри, які дозволять створити гарну комп'ютерну мову – справжню мову, яку цінують і приймають у всьому світі. Тому немає сенсу гадати, чому python кращий за інші мови.

Ми знаємо переваги вивчення Python на сучасному світовому ринку праці. Як надійна мова програмування та забезпечує просте використання рядків коду, Python має всі функції, які роблять його загальноприйнятним і цінним. Його технічне обслуговування та налагодження також можна легко виконати. Це також одна з головних причин, чому пошуковий гігант Google зробив Python однією зі своїх офіційних мов програмування.

2.4 Огляд основних бібліотек для машинного навчання

Python використовують не тільки через те, що легка і проста у вивченні. Але є й інша вагома причина, за замовчуванням в Python вже додано безліч корисних бібліотек. Але, на мій погляд, популярним і потужним Python роблять 230 тисяч пакетів, створених користувачами.

За версією сервіса Github найпопулярніші бібліотеки в 2018 році зображені на рисунку 2.15 [19].

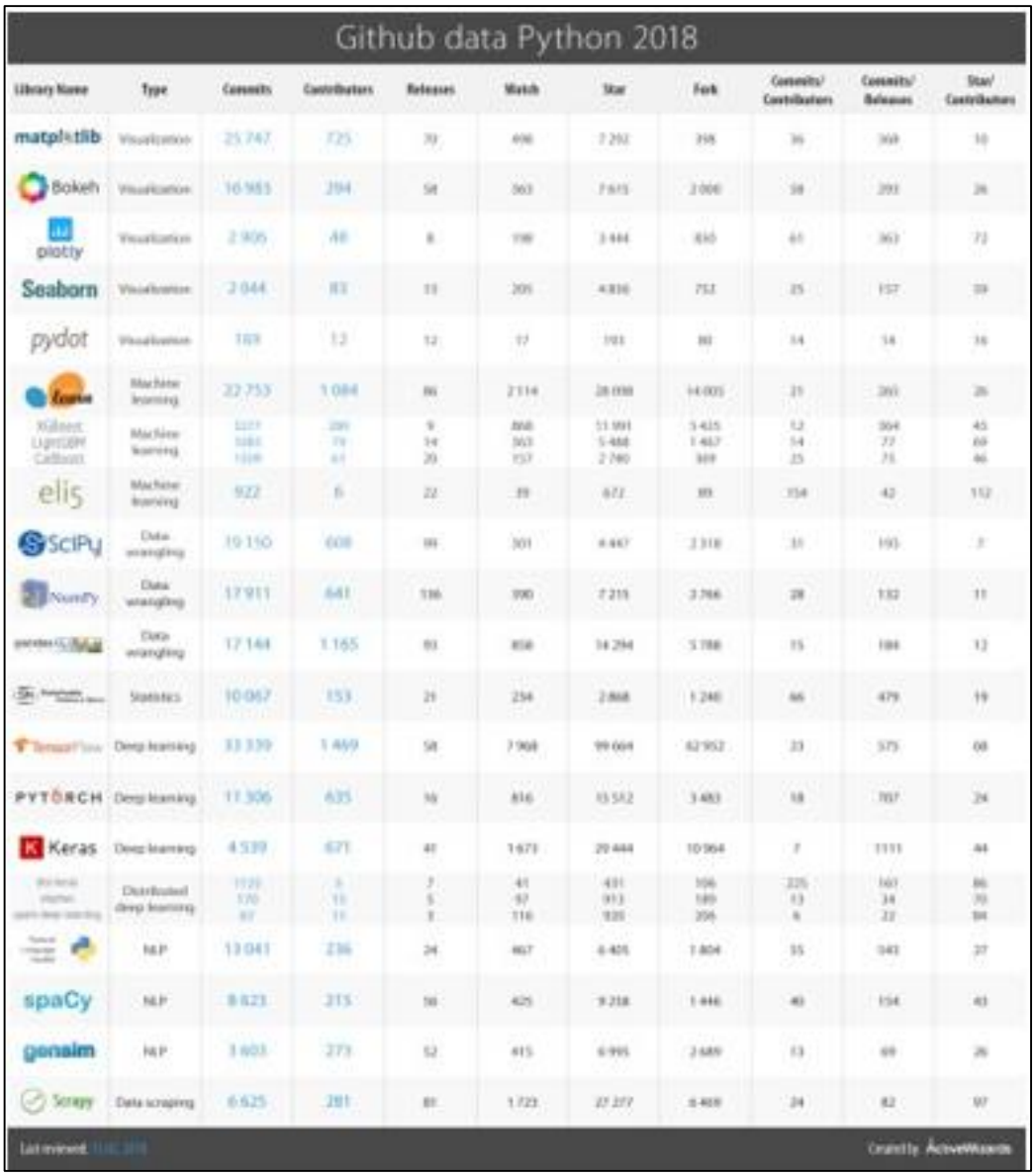

Рисунок 2.15 – Найпопулярніші бібліотеки Python для машинного

Стандартна бібліотека Python надає модулі для загальних завдань програмування - математики, обробки рядків, доступу до файлів і каталогів, роботи в мережі, асинхронні операції, створення потоків, багатопроцесорне управління тощо.

Розподіл Python за замовчуванням також забезпечує елементарну, але корисну, міжплатформенну бібліотеку графічного інтерфейсу через Tkinter та вбудовану копію бази даних SQLite 3.

Тисячі сторонніх бібліотек, доступних за допомогою індексу пакетів Python (PyPI).

Наприклад:

– бібліотека BeautifulSoup надає універсальний набір інструментів для вишкрібання HTML;

– requests роблять роботу з HTTP-запитами в масштабі безболісною тапростою;

– фреймворки, Flask та Django, дозволяють швидко розвивати веб-сервіси, що охоплюють як прості, так і розширені випадки використання;

– кілька хмарних служб можна керувати за допомогою об’єктної моделі Python за допомогою Apache Libcloud;

– NumPy, Pandas та Matplotlib прискорюють математичні та статистичні операції та полегшують створення візуалізації даних.

2.5 Висновки

В другому розділі проведено огляд та дослідження вхідних даних, на основі яких здійснюється прогнозування на платформі Kaggle. Проведено короткий огляд методів ML та аналіз технології передбачення Facebook Prophet, а також обгрунтовано вибір мови програмування для реалізації даної програми та огляд її бібліотек.

3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ КІЛЬКОСТІ НОВИХ ХВОРИХ НА КОРОНАВІРУС У СПОЛУЧЕНИХ ШТАТАХ АМЕРИКИ

3.1 **Розроблення алгоритму інформаційної технології**

Для розв’язання даного завдання створено алгоритм інформаційної технології з опрацювання вхідних даних та налаштування і застосування моделі, яка містить такі кроки (рис. 3.1):

Крок 1. Збір та опрацювання відкритих даних.

Крок 2. Перший крок створення моделі та оптимізація її параметрів, включно – перебір усіх можливих варіантів значень ширини адаптивного вікна в діапазоні [-3, …, 3] (усі цілі числа і нуль), сили впливу (масштабу) свят і псевдосвят та варіантів режиму урахування свят (мультиплікативний чи адитивний).

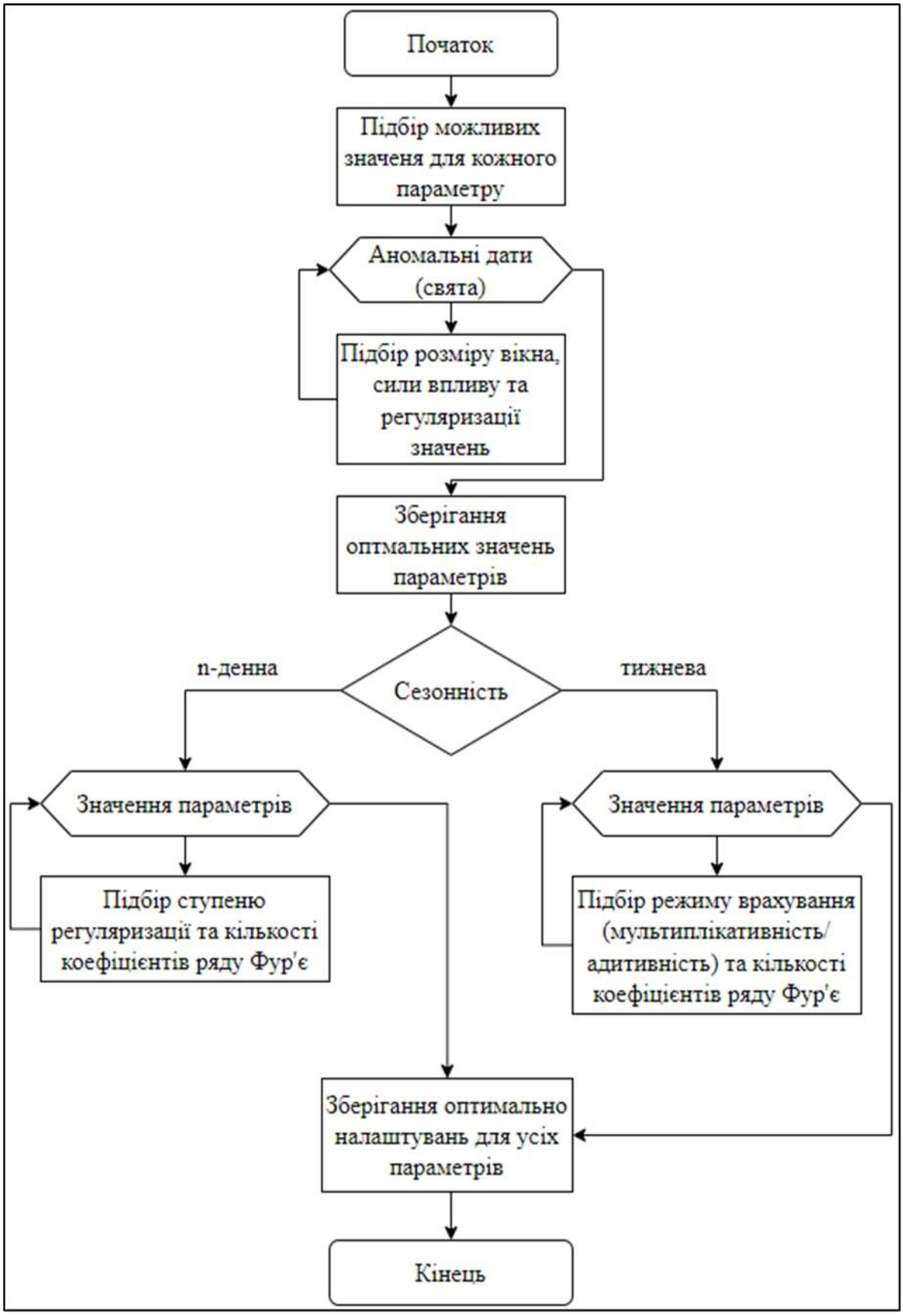

Рисунок 3.1 – Блок-схема алгоритму роботи інформаційної технології

Крок 3. Наступний крок побудови моделі та оптимізація її параметрів із ними робити перебір значень таких параметрів: варіанти режиму урахування, ступінь регуляризації та кількість коефіцієнтів ряду Фур'є для опису тижневої та, окремо, 4-денного видів сезонності.

Крок 4. Проаналізувати закономірності по структурі ідентифікованої оптимальної моделі.

Крок 5. Формування прогнозів на задану кількість N днів вперед.

Як виняток, може бути використана спрощена модель на основі Facebook Prophet, яка може відрізнятись від основної моделі такими спрощеннями:

- урахування тільки державних свят з 7-денним зсувом та адаптивним вікном;

- урахування тільки тижневої сезонності.

Для даної спрощеної моделі досить одного етапу ідентифікації, на якому варто оптимізувати тільки такі параметри: розмір вікна, сила впливу і ступінь регуляризації значень в аномальні дати. Інші – потрібно задавати сталими, на основі минулих розрахунків [18].

3.2 Прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки

Для того, щоб почати роботу необхідно підключити до нашого коду бібліотеки, які будуть в подальшому використовуватися. Код для підключення бібліотек наведено на рисунку 3.2.

```
import os
import pandas as pd
import numpy as np
import requests
import seaborn as sns
from matplotlib import pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
```

Рисунок 3.2 – Підключення бібліотек

Ми будемо використовувати бібліотеку Facebook Prophet яка досить добре себе показує з даними які мають сезонні зміни. І також підключаємо аномальні дати (свята). Дані підключення вказані на рисунку 3.3.

```
from datetime import date, timedelta, datetime
from fbprophet import Prophet
from fbprophet.make_holidays import make_holidays_df
from fbprophet.diagnostics import cross_validation, performance_metrics
from fbprophet.plot import plot_cross_validation_metric
import holidays
from collections import Counter
import pycountry
```

Рисунок 3.3 – Підключення бібліотеки Facebook Prophet та аномальних дат

Для початку скачаємо потрібні нам дані з Github. Даний код зображено на рисунку 3.4.

```
# Thanks https://github.com/CSSEGISandData/COVID-19
myfile = requests.get('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_
open('data', 'wb').write(myfile.content)
confirmed_global_df = pd.read_csv('data')
confirmed_global_df
```

Рисунок 3.4 – Завантаження даних з Центру Джона Хопкінса через API

Результатом даного коду являється таблиця з статистичними даними де dates – дата, confirmed - кількість нових хворих, country – країна. Дана таблиця зображена на рисунку 3.5.

	Date	Country	Confirmed
0	2020-01-22	US	1
1	2020-01-23	US	1
2	2020-01-24	US	2
3	2020-01-25	US	2
4	2020-01-26	US	5
...	...	...	...
981	2022-09-29	US	96347994
982	2022-09-30	US	96385048
983	2022-10-01	US	96392543
984	2022-10-02	US	96397885
985	2022-10-03	US	96438726

Рисунок 3.5 – Статистичні дані

Наступним кроком будуть формуватися аномальних дат (рис. 3.6).

```
def cut_df(date0: str,
           df:pd.DataFrame(),
           col: str):
    # Deletes all rows of dataframe df with df[col] < date0
    format0 = '%Y-%m-%d'
    df_temp = df.copy()
    df_temp['col_dt'] = pd.to_datetime(df_temp[col], format=format0, er
rors='coerce')
    date0_dt = datetime.strptime(date0, format0)
    df_temp = df_temp[df_temp['col_dt'] >= date0_dt]

    return df_temp
```

Рисунок 3.6 – Формування аномальних дат

Далі задаються різні види аномальних даних (рис. 3.7).

```
# Thanks to dataset https://www.kaggle.com/vbmokin/covid19-holidays-of-countries
holidays_df = pd.read_csv('../input/covid19-holidays-of-countries/holidays_df_of_70_countrie
holidays_df[holidays_df['country'] == country_main]
```

Рисунок 3.7 – Задання аномальних даних

Наступним кроком є скачування даних Google (рис. 3.8).

```
# Thank to https://github.com/GoogleCloudPlatform/covid-19-open-data
data = pd.read_csv(f"https://storage.googleapis.com/covid19-open-data/v2/{country_iso_alpha}.
```

Рисунок 3.8 – Скачування датасету Google

На рисунку 3.9 зображено код критерія оксфордської лабораторії stringence index, а на рисунку 3.10 зображено його графік разом з датами послаблення.

```
# Thanks to https://www.kaggle.com/vbmokin/covid-19-in-ukraine-prophet-holidays-tuning
data['stringency_index_jump'] = 0
for i in range(len(data)-1):
    if (data.loc[i+1,'stringency_index'] is not None) and (data.loc[i,'stringency_index'] is
        data.loc[i+1, 'stringency_index_jump'] = 1
source_gov = 'https://www.bsg.ox.ac.uk/research/research-projects/oxford-covid-19-government
dates_gov_list = data[data['stringency_index_jump'] == 1]['date'].tolist()
holidays_df = aux_holidays_df_generator(holidays_df, dates_gov_list, 'the weakening of quara
plot_with_anomalies(data, ["stringency_index"], {"stringency_index" : "Stringency index and
```

Рисунок 3.9 – Код критерія Оксфордської лабораторії Stringence index

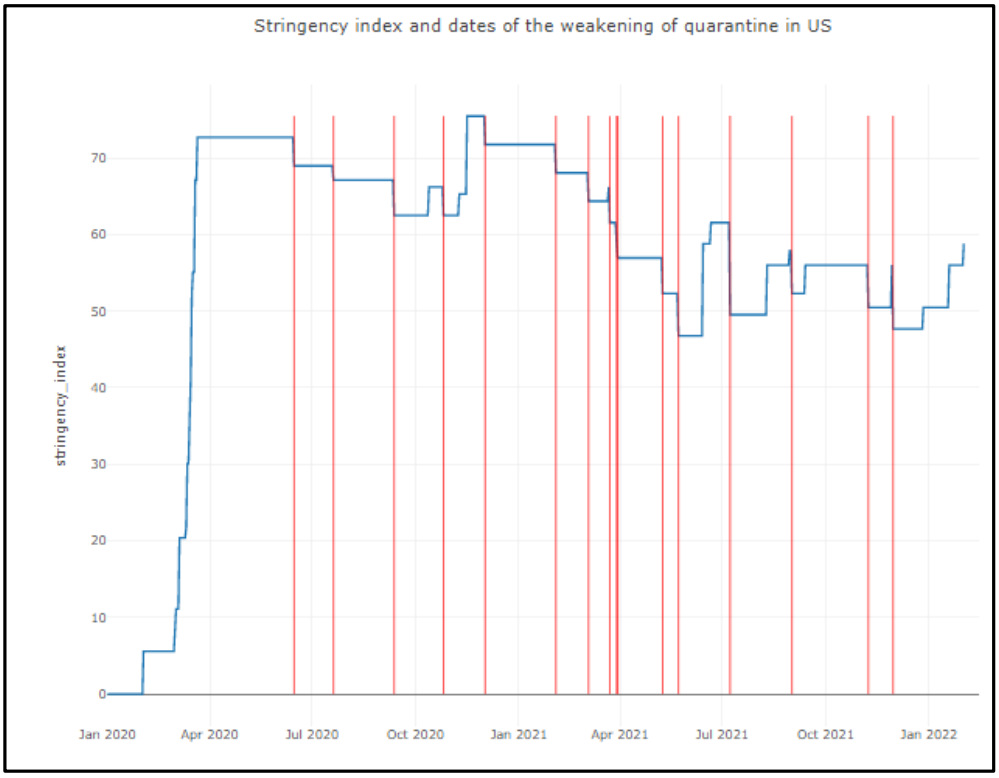

Рисунок 3.10 – Графік критерія Оксфордської лабораторії Stringence index

Далі беруться дуже комфортабельні умови, відслідковуються опади і середня температура (рис. 3.11).

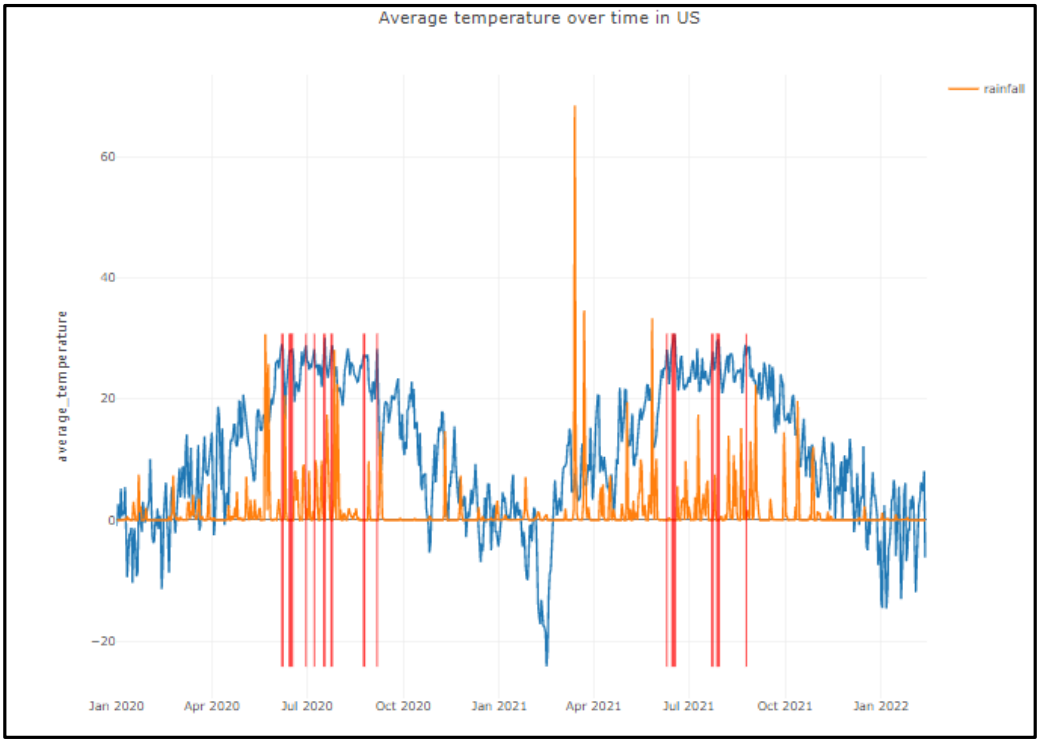

Рисунок 3.11 – Графік залежностей опадів і середньої температури

Далі ми додаємо дані по роботі лабораторій по святах, дані по локдауну, дані вихідного дня (рис. 3.12) і в результаті отримуємо графік по кількості нових хворих у США (рис. 3.13).

```
# Removing the holidays for the period when there were still diseases < 10
holidays_df['ds_dt'] = pd.to_datetime(holidays_df['ds'], format='%Y-%m-%d', errors='ignore')
date_the_first_many_cases = datetime.strptime(df2[df2.Confirmed >= 10].bfill(axis=1)['Date'].tolist()[0], '%Y-%m-%d')
holidays_df = holidays_df[holidays_df['ds_dt'] >= date_the_first_many_cases]
holidays_df
```

	ds holidays	holiday	ds	country	code	country official name	lower window	upper window	prior scale	source	ds dt
1	2020-05-25	Memorial Day	2020-06-01	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2020-06-01
34	2020-06-15	the weakening of quarantine	2020-06-22	United States	US	United States of America	-3	3	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-06-22
3	2020-07-03	Independence Day (Observed)	2020-07-10	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2020-07-10
2	2020-07-04	Independence Day	2020-07-11	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2020-07-11
22	2020-07-20	the weakening of quarantine	2020-07-27	United States	US	United States of America	-3	3	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-07-27
4	2020-09-07	Labor Day	2020-09-14	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2020-09-14
24	2020-09-12	the weakening of quarantine	2020-09-19	United States	US	United States of America	-3	3	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-09-19
5	2020-10-12	Columbus Day	2020-10-19	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2020-10-19
25	2020-10-26	the weakening of quarantine	2020-11-02	United States	US	United States of America	-3	3	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-11-02
6	2020-11-11	Veterans Day	2020-11-18	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2020-11-18
7	2020-11-26	Thanksgiving	2020-12-03	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2020-12-03
23	2020-12-02	the weakening of quarantine	2020-12-09	United States	US	United States of America	-3	3	10	https://www.bsg.ox.ac.uk/research/research-pro...	2020-12-09
8	2020-12-25	Christmas Day	2021-01-01	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2021-01-01
9	2021-01-01	New Year's Day	2021-01-08	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2021-01-08
11	2021-01-18	Martin Luther King Jr. Day	2021-01-25	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2021-01-25
30	2021-02-03	the weakening of quarantine	2021-02-10	United States	US	United States of America	-3	3	10	https://www.bsg.ox.ac.uk/research/research-pro...	2021-02-10
12	2021-02-15	Washington's Birthday	2021-02-22	United States	US	United States of America	-3	3	10	https://github.com/dr-prodigy/python-holidays	2021-02-22

Рисунок 3.12 - Дані по роботі лабораторій по святах, дані по локдауну, дані вихідного дня

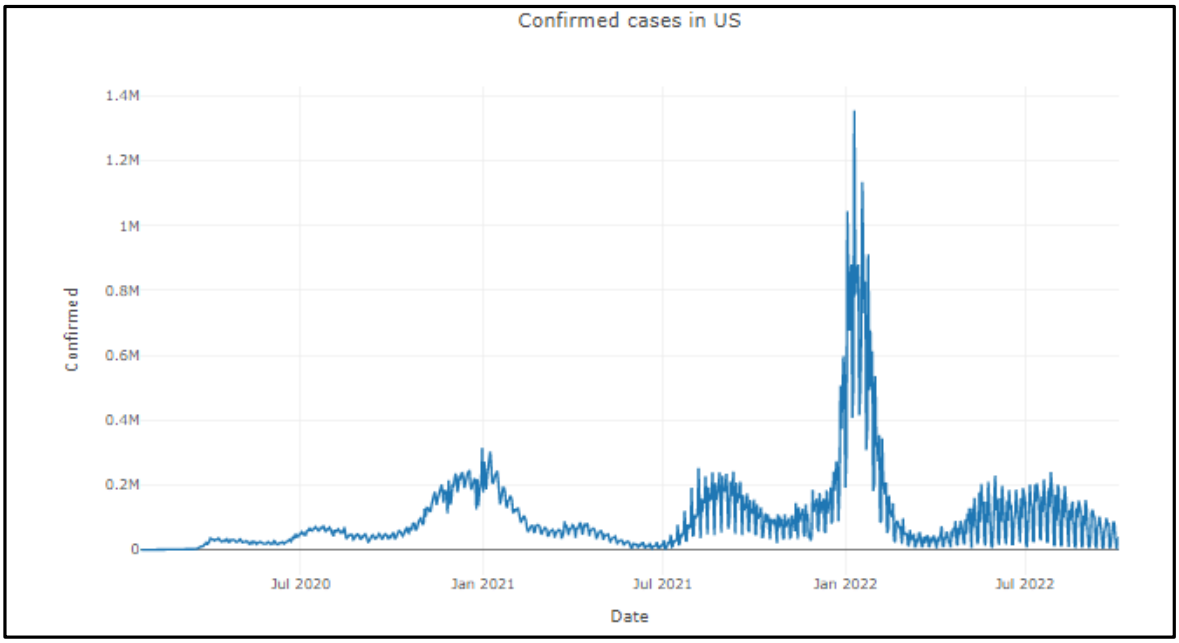

Рисунок 3.13 – Кількість нових хворих у США

Поєднавши аномальні дати з кількістю нових хворих отримаємо даний графік (рис. 3.14).

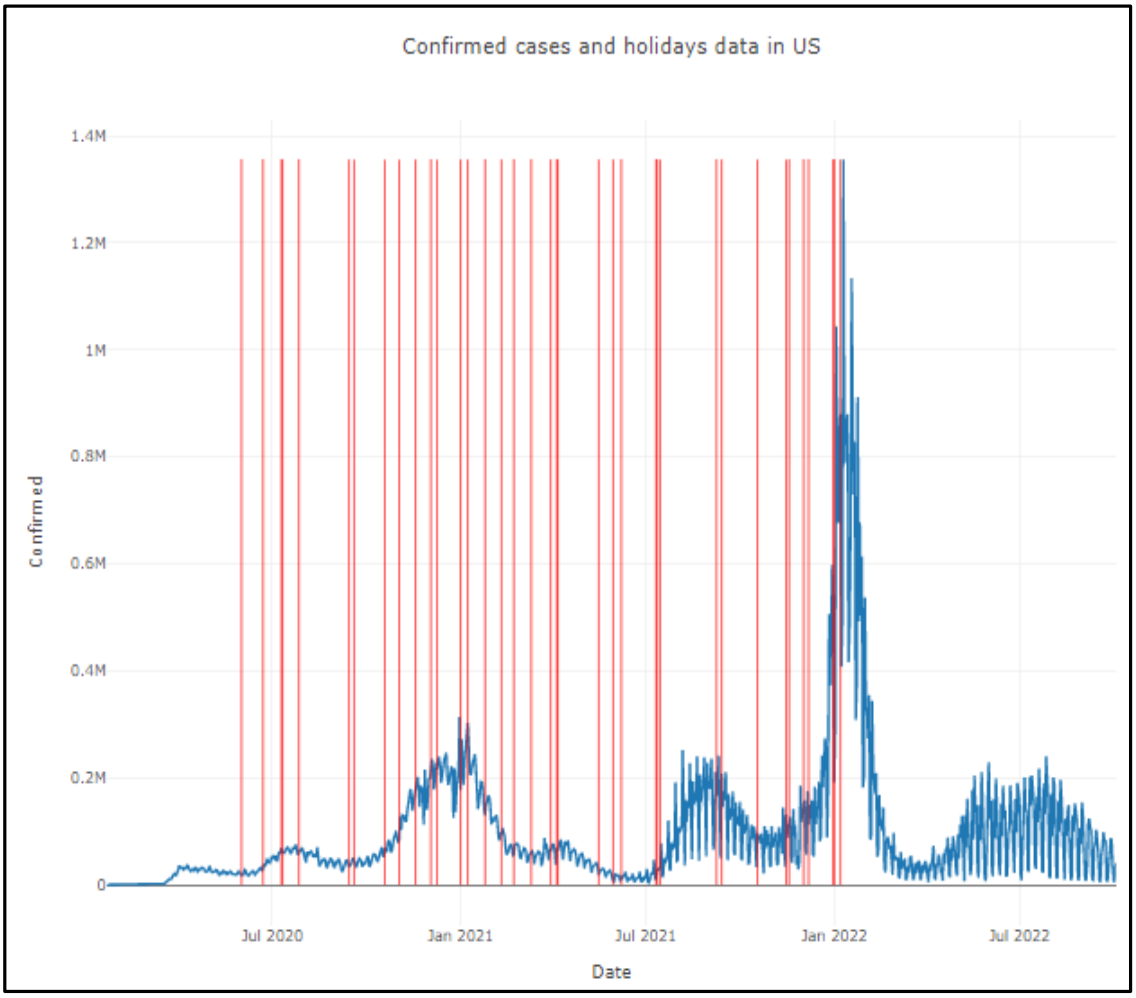

Рисунок 3.14 – Кількість нових хворих у США з аномальними датами

Наступним кроком задаємо на скільки днів робиться прогноз і скільки даних вносяться у валідаційний датасет (рис. 3.15).

```
        print('i =',i+n*k,' from',2*n-1,':  lower_window =', lower_window_i, 'upper_window =',upper_window_i, 'prior_scale =', prior_scale_i)
        print('relative_error_holidays =',relative_error_holidays, 'relative_error_holidays_min =',relative_error_holidays_min, '\n')

    # Results visualization
    print('Seasonality mode is', mode_main_list[k])
    print('The best errors of model with holidays is', relative_error_holidays_min, 'with lower_window =', str(lower_window_best),
          ' upper_window =', str(upper_window_best), ' prior_scale =', str(prior_scale_best))
    print('The error of model without holidays is', relative_error, '\n')

# Save results to dataframe with all dates
forecast_holidays_df_best['country'] = country_main
forecast_holidays_df_best.rename(columns={'yhat':'confirmed'}, inplace=True)
forecast_holidays_dfs = forecast_holidays_df_best.tail(days_to_forecast_for_evalution)

# Forecasting the future
if relative_error < relative_error_holidays_min:
    # The forecast without taking into account the holidays is the best
    model_future_best, forecast_future_best = model_training_forecasting(country_df_future, days_to_forecast, mode_main=mode_best)
    forecast_plot = model_future_best.plot(forecast_future_best, ylabel='Confirmed in '+ country_main + ' (forecasting without holidays) - ' + mode_main_list[k])
    cols = cols_w
    print('The best model is model without holidays')
else:
    # The forecast taking into account the holidays is the best
    print('The best model is model with holidays')
    model_future_best, forecast_future_best = model_training_forecasting(country_df_future, days_to_forecast, holidays_df,
                                                                          mode_main=mode_best)
    forecast_plot = model_future_best.plot(forecast_future_best, ylabel='Confirmed in '+ country_main + ' (forecasting with holidays) - ' + mode_best)
    cols = cols_h
# Save forecasting results
forecast_future_df_best = forecast_future_best[cols]
forecast_future_df_best['country'] = country_main
forecast_future_df_best.rename(columns={'yhat':'confirmed'}, inplace=True)
forecast_future_dfs = forecast_future_df_best.tail(days_to_forecast)
fig = model_future_best.plot_components(forecast_future_best)
return forecast_holidays_dfs, relative_errors_holidays, forecast_future_dfs, results
```

Рисунок 3.15 – Код задання періоду прогнозування

Наступним кроком перебираються всі комбінації, яких є 127 і в кінці можна побачити, що похибка, де включаються свята є досить малою (рис. 3.16). Результат моделювання даних можна побачити на рисунку 3.17.

```
i = 116  from 127 :  lower_window = -4 upper_window = 2 prior_scale = 0.05
relative_error_holidays = 40.87318061360494 relative_error_holidays_min = 36.443913112691924

i = 117  from 127 :  lower_window = -4 upper_window = 2 prior_scale = 1
relative_error_holidays = 40.46802281119584 relative_error_holidays_min = 36.443913112691924

i = 118  from 127 :  lower_window = -4 upper_window = 2 prior_scale = 5
relative_error_holidays = 40.018541119771264 relative_error_holidays_min = 36.443913112691924

i = 119  from 127 :  lower_window = -4 upper_window = 2 prior_scale = 20
relative_error_holidays = 39.856540426820324 relative_error_holidays_min = 36.443913112691924

i = 120  from 127 :  lower_window = -4 upper_window = 3 prior_scale = 0.05
relative_error_holidays = 41.00334302007997 relative_error_holidays_min = 36.443913112691924

i = 121  from 127 :  lower_window = -4 upper_window = 3 prior_scale = 1
relative_error_holidays = 38.44441877959478 relative_error_holidays_min = 36.443913112691924

i = 122  from 127 :  lower_window = -4 upper_window = 3 prior_scale = 5
relative_error_holidays = 37.800473828616376 relative_error_holidays_min = 36.443913112691924

i = 123  from 127 :  lower_window = -4 upper_window = 3 prior_scale = 20
relative_error_holidays = 38.126035914835704 relative_error_holidays_min = 36.443913112691924

i = 124  from 127 :  lower_window = -4 upper_window = 4 prior_scale = 0.05
relative_error_holidays = 41.2967546797599 relative_error_holidays_min = 36.443913112691924

i = 125  from 127 :  lower_window = -4 upper_window = 4 prior_scale = 1
relative_error_holidays = 39.38021468993567 relative_error_holidays_min = 36.443913112691924

i = 126  from 127 :  lower_window = -4 upper_window = 4 prior_scale = 5
relative_error_holidays = 39.182630029746946 relative_error_holidays_min = 36.443913112691924

i = 127  from 127 :  lower_window = -4 upper_window = 4 prior_scale = 20
relative_error_holidays = 38.75780740333803 relative_error_holidays_min = 36.443913112691924

Seasonality mode is multiplicative
The best errors of model with holidays is 36.443913112691924 with lower_window = -1  upper_window = 3  prior_scale = 20
The error of model without holidays is 40.77454435342325

The best model is model with holidays
CPU times: user 18min 7s, sys: 20.5 s, total: 18min 27s
Wall time: 12min 42s
```

Рисунок 3.16 – Перебір всіх комбінацій за чотирма показниками

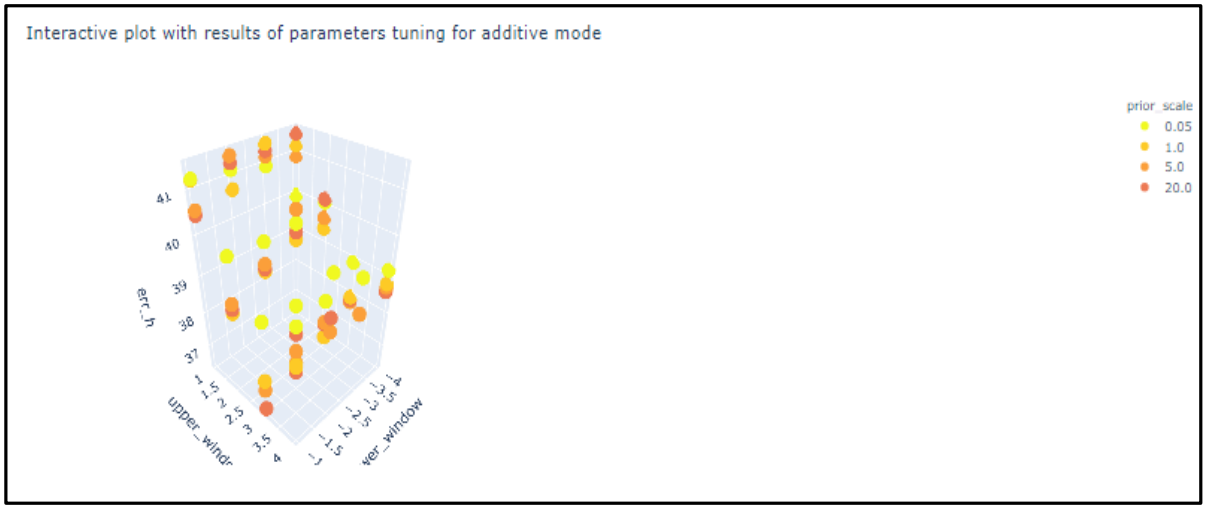

Рисунок 3.17 – Результат моделювання

Далі йде другий етап, в якому перебираються сезони. На рисунку 3.18 показано елементи сезонності де weekly - щотижнева, several_days - період хвиль, 4 days між тижнева, аналіз показав що оптимальною буде чотири дні.

```
# Thanks to https://www.kaggle.com/vbmokin/covid-19-in-ukraine-prophet-holidays-tuning
def model_training_forecasting(df, forecast_days, holidays_df=None, mode_main='multiplicative',
                               weekly_fourier_order=10, triply_fourier_order=10,
                               changepoint_prior_scale = changepoint_prior_scale_initial_level, mode_seasonality
    # Optimal Prophet model training and forecasting

    model = Prophet(daily_seasonality=False, weekly_seasonality=False, yearly_seasonality=False,
                    holidays=holidays_df, changepoint_range=1, changepoint_prior_scale = changepoint_prior_scale,
                    seasonality_mode = mode_main)
    if weekly_fourier_order > 0:
        model.add_seasonality(name='weekly', period=7, fourier_order=weekly_fourier_order, mode = mode_seasonali
                              prior_scale = changepoint_prior_scale*add_season_reg_coef)
    if triply_fourier_order > 0:
        model.add_seasonality(name='triply', period=3, fourier_order=triply_fourier_order, mode = mode_seasonali
                              prior_scale = changepoint_prior_scale/add_season_reg_coef)
    model.fit(df)
    future = model.make_future_dataframe(periods=forecast_days)
    forecast = model.predict(future)

    forecast[forecast['yhat'] < 0]['yhat'] = 0
    forecast['yhat_lower'] = forecast['yhat_lower'].round().astype('int')
    forecast['yhat'] = forecast['yhat'].round().astype('int')
    forecast['yhat_upper'] = forecast['yhat_upper'].round().astype('int')

    return model, forecast
```

Рисунок 3.18 – Елементи сезонності

Далі проводиться розрахунок комбінацій і моделюються графіки (рис. 3.19 - 3.23).

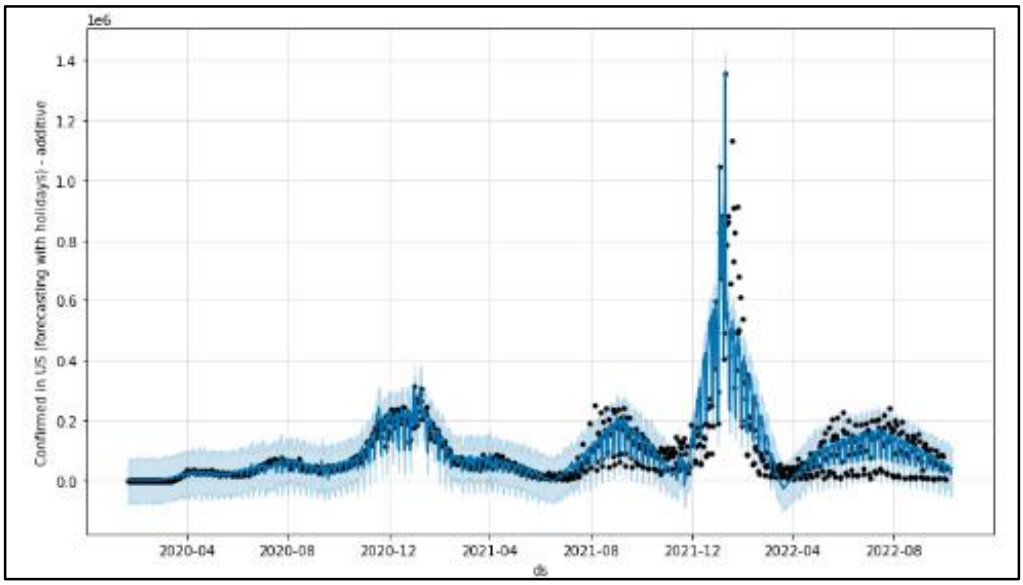

Рисунок 3.19 – Загальний результат моделювання

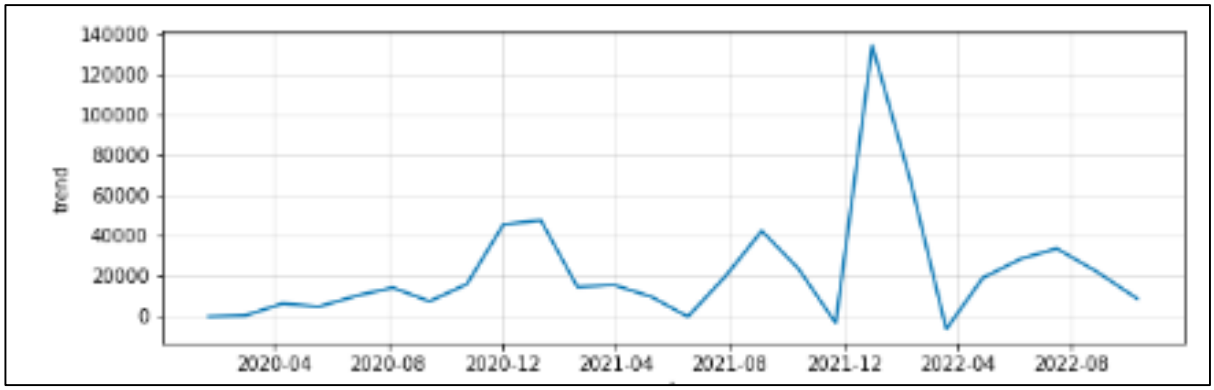

Рисунок 3.20 – Кусково - лінійна апроксимація

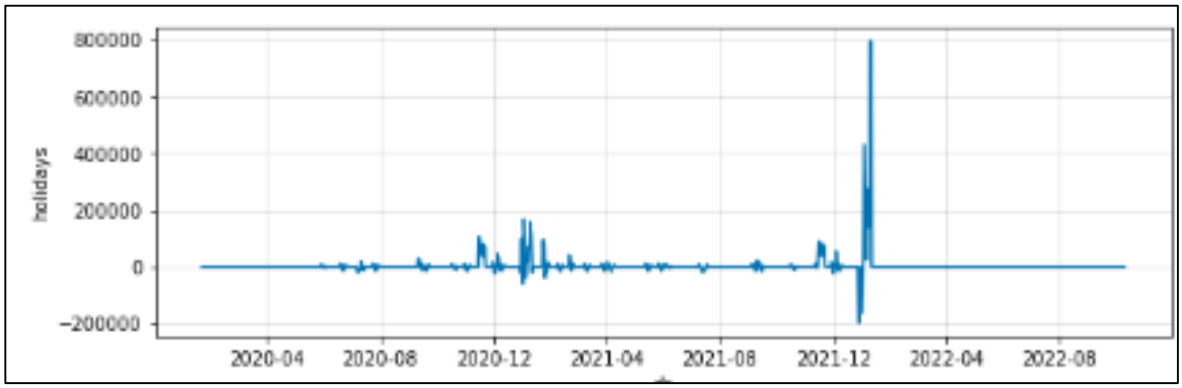

Рисунок 3.21 – Вплив свят

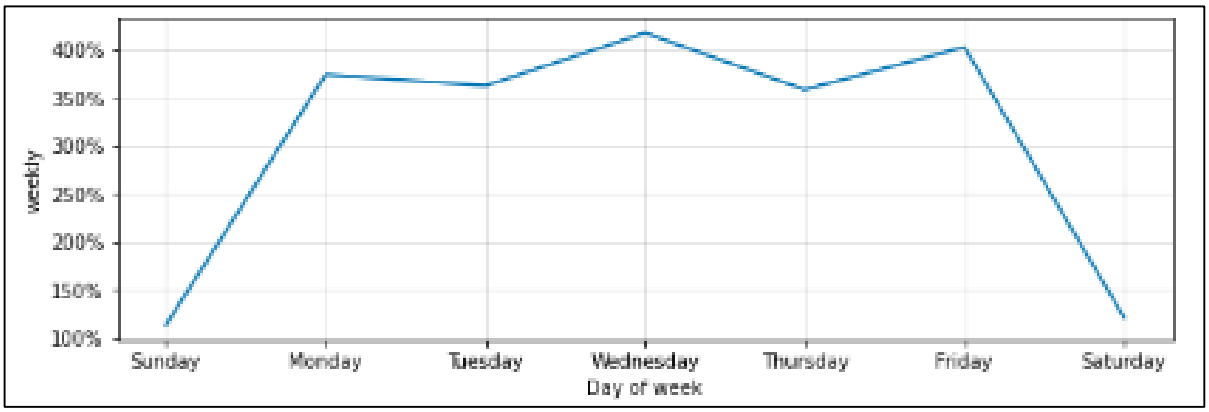

Рисунок 3.22 – Тижневий розподіл

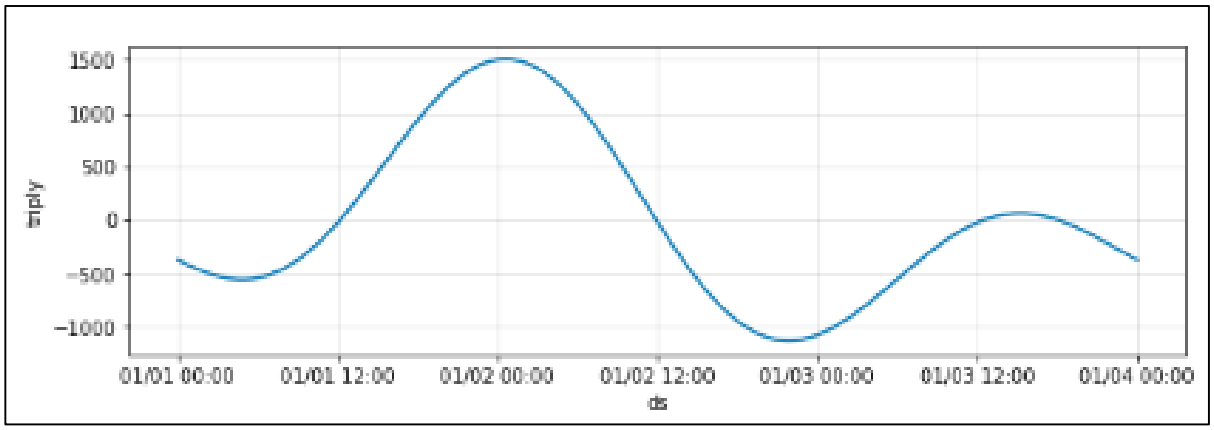

Рисунок 3.23 – Міжтижневий розподіл

Далі ми бачимо прогноз на тиждень вперед (рис. 3.24).

	ds	yhat_lower	yhat	yhat_upper
875	2022-06-15	5768102	5810525	5853830
876	2022-06-16	5772515	5813495	5857439
877	2022-06-17	5776248	5817230	5857642
878	2022-06-18	5775035	5820561	5860846
879	2022-06-19	5775144	5819397	5864299
880	2022-06-20	5774424	5817905	5859430
881	2022-06-21	5774899	5819839	5860690

Рисунок 3.24 – Прогноз на тиждень

Отже, найкраще модель показала себе при таких показниках (рис. 3.25).

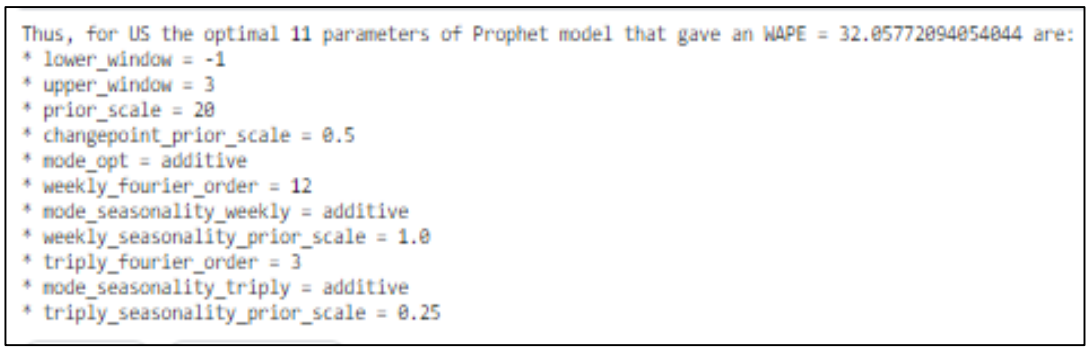

Рисунок 3.25 – Показники найкращої моделі

3.3 Висновки

У даному розділі реалізовано інформаційну технологію прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки та отримані результати прогнозування на різний період часу.

Дослідження показали, що найкращою моделлю є модель Facebook Prophet з тижневою мультиплікативною сезонністю, інтерпольованою рядом Фур'є 10 порядку, та 3-денною мультиплікативною сезонністю, інтерпольованою рядом Фур'є 3 порядку. Відносна похибка цієї моделі у разі прогнозування на тиждень вперед з урахуванням "свят" (дат аномалій) з з регуляризацією 0,5 становить 36,44%. Натомість, без урахування цих аномалій оптимальна модель досягає точності лише у 40,77%. Отже, за рахунок врахування та регуляризації аномалій вдалось підвищити точність на 10,6%.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нової інформаційної технології прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки. Метою дослідження є підвищення точності прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки. Подібні технології вже існують, як для США так і для інших країн. Особливістю розроблюваної технології є те, що вона дозволяє підвищити точність прогнозування за рахунок використання методів машинного навчання.

За аналоги була взята технологія прогнозування коронавірусу в Україні, яка створена на основі відкритих даних. Протее даний аналог показує коректні результати лише при короткостроковому прогнозуванні. Орієнтовна вартість такої системи, для комерційного сегменту, складає близько 45000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, у відповідності із таблиці 4.1 [21].

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність в реальних умовах
Ринкові переваги					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною .	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження таблиці 4.1

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х
12	Необхідна розробка регламентних документів та отримання великої кількості до-	Необхідно отримання великої кількості дозвільних документів на виробництво	Процедура отримання дозвільних документів для виробництва та реалізації	Необхідно тільки повідомлення відповідним органам	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Хмарук С. М.	Пасічнюк Д. В.	Радецький О. В.
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	4	4	3
Ринок збуту	3	3	2

Продовження таблиці 4.2

Критерії оцінювання	ПІБ експертів		
	Хмарук С. М.	Пасічнюк Д. В.	Радецький О. В.
	Бали		
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	4	3	3
Сума	43	42	41
Середньоарифметична сума балів	(43+42+41) / 3 = 42		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що інформаційна технологія прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки надає підвищену точність прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки. Подібні технології вже існують, як для США так і для інших країн, протее через те, що модель будувалась тільки на ділянці приросту вона перестала бути ефективною. Розроблювана технологія дозволяє підвищити точність прогнозування за рахунок використання методів машинного навчання

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$З_о = \frac{М}{Т_р} \cdot t, \quad (4.1)$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн.;

$Т_р$ – число робочих днів за місяць, 20 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	34000	1700,00	28	47600,000
Програміст	31500	1575,00	28	44100,000
Всього				91700,00

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 13 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 13\ \% / 100\ \% \quad (4.2)$$

$$З_д = (91700{,}00 \cdot 13\ \% / 100\ \%) = 11921{,}00\ \text{(грн)}.$$

Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22\ \% / 100\% \qquad (4.3)$$

$$Н_з = (91700{,}00 + 11921{,}00) \cdot 22\ \% / 100\ \% = 22796{,}62\ \text{(грн)}.$$

Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючи, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$А = \frac{Ц}{Тв} \cdot \frac{t_{вик}}{12}\ \text{[грн]}. \qquad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

Т – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{вик}$ – термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп’ютер балансова вартість якого становить 32000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,40 міс.

$$А_{обл} = \frac{32000}{2} \times \frac{1{,}4}{12} = 1866{,}67\ \text{(грн)}. \qquad (4.5)$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5.

Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних активів є безкоштовною, то $В_{нем.ак.}$ = 0 грн.

Таблиця 4.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортиза-ційні відрахуван-ня, грн.
Комп'ютер та комп'ютерна периферія (Acer Aspire 7 A715-42G-R3EZ)	32000	2	1,40	1866,667
Офісне обладнання (меблі)	22000	4	1,40	641,667
Приміщення	980000	20	1,40	5716,667
Всього				8225,00

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \quad (4.5)$$

де В – вартість 1 кВт-години електроенергії для 1 класу підприємства, В = 6,2 грн./кВт;

П – встановлена потужність обладнання, кВт. П = 0,35 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

$К_п$ – коефіцієнт використання потужності, $К_п$ = 0,9.

$$В_е = 0,9 \cdot 0,35 \cdot 8 \cdot 28 \cdot 6,2 = 437,472 \text{ (грн)}.$$

Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.6)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$$І_в = 91700,00 * 57\% / 100\% = 52269 \text{ (грн)}.$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов’язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов’язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов’язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників:

$$Н_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.7)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$Н_{нзв} = 91700,00 * 135 \% / 100 \% = 123795 \text{ (грн)}.$$

Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$В_{заг} = 91700,00+11921,00+22796,62+8225,00+437,47+52269+$$
$$+123795 = 311144,09 \text{ (грн)}.$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} \text{ [грн]}, \quad (4.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η=0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9. Оберемо η = 0,5, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 311144,09 / 0,5 = 622288 \text{ (грн)}.$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб’єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов’язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

– абсолютного економічного ефекту (чистого дисконтованого доходу);

– внутрішньої економічної дохідності (внутрішньої норми дохідності);

– терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об’єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.9)$$

де $\pm\Delta Ц_о$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_о$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_о = Ц_б \pm \Delta Ц_о$;

Цб – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт λ = 0,8333.

р – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році ϑ =18%.

Припустимо, що при прогнозованій ціні 12000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на *1500* грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 1000 шт., протягом другого року – на 800 шт., протягом третього року на 500 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$\Delta П_1$ = (0*1500 + (12000 + 1500)*1000)* 0,8333* 0,32) * (1 - 0,18) = 2623999,895 (грн0.

$\Delta П_2$ = (0*1500 + (12000 + 1500)*(1000+800)* 0,8333* 0,32) * (1 - 0,18) = 5313599,787 (грн).

$\Delta П_3$ = (0*1500 + (12000 + 1500)*(1000+800+500)* 0,8333* 0,32) * (1 - 0,18) = 6789599,728 (грн).

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 14727199,41 грн.

Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0{,}05…0{,}15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$ПП = (2623999{,}895/(1+0{,}1)^1)+(5313599{,}787/(1+0{,}1)^2)+(6789599{,}728/(1+0{,}1)^3) = 2385454{,}45 + 4391404{,}783 + 5101126{,}768 = 11877986 \text{ (грн).}$$

Далі розраховують величину початкових інвестицій *PV*, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} * ЗВ, \quad (4.11)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}=2...5$, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 622288 = 1244576{,}37 \text{ (грн).}$$

Тоді абсолютний економічний ефект $Е_{абс}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$Е_{абс} = ПП – PV, \quad (4.12)$$

$$Е_{абс} = 11877986 – 1244576,37 = 10633409,63 \text{ (грн)}.$$

Оскільки $Е_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої но-рми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.13)$$

$Т_ж$ –життєвий цикл наукової розробки, роки.

$$Е_в = \sqrt[3]{(1 + 10633409,63/1244576,37} - 1 = 1,121$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \tag{4.14}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d = (0,09…0,14);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{\min} = 0,14 + 0,05 = 0,19$$

Так як $E_в > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в}, \tag{4.15}$$

$$T_{ок} = 1 / 1,121 = 0,89 \text{ р.}$$

Оскільки $T_{ок}$< 3-х років, а саме термін окупності рівний 0,89 роки, то фінансування даної наукової розробки є доцільним.

4.4 Висновки

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 622288 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В

результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,89 років.

ВИСНОВКИ

В процесі виконання магістерської роботи розроблено інформаційну технологію прогнозування кількості нових хворих на коронавірус у США. Проаналізувавши дані для вирішення поставленої задачі обрано модель Facebook Prophet, передусім, через її можливості роботи із зашумленими даними з певною періодичністю.

Розглянуто ситуацію з коронавірусом у світі. Проаналізовано статистичні дані по кількості нових хворих, і по інших показниках, а також розглянуто деякі з найбільш поширених методів прогнозування інфекційних захворювань, їх переваги і недоліки.

Проведено огляд та дослідження вхідних даних, на основі якого здійснюється прогнозування на платформі Kaggle. Проведено короткий огляд методів ML та аналіз технології передбачення Facebook Prophet, а також обгрунтовано вибір оптимальних технологій, мови програмування для реалізації даної програми та огляд її бібліотек.

Реалізовано інформаційну технологію, яка дозволила підвищити точність прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки та отримані результати прогнозування на різний період часу.

Дослідження показали, що найкращою моделлю є модель Facebook Prophet з тижневою мультиплікативною сезонністю, інтерпольованою рядом Фур'є 10 порядку, та 3-денною мультиплікативною сезонністю, інтерпольованою рядом Фур'є 3 порядку. Відносна похибка цієї моделі у разі прогнозування на тиждень вперед з урахуванням "свят" (дат аномалій) з з регуляризацією 0,5 становить 36,44%. Натомість, без урахування цих аномалій оптимальна модель досягає точності лише у 40,77%. Отже, за рахунок врахування та регуляризації аномалій вдалось підвищити точність на 10,6%.

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 622288 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат.

Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,89 років.

Отже, завдання, поставлене у магістерській кваліфікаційній роботі, виконано в повному обсязі.

За результатами даної роботи написані тези доповіді, які були апробовані на Всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» (МН-2023) та опубліковані у матеріалах цієї конференції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Козачко О. М. Лопухов Б. Л. Інформаційна технологія прогнозування кількості нових хворих на коронавірус в Сполучених Штатах Америки. *Молодь в науці: дослідження, проблеми, перспективи (МН-2022)*. Вінниця, 2023. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/viewFile/16830/14043

2. Андрейчин М. Н. Коронавірусна інфекція : Енциклопедія сучасної України : НАН України, НТШ, 2014. 767 с.

3. Голубовська М. А. Інфекційні хвороби : підручник. Київ : ВСВ , 2018. 633 с.

4. Перебіг коронавірусу в Сполучених Штатах Америки. URL: https://uk.wikipedia.org/wiki/Коронавірусна_хвороба_2019_у_Сполучених_Штатах_Америки

5. Національна академія наук України. URL: https://www.nas.gov.ua/UA/Messages/Pages/View.aspx?MessageID=7885

6. Мокін В. Б., Лосенко А. В., Ящолт А. Р. Інформаційна технологяі аналізу та прогнозування кількості нових випадків захворювань на коронавірус SARS-CoV-2 в Україні на основі моделі Prophet, *Вісник ВПІ*. 2020. № 5. С. 71–83

7. Мокін В. Б, Лосенко А.В, Ящолт А.Р. «Інформаційна технологія аналізу та прогнозування багатохвильової кількості нових випадків захворювань на коронавірус COVID-19 на основі моделі Prophet» : *Вісник ВПІ*, 2020. 65 с.

8. Річерд В., Кельйо П. Створення систем машинного навчання за допомогою Python. Мехіко : Packt Publishing, 2013. 312 с.

9. Машинне навчання. URL: https://www.it.ua/knowledge-base/technology-innovation/machine-learning

10. Machine learning for time series forecasting. URL: https://codeit.us/blog/machine-learning-time-seriesforecasting

11. Бідюк П. І., Романенко В. Д., Тимощук О. Л. Аналіз часових рядів. Київ : НБУВ, 2010. 109 с.

12. Facebook Prophet. URL:https://facebook.github.io/prophet/

13. Machine learning. URL: https://www.expert.ai/blog/machine-learning-definition/

14. Emptyq. URL:https://ua.emptyq.net/a?ID=9698dcc2-c9af-42f0-9253-67cb80372053

15. Путівник мовою програмування пайтон. URL: https://pythonguide.rozh2sch.org.ua

16. Історія мови програмування пайтон. URL: https://www.javatpoint.com/python-history

17. What is python. URL: https://www.infoworld.com/article/3204016/what-is-python-powerful-intuitive-programming.html

18. Бібліотеки пайтон. URL: https://vc.ru/dev/172532-15-python-paketov-kotorye-nuzhno-poprobovat

19. Мокін В. Б, Лосенко А.В, Ящолт А.Р. «Інформаційна технологія аналізу та прогнозування багатохвильової кількості нових випадків захворювань на коронавірус COVID-19 на основі моделі Prophet» : *Вісник ВПІ*, 2020. № 5. С. 65-70.

20. Лопухов Б. Л. Прогнозування кількості хворих на коронавірус в Сполучених Штатах Америки. URL: https://www.kaggle.com/code/lopukhov/covid-19-in-usa-prophet-with-holidays-tuning

21. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт : уклад. Вінниця : ВНТУ, 2021. 42 с.

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Мокін В. Б.
«_19_» _____09_____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ КІЛЬКОСТІ НОВИХ ХВОРИХ НА КОРОНАВІРУС У СПОЛУЧЕНИХ ШТАТАХ АМЕРИКИ»
08-53.МКР.006.02.000.ТЗ

Керівник: к.т.н., доц. каф. САІТ
_____ Козачко О. М
«_19_» _____09_____ 2022 р.

Розробив: студент гр. 2ІСТ-21м
_____ Лопухов Б. Л.
«_19_» _____09_____ 2022 р.

Вінниця 2022

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № 203 по ВНТУ від «14» _09_ 2022 р., та індивідуальне завдання на МКР, затверджене протоколом № 3 засідання кафедри САІТ від «14» ___09____ 2022 р.

2. Джерела розробки:

– Мокін В. Б, Лосенко А.В, Ящолт А.Р. «Інформаційна технологія аналізу та прогнозування багатохвильової кількості нових випадків захворювань на коронавірус COVID-19 на основі моделі Prophet» : *Вісник ВПІ*, 2020. 65-70 с

– COVID-19: Holidays of countries URL: https://www.kaggle.com/vbmokin/covid-19-ua-one-region-forecasting

3. Мета і призначення роботи:

Підвищення точності прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки.

4.Вихідні дані для проведення робіт:

Дані зі статистикою по кількості нових хворих у Сполучених Штатах Америки.

5.Методи дослідження:

Прогнозування та аналіз часових рядів.

6. Етапи роботи і терміни їх виконання:

1. Аналіз предметної області ..... 20.09 – 30.09
2. Вибір оптимальної інформаційної технології. ..... 01.10 – 20.10
3. Розробка інформаційної технології. ..... 21.10 – 05.11
4. Економічна частина. ..... 06.11 – 19.11
5. Оформлення матеріалів до захисту МКР. ..... 20.11 – 30.11

7. Очікувані результати та порядок реалізації:

Отримана інформаційна технологія прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання».

9. Порядок приймання роботи

Публічний захист ..... «_20_» _____12_____ 2022 р.
Початок розробки ..... «_20_» _____09_____ 2022 р.
Граничні терміни виконання МКР ..... «_30_» _____11_____ 2022 р.

Розробив студент групи 2ІСТ-21м _____ Лопухов Б. Л.

Додаток Б

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія прогнозування кількості нових хворих на коронавірус у Сполучених Штатах Америки»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ
Науковий керівник: Козачко О. М. к.т.н., доц. каф. САІТ

Показники звіту подібності Unicheck

Оригінальність	99 %
Схожість	1 %

Аналіз звіту подібності (відмітити потрібне):

- ✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Опис прийнятого рішення:

Робота допускається до захисту

Особа, відповідальна за перевірку Жуков С. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи Лопухов Б. Л.

Керівник роботи Козачко О. М.

Додаток В

Лістинг програми

```
import pandas as pd
import numpy as np
import requests
import seaborn as sns
import openpyxl
import matplotlib
from matplotlib import pyplot as plt
import plotly.express as px
import plotly.graph_objects as go

from PIL import Image
from IPython.display import FileLink

from datetime import date, timedelta, datetime
from fbprophet import Prophet
from fbprophet.make_holidays import make_holidays_df
from fbprophet.diagnostics import cross_validation, performance_metrics
from fbprophet.plot import plot_cross_validation_metric
import holidays
from collections import Counter
import pycountry

from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

import warnings
warnings.simplefilter('ignore')
```

```
# https://api-covid19.rnbo.gov.ua/charts/main-data?mode=ukraine
print(f'Download confirmed daily data from RNBO of Ukraine')
myfile = requests.get('https://api-covid19.rnbo.gov.ua/charts/main-
data?mode=ukraine&country=4907')
open('data', 'wb').write(myfile.content)
data = pd.read_json('data')
data
data['n_confirmed'] = data['confirmed'].diff()
# A new wave from 6 July
data = data[166:].reset_index(drop=True)
data
df2 = data[['dates','n_confirmed']].dropna()
df2 = df2[df2['n_confirmed'] > 0].reset_index(drop=True)
df2['n_confirmed'].plot()

def cut_df(date0: str,
           df:pd.DataFrame(),
           col: str):
    # Deletes all rows of dataframe df with df[col] < date0
    format0 = '%Y-%m-%d'
    df_temp = df.copy()
    df_temp['col_dt'] = pd.to_datetime(df_temp[col], format=format0,
errors='coerce')
    date0_dt = datetime.strptime(date0, format0)
    df_temp = df_temp[df_temp['col_dt'] >= date0_dt]

    return df_temp
```

```
# Type of holidays or pseudo-holidays
        if name == 'the weakening of quarantine':
            holidays_df.loc[last_row+i, 'type'] = 'SI'
        elif name == 'Very comfortable conditions for rest':
            holidays_df.loc[last_row+i, 'type'] = 'meteo'
        elif name == 'Holidays as days of less efficient work of laboratories':
            holidays_df.loc[last_row+i, 'type'] = 'lab'
            holidays_df.loc[last_row+i, 'ds'] = (ds_dt +
timedelta(days=2)).strftime('%Y-%m-%d')
        elif name == 'Weekend quarantine as holidays':
            holidays_df.loc[last_row+i, 'type'] = 'weekend'

        if shift7:
            # Make shift with 7 day ahead
            holidays_df.loc[last_row+i, 'ds'] = (ds_dt +
timedelta(days=7)).strftime('%Y-%m-%d')
        else:
            # Don't make shift with 7 day ahead
            holidays_df.loc[last_row+i, 'ds'] = ds_dt.strftime('%Y-%m-%d')

    return holidays_df.sort_values(by=['ds'])

# Thank to https://github.com/GoogleCloudPlatform/covid-19-open-data
data = pd.read_csv(f"https://storage.googleapis.com/covid19-open-
data/v2/UA/main.csv")

data['stringency_index_jump'] = 0
for i in range(len(data)-1):
    if (data.loc[i+1,'stringency_index'] is not None) and
(data.loc[i,'stringency_index'] is not None) and \
```

```
    (data.loc[i+1,'stringency_index'] < data.loc[i,'stringency_index']):
        data.loc[i+1, 'stringency_index_jump'] = 1
source_gov = 'https://www.bsg.ox.ac.uk/research/research-projects/oxford-covid-19-government-response-tracker'
dates_gov_list = data[data['stringency_index_jump'] == 1]['date'].tolist()
holidays_df = aux_holidays_df_generator(holidays_df, dates_gov_list, 'the weakening of quarantine', source_gov, 2)
plot_with_anomalies(data, ["stringency_index"], {"stringency_index" : "Stringency index and dates of the weakening of quarantine in " + country_main}, 'date', 'stringency_index_jump', 1)

data['rest_comfort'] = 0
data.loc[(data['average_temperature'] >= data['average_temperature'].quantile(.95)) & (data['rainfall'] <= data['rainfall'].quantile(.05)), 'rest_comfort'] = 1
dates_weather_list = data[data['rest_comfort'] == 1]['date'].tolist()
holidays_df = aux_holidays_df_generator(holidays_df, dates_weather_list, 'Very comfortable conditions for rest', 'https://www.ncei.noaa.gov/', 2)
plot_with_anomalies(data, ["average_temperature", "rainfall"], {"average_temperature" : "Average temperature over time in " + country_main, "rainfall" : "rainfall"}, 'date', 'rest_comfort', 1)

data['holidays_date'] = 0
holidays_df = aux_holidays_df_generator(holidays_df, holidays_dates, 'Holidays as days of less efficient work of laboratories',

holidays_weekend_quarantine = ['2020-11-14', '2020-11-15',
                               '2020-11-21', '2020-11-22',
                               '2020-11-28', '2020-11-29']
```

```
holidays_lockdown = ['2021-01-08', '2021-01-09','2021-01-10','2021-01-11','2021-
01-12',
                '2021-01-13', '2021-01-14','2021-01-15','2021-01-16','2021-01-17',
                '2021-01-18', '2021-01-19','2021-01-20','2021-01-21','2021-01-22',
                '2021-01-23', '2021-01-24']

fig = px.line(df2, x="Date", y="Confirmed",
            title="Confirmed cases in " + country_main,
            log_y=False,template='gridon',width=700, height=600)
fig.show()

# For stage 1 of tuning
changepoint_prior_scale_initial_level = 0.15
weekly_season_reg_coef = 1
lower_window_list = [0, -1, -2, -3] # must be exactly 4 values (identical allowed)
upper_window_list = [0, 1, 2, 3] # must be exactly 4 values (identical allowed)
prior_scale_list = [1.5, 2, 2.5, 3] # must be exactly 4 values (identical allowed)
holidays_adaptive = ['holiday', 'SI', 'meteo'] # holidays with adaptive window

# For stage 2 of tuning
several_days_period = 620
several_days_season_reg_coef = 2
several_days_short_period = 4
several_days_short_days_fourier_order = 10
several_days_short_days_season_reg_coef = 0.5 # 0.4
changepoint_prior_scale_list = [0.15, 0.2, 0.25, 0.3] # must be exactly 4 values
(identical allowed)
weekly_fourier_order_list = [2, 4, 6, 8] # must be exactly 4 values (identical
allowed),
```

```
several_days_fourier_order_list = [3, 4, 5, 6] # must be exactly 4 values (identical allowed)
# 0 in fourier_order lists means the absence of this component

# Check length of lists
if (len(lower_window_list) != 4) or (len(upper_window_list) != 4) or \
   (len(prior_scale_list) != 4) or (len(weekly_fourier_order_list) != 4) or (len(several_days_fourier_order_list) != 4):
    print('Number of data is wrong!')

days_to_forecast = 7 # in future (after training data)
days_to_forecast_for_evalution = 7 # on the latest training data - for model training
first_forecasted_date = sorted(list(set(df2['ds'].values)))[-days_to_forecast_for_evalution]
end_forecasted_date = (datetime.strptime(df2['ds'].max(), "%Y-%m-%d")+timedelta(days = days_to_forecast)).strftime("%Y-%m-%d")
first_data_date = df2['ds'].min()

print('The first date of data for modeling is: ' + first_data_date)
print('The first date to perform forecasts for evaluation is: ' + first_forecasted_date)
print('The end date to perform forecasts in future for is: ' + end_forecasted_date)

def export_plot_to_tiff(plot, fig_name):
    DPI = 100 # this value controls the quality of final image
    fig_name = "{}.tiff".format(fig_name)
    buf = io.BytesIO()
    try:
        plot.figure.savefig(buf, format="png", dpi=DPI)
    except AttributeError:
```

```
        plot.savefig(buf, format="png", dpi=DPI)
    else:
        print("Function supports only objects that have savefig() method.")
        return
    buf.seek(0)
    im = Image.open(buf)
    im.save(fig_name)
    im.save(buf, format="tiff")
    buf.close()
    return FileLink(fig_name)

first_eval_index = len(df2)-days_to_forecast_for_evalution
second_eval_index = len(df2)
y_real = df2.tail(days_to_forecast_for_evalution)['y']
y_real_sum = df2.tail(days_to_forecast_for_evalution)['y'].sum()
country_df_val = df2.copy()
country_df_val['ds'] = pd.to_datetime(country_df_val['ds'])
country_df_val = country_df_val[(country_df_val['ds'] >=
pd.to_datetime(first_forecasted_date))]
country_df_val

def make_forecasts(country_df, holidays_df, days_to_forecast,
days_to_forecast_for_evalution, first_forecasted_date):

    def model_training_forecasting(df, forecast_days, holidays_df=None,
mode_main='multiplicative'):
        # Prophet model training and forecasting

        model = Prophet(daily_seasonality=False, weekly_seasonality=False,
yearly_seasonality=False,
```

```
                holidays=holidays_df, changepoint_range=1,
changepoint_prior_scale = changepoint_prior_scale_initial_level,
                seasonality_mode = mode_main)
        model.add_seasonality(name='weekly', period=7, fourier_order=8, mode =
'multiplicative',
                        prior_scale =
changepoint_prior_scale_initial_level/weekly_season_reg_coef)
        model.add_seasonality(name='several_days', period=620, fourier_order=4,
mode = 'multiplicative', prior_scale = 0.3)
        #model.add_seasonality(name='2 weeks', period=14, fourier_order=1, mode =
'multiplicative', prior_scale = 0.15)
        model.fit(df)
        future = model.make_future_dataframe(periods=forecast_days)
        forecast = model.predict(future)
        forecast[forecast['yhat'] < 0]['yhat'] = 0
        return model, forecast


    cols_w = ['ds', 'trend', 'yhat', 'yhat_lower', 'yhat_upper', 'trend_lower',
'trend_upper', 'additive_terms', 'additive_terms_lower', 'additive_terms_upper',
            'multiplicative_terms','multiplicative_terms_lower',
'multiplicative_terms_upper', 'weekly', 'weekly_lower', 'weekly_upper']
    cols_h = ['ds', 'trend', 'yhat', 'yhat_lower', 'yhat_upper', 'trend_lower',
'trend_upper', 'additive_terms', 'additive_terms_lower', 'additive_terms_upper',
            'holidays', 'holidays_lower', 'holidays_upper',
'multiplicative_terms','multiplicative_terms_lower', 'multiplicative_terms_upper',
'weekly',
            'weekly_lower', 'weekly_upper']
    #mode_main_list = ['additive', 'multiplicative']
    mode_main_list = ['multiplicative'] # take only this mode
    relative_errors_holidays = []
```

```
    counter = 0
    results = pd.DataFrame(columns=['Conf_real', 'Conf_pred', 'Conf_pred_h',
'mode', 'n_h', 'err', 'err_h', 'prior_scale', 'how_less, %'])
    country_holidays_df = holidays_df[holidays_df['code'] ==
country_iso_alpha][['ds', 'holiday', 'lower_window', 'upper_window', 'prior_scale',
'type']].reset_index(drop=True)
    country_dfs = []
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ КІЛЬКОСТІ НОВИХ ХВОРИХ НА КОРОНАВІРУС У СПОЛУЧЕНИХ ШТАТАХ АМЕРИКИ

Виконав: студент гр. 2ІСТ-21м

_____ Лопухов Б.Л.

«_01_» _____12_____ 2022 р.

Керівник: к.т.н., доцент

_____ Козачко О. М.

«_02_» _____12_____ 2022 р.

Нормоконтроль: к.т.н., доцент

_____ Жуков С. О.

«_02_» _____12_____ 2022 р.

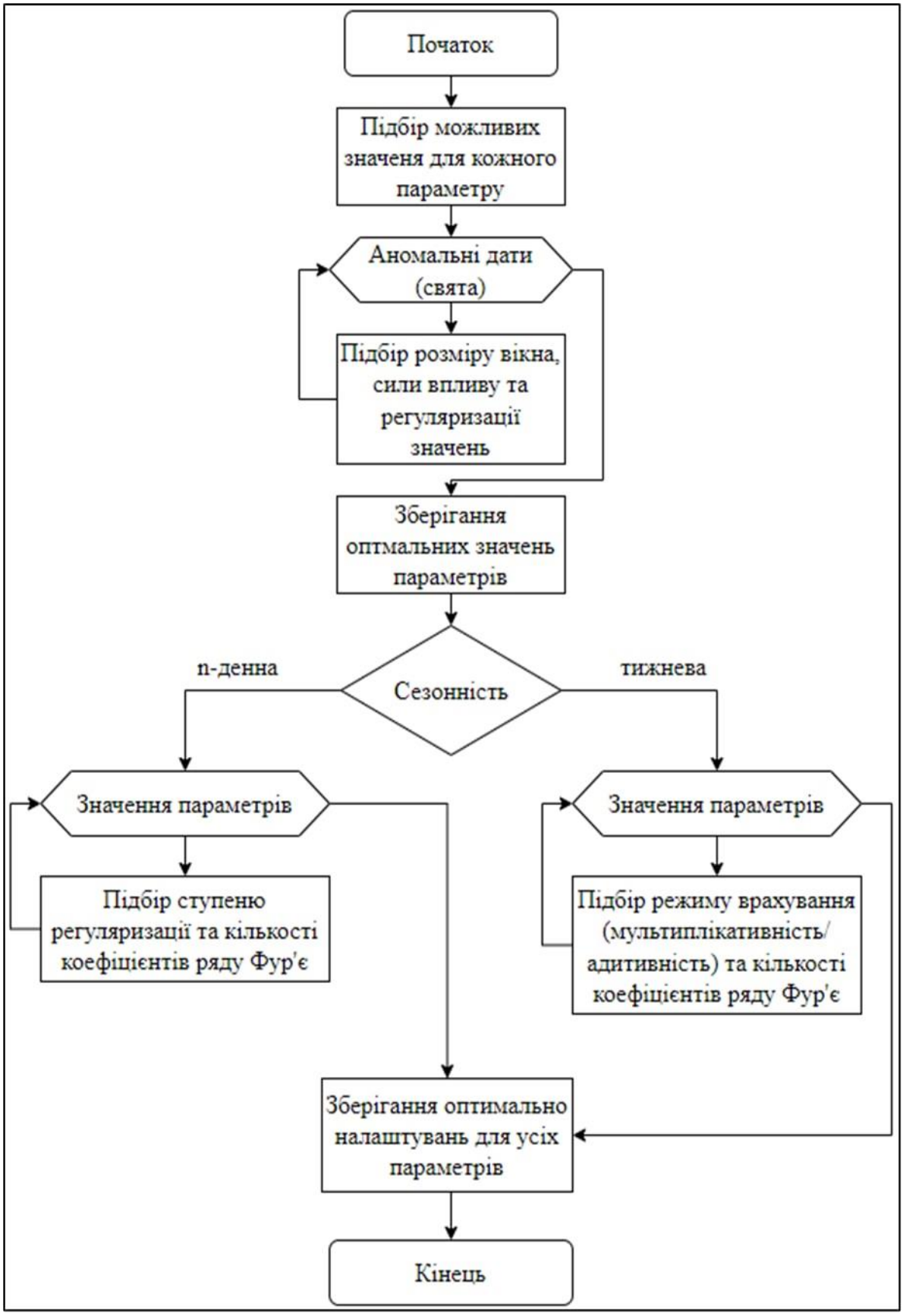

Рисунок Г.1 – Блок-схема алгоритму роботи інформаційної технології

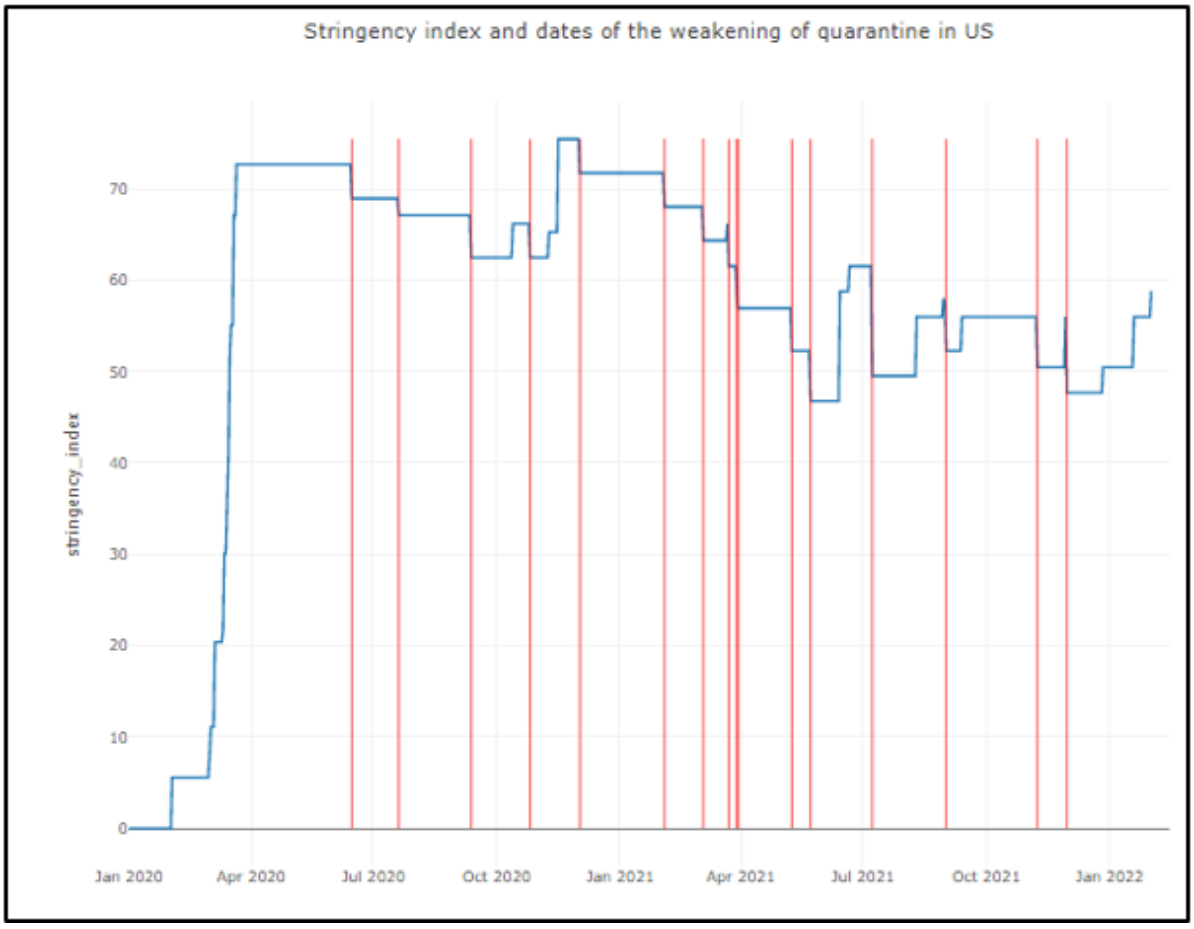

Рисунок Г.2 – Графік критерія Оксфордської лабораторії Stringence index

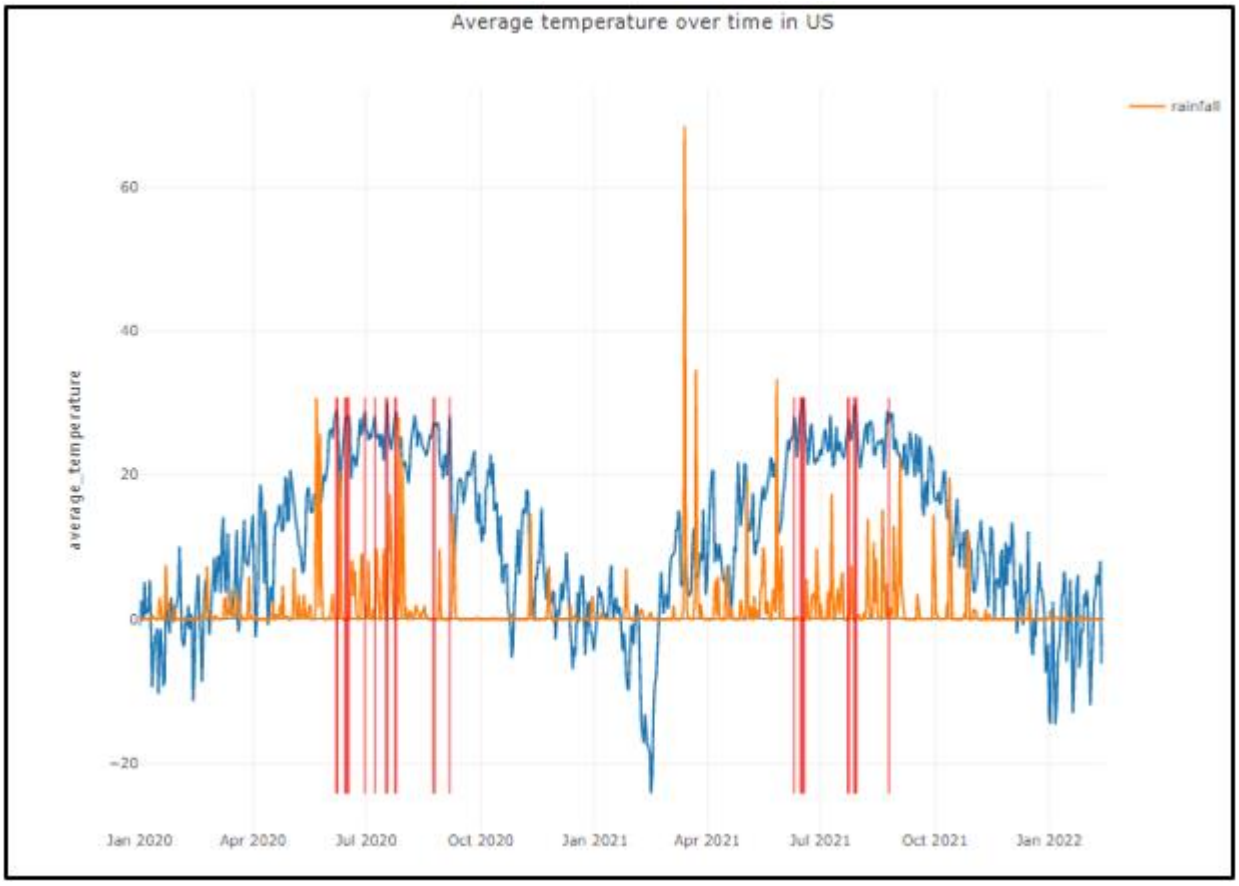

Рисунок Г.3 – Графік залежностей опадів і середньої температури

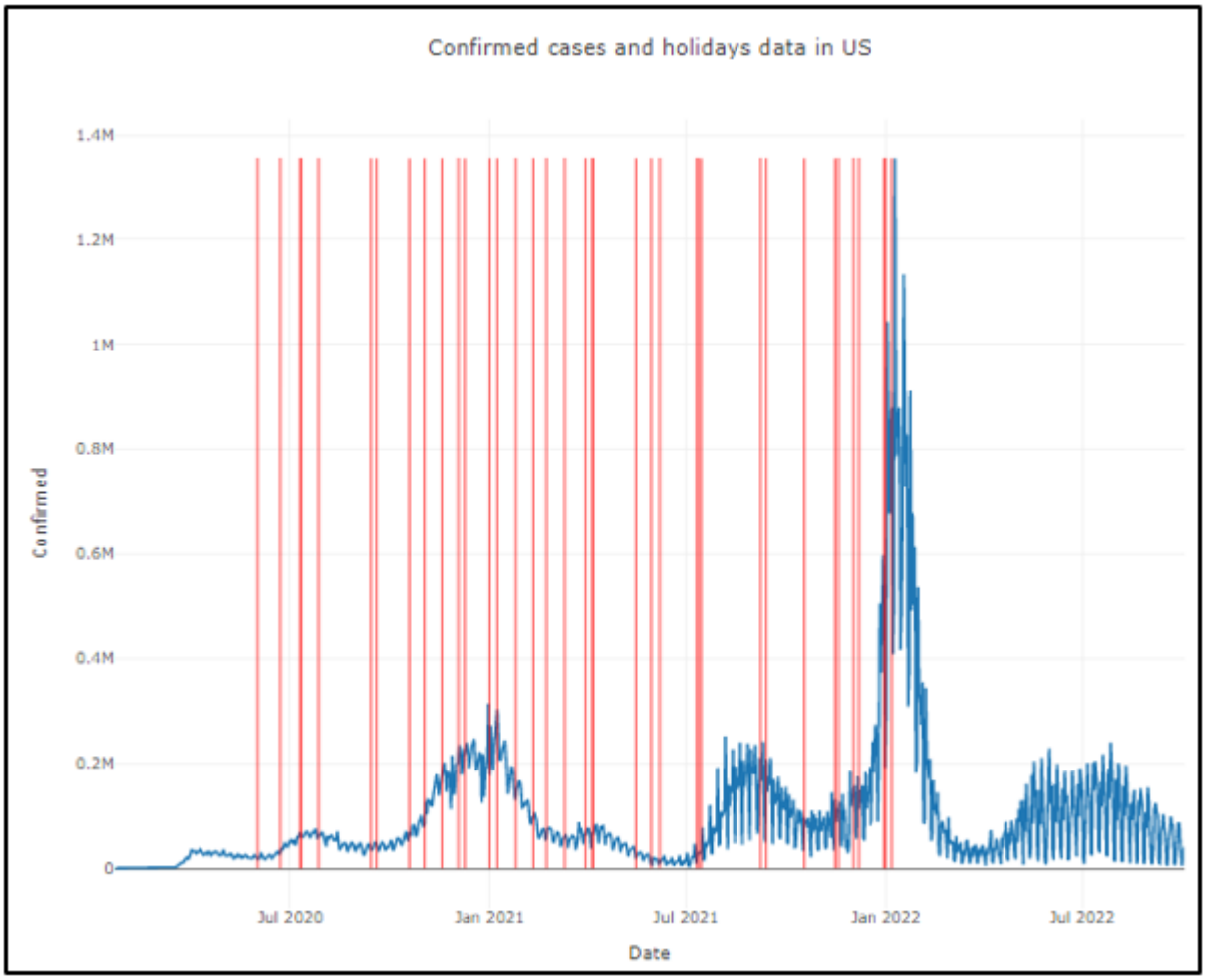

Рисунок Г.4 – Графік кількість нових хворих у США з накладеними аномальними датами

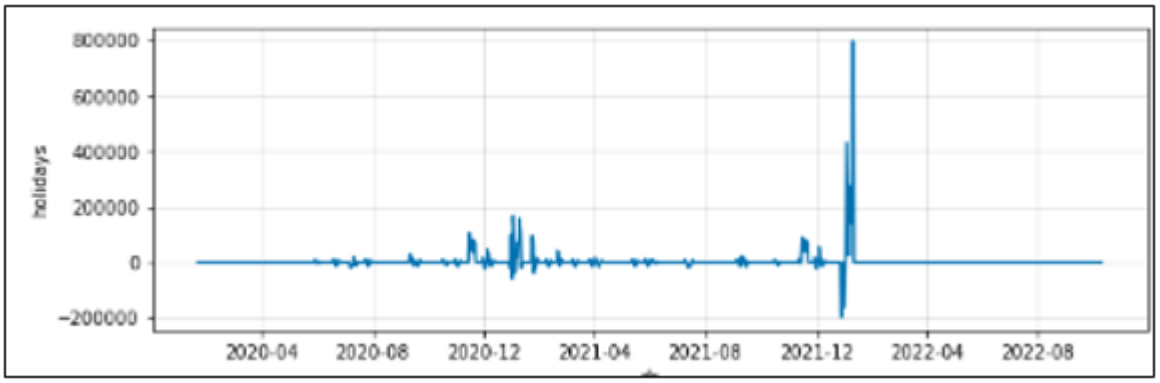

Рисунок Г.5 – Графік впливу свят на прогноз

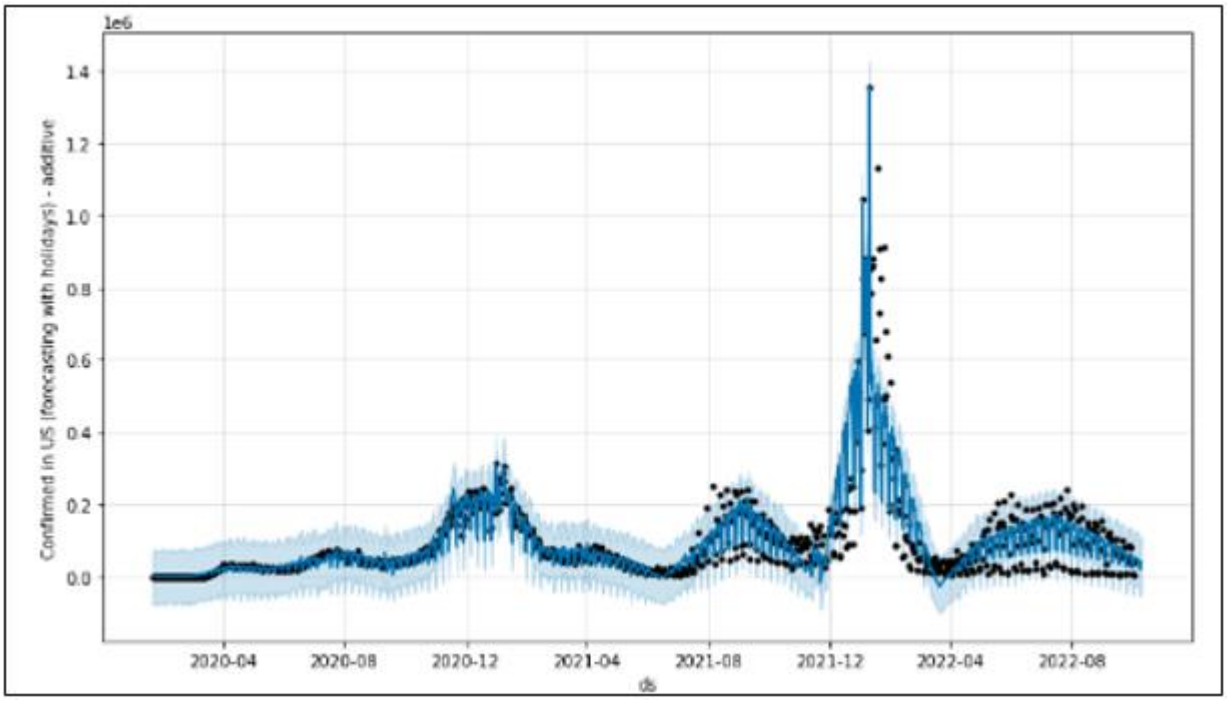

Рисунок Г.6 – Результат прогнозування кількості нових хворих на коронавірус у США методом Facebook Prophet