КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

Кваліфікаційна робота

на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:

КОРЕЛЯЦІЙНИЙ АНАЛІЗ ДАНИХ ІНТЕРНЕТ-АКТИВНОСТІ КОРИСТУВАЧА ТА ПОКАЗНИКІВ У СФЕРАХ ЙОГО ЖИТТЯ

Виконала студентка 4 курсу
Анастасія ЦИБИК _____

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Ростислав ТРОХИМЧУК _____

(підпис)

Засвідчую, що в цій курсовій роботі немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри теоретичної кібернетики
«____» _____ 2023 р.,
протокол № ___
Завідувач кафедри
доктор фіз.-мат. наук, професор
Юрій КРАК _____

(підпис)

КИЇВ – 2023

РЕФЕРАТ

Обсяг роботи складає 68 сторінок, в ній містяться 29 ілюстрацій, 8 таблиць, використано 27 джерел посилань.

ОБРОБКА ДАНИХ, СТАТИСТИЧНИЙ АНАЛІЗ, ВІЗУАЛІЗАЦІЯ ДАНИХ, ГРАФІЧНИЙ АНАЛІЗ, КОРЕЛЯЦІЙНИЙ АНАЛІЗ, PYTHON, PANDAS, SEABORN, STATSMODELS, MATPLOTLIB, NUMPY

Об’єктом роботи є процес аналізу інтернет-активності користувача та показників у сферах його життя, за даними із різних джерел.

Метою роботи є розробка інструменту для здійснення збору, огляду, підготовки, візуалізації та аналізу даних інтернет-активності користувача та показників у сферах його життя. Також метою роботи є на основі виявлених кореляцій між активністю особи в інтернет-просторі, її станом та явищами реального світу оцінити отримані результати та зробити припущення щодо інших можливих значущих факторів впливу, створити базу для подальших досліджень.

Сферою застосування цього інструменту є соціологічні дослідження, бізнес-аналітика для соціальних мереж та інших цифрових застосунків. У вигляді більш дружнього для користувача додатку може використовуватись як потужний засіб для саморефлексії та оптимізації його життя.

Інструменти розробки: мова програмування Python, модулі Pandas, Seaborn, Numpy, Statsmodels та Matplotlib.

Результатом роботи є таблиці отриманих статистичних метрик, а також графічні представлення даних у вигляді графіків та діаграм.

ЗМІСТ

ВСТУП

Оцінка сучасного об’єкта розробки. З плином часу, невблаганний перебіг історії безупинно вносив зміни в середовище існування людини, все більш радикальні з кожною новою епохою. З моменту зародження людства, перші люди жили в гармонії з природою, у близькій взаємодії з довкіллям. Однак з розвитком технологій, урбанізацією та прогресом суспільства ми стали все більш відчуженими від природних ритмів та умов, поринаючи у неприродне і незвичне середовище. Еволюційний розвиток людини не здатний утримувати темп таких змін, забезпечити якостями, необхідними для пристосування. У процесі еволюції наш організм і мозок адаптувалися до умов, властивих життю на Савані в період пізнього плейстоцену, приблизно від 2,6 мільйонів до 11,7 тисяч років тому, де переважали стабільність, простота та і неминучий тісний зв'язок із природою [1]. Проте сучасне середовище існування характеризується швидким темпом технологічних, соціальних та культурних змін, які наша еволюція не встигає врахувати. Це створює невідповідність між нашими природними можливостями адаптації та вимогами сучасного світу, викликаючи стрес, психологічні та фізичні навантаження на організм, ускладнюючи, якщо не унеможливлюючи, завдання досягнення відчуття задоволення життям та благополуччя.

Усвідомлення цієї проблеми змушує замислитись про можливі методи полегшення долі сучасної людини, про те, як допомогти їй компенсувати власну дезадаптацію зовнішніми інструментами, полегшити їй досягнення благополуччя. Для цього необхідно як надання індивідуальної можливості для людини визначити свою оптимальну поведінку в не гармонійному середовищі, так і аналогічно показати колективним об'єднанням та організаціям шляхи зміни свого впливу на середовище у позитивному напрямку та покращення послуг, що надаються.

Часи змін вимагають від людей здатності адаптуватися до нових умов та організовувати своє життя, щоб уникнути відчуття безладу та нестачі часу. У сучасному світі люди стикаються з низкою факторів, які можуть створювати хаос у їхньому житті. Нестійка економічна ситуація, інформаційний перенасичений потік, постійні зміни та несподіванки можуть негативно впливати на психологічний стан та організацію життя людей. Однак, замість того, щоб піддатися цьому хаосу, важливо розробити стратегії самоаналізу та систематизації, які допоможуть нам пристосуватися до нових умов і створити більш кероване, організоване та свідоме життя. Важливість об'єднання даних з різних джерел [2], їх аналізу та наявності доступних інструментів систематизації стає критичною, а особливу роль у цьому процесі відіграють корпорації, що збирають дані про користувачів, надають послуги та безпосередньо впливають на життя своїх клієнтів.

Актуальність роботи та підстави для її виконання. З розвитком технологій та інтернету з'явилося багато інструментів та методів, які допомагають людям систематизувати своє життя, а комерційним компаніям проводити бізнес-аналітику. Мобільні програми для планування та управління часом, органайзери, онлайн-сервіси для управління завданнями та проектами – все це надає людям можливості для організації їхнього життя та підвищення ефективності. Однак, наявні інструменти часто є закритими проектами, недоступними матеріально, або складними у використанні. Також кожен з них зазвичай може дати виключно однобічний і обмежений погляд на ситуацію, не дозволяючи повною мірою охопити всю картину і перетворити зібрані дані на справді цінну інформацію, здатну призвести до кардинальних змін як на індивідуальному, так і на громадському рівні.

У разі повсюдного поширення доступного інструменту для більш повного комплексного аналізу можна значно краще зрозуміти оголошену проблему, дати можливість індивідуумам та кооперативам визначати свої

сильні та слабкі сторони, галузі, які потребують особливої уваги, та покращення. Це дозволить досягти більшої усвідомленості, ефективності, та поінформованості в прийнятті рішень на багатьох рівнях структури суспільства.

Мета і завдання роботи. Саме з огляду на актуальність та серйозність оголошеної проблеми, а також відсутність наявних задовільних інструментів її вирішення, **метою роботи** є розробка інструменту для здійснення збору, огляду, підготовки, візуалізації та аналізу даних інтернет-активності користувача та показників у сферах його життя.

Аби мета була досягнута, необхідно:

1. Визначити джерела підлеглі аналізу, зібрати та уніфікувати дані.
2. Обчислити статистичні метрики, побудувати графіки.
3. Провести статистичний аналіз усіх показників.
4. Оцінити отримані результати та зробити висновки.
5. Сформулювати шляхи поглиблення подальших досліджень.

Об’єктом дослідження є дані, що належать одному анонімному користувачеві, у вигляді кількісних та якісних показників за кожен окремий день на часовому проміжку з 1 січня 2023 року до 20 травня 2023 року (140 днів).

Предметом дослідження роботи є процес аналізу інтернет-активності користувача та показників у сферах його життя, за даними із різних джерел.

Засоби реалізації: мова програмування Python, модулі Pandas, Seaborn, Numpy, Statsmodels та Matplotlib.

1 ПРЕДМЕТНА ОБЛАСТЬ

1.1 Вплив інтернет-ресурсів на людину

Інтернет-ресурси, такі як соціальні мережі, мобільні та веб-додатки, месенджери, цифрові платіжні системи, цифрові мапи та відстежувачі локації і т.д., на сьогоднішній день досягли значного рівня впливу, який охоплює безліч різноманітних сфер людського життя, зокрема комунікацію, комерцію, розваги та навіть політику. Вони зробили значно легшою підтримку зв’язку з друзями, сім’єю, колегами та навіть анонімними незнайомцями, незалежно від фізичного розташування, спростили обмін повідомленнями, фотографіями та відео, сприяють формуванню спільнот та розширенню соціальних контактів, надають нескінченні можливості для отримання інформації - пошук знань, читання новин, опанування нових навичок, обмін досвідом та здобуття освіти онлайн. Цифрові платіжні системи дозволяють людям здійснювати швидкі та безпечні фінансові операції, надають зручність при онлайн-покупках, переказах грошей та оплаті послуг. Мобільні програми та цифрові карти полегшують переміщення та навігацію в незнайомих місцях, пропонують маршрути, підказки про дорожню обстановку, інформацію про пам'ятки та місця для відвідування. Різні пристрої та програми для відстеження локації та здоров'я дозволяють людям контролювати свою фізичну активність, серцевий ритм, сон та інші параметри здоров'я - це допомагає їм приймати більш поінформовані рішення про свій спосіб життя та піклуватися про своє благополуччя. Інтернет та цифровізація відкрили для бізнесу нові шляхи досягти своїх клієнтів та потенційних покупців. А одне із найпомітніших зрушень інтернет-ресурси викликали у сфері розваг, забезпечивши можливість миттєвого доступу до широкого спектру контенту: музики,

фільмів, телевізійних шоу та ігр. Неосяжна різноманітність додатків та соціальних мереж особливо сильно приваблює молодше покоління, здійснюючи на нього найбільший вплив [3].

Однак, незважаючи на безліч, здавалося б переваг, невідомо, якою мірою більшість змін, принесених цифровим світом, природні та корисні для життя людини. Як вже безумовно доведений, так і потенційний негативний вплив інтернет-ресурсів є колосальним. У людей виникає залежність від постійної присутності в соціальних мережах, іграх або надмірного використання мобільних додатків, вони можуть відчувати тиск, щоб створити і підтримувати певний образ або не відставати від постійного потоку інформації. Це призводить до втрати часу, зменшення продуктивності та відволікання від важливіших завдань та взаємодій у реальному житті. Використання соціальних мереж і месенджерів може сприяти зниженню особистої комунікації та взаємодії віч-на-віч. Це може призвести до відчуття самотності, низької самооцінки, погіршення якості міжособистісних відносин та виникнення конфліктів. Використання інтернет-ресурсів пов'язане з ризиком розвитку стресу, тривожності, та депресії. Порівняння себе з ідеалізованими образами на соціальних платформах може викликати незадоволення своїм життям та створювати спотворене уявлення про реальність. Використання цифрових платіжних систем, цифрових карток та відстежувачів локації може супроводжуватися ризиком порушення кібербезпеки. Особиста інформація може бути зібрана або навіть вкрадена без згоди людини, а потім використана на шкоду. Соціальні мережі також викликають занепокоєння щодо конфіденційності, оскільки треті сторони можуть отримати доступ до особистої інформації, яка поширюється в Інтернеті, без відома або згоди користувачів, що призведе до негативних наслідків. Соціальні мережі можуть бути сприятливою платформою для кібербулінгу або масової дезінформації. Використання мобільних пристроїв та комп'ютерів протягом тривалого часу

може призводити до проблем із зором, поставою та фізичною активністю. Також постійна доступність і занурення в цифровий світ може викликати розлади сну. Різні платформи регулярно стикаються з суперечками, включаючи занепокоєння щодо практики збору даних і потенційних ризиків для національної безпеки, а також їх впливу на психічне здоров’я користувачів.

Отже, зважаючи на здобуття цифровим світом помітної і вражаючої влади над життям та розумом людей, важливо ставитися до них з відповідною серйозністю та обережністю. Залежно від того, які інтернет-ресурси, яким чином і наскільки часто людина використовує, вплив буде якісно і кількісно відрізнятися. Варто остерігатися негативного впливу, відслідковувати, коли інтернет стає джерелом стресу чи тривоги, або марнує час, та припиняти надмірне користування. Загалом важливо користуватися інтернет-ресурсами помірно і помірковано та переконатися, що вони вписуються у ваш загальний спосіб життя здоровим і збалансованим способом.

1.2 Інші фактори впливу на життя людини

Людина схильна до впливу різних факторів, які можуть істотно впливати на її фізичний і психологічний стан, прийняті рішення та загальну життєву активність. Можливість дослідити та зрозуміти взаємозв'язок між цими факторами є ключовим аспектом для покращення якості життя та прийняття усвідомлених рішень.

Наприклад, погодні умови, такі як температура, вологість, опади та сонячне світло, можуть впливати на настрій, енергію та фізичне самопочуття людини. Дослідження [4] показують, що деякі люди почуваються активнішими і бадьорішими в сонячні дні, а також можуть відчувати млявість і втому в похмурі. Вивчення цього взаємозв'язку

допомагає людям краще зрозуміти, як погода впливає на їхнє життя і як їм адаптуватися до умов, що змінюються.

Деякі люди вважають, що фази місяця можуть впливати на їх емоційний стан та поведінку. Наприклад, повний місяць часто асоціюється з посиленням емоцій та занепокоєнням. Дослідники намагаються виявити зв'язок між фазами місяця та різними аспектами життя, включаючи сон, настрій, поведінку та здоров'я.

У жінок менструальний цикл може впливати на їх фізичний та емоційний стан. Дослідження [5] показують, що деякі жінки можуть відчувати зміни настрою, рівню енергії та фізичної активності у різні фази свого циклу. Вивчення цих взаємозв'язків допомагає краще зрозуміти фізіологічні та психологічні аспекти менструального циклу та розробити стратегії управління цими змінами.

Кількість витрат і доходів також відіграють важливу роль життя людини. Фінансова стабільність може впливати на емоційне благополуччя, рівень стресу та можливості для досягнення особистих та професійних цілей. Дослідження цього взаємозв'язку допомагає людям краще керувати своїми фінансами, планувати та приймати усвідомлені фінансові рішення.

Рівень фізичної активності та переміщень також можуть впливати на загальний стан здоров'я та добробут людини. Дослідження [6] показують, що регулярна фізична активність сприяє покращенню фізичного та психологічного здоров'я, підвищує настрій та енергію. Вивчення цього взаємозв'язку допомагає людям зрозуміти важливість фізичної активності та розробити оптимальні стратегії підтримки активного життя.

Взаємозв'язок між чинниками, які впливають життя людей, має значення для розуміння і оптимізації їхнього життя. Дослідження, базовані на статистичному аналізі та кореляційних методах, допомагають розкрити ці зв'язки та надають нам цінну інформацію для прийняття усвідомлених рішень, покращення здоров'я та підвищення якості життя. Використання

цієї інформації та проведення особистого аналізу дозволяють людям краще зрозуміти та керувати їхнім життям на основі цих факторів.

Також дуже важливо усвідомлювати, що кожна людина є унікальним індивідуумом зі своїми унікальними життєвими обставинами, цілями та потребами. Саме тому дослідження факторів впливу на життя також важливо проводити індивідуально для кожної людини, а не тільки розглядати її просто як цифру у великій статистиці. Розуміння специфіки впливу цих факторів саме на себе дозволяє краще зрозуміти власні потреби та переваги, типові реакції на подразники. Через аналіз та дослідження факторів впливу людського життя, можна відкрити нові можливості для саморозвитку, покращення здоров'я та досягнення особистого благополуччя.

1.3 Використання прав, зазначених у законах про збір даних

Закони, які регулюють збір, обробку та надання даних, відіграють важливу роль у захисті прав споживачів та забезпеченні прозорості та контролю за їхніми даними.

Загальний регламент захисту даних GDPR [7] є одним із найбільш значущих міжнародних законодавчих актів, що регулюють збір, використання та обробку персональних даних громадян Європейського Союзу. GDPR було введено в дію у 2018 році та забезпечує високий рівень захисту приватності та контролю над персональними даними.

Цей регламент зобов'язує компанії дотримуватись певних принципів при зборі та обробці персональних даних, таких як законність, справедливість та прозорість. Він також встановлює права громадян, включаючи право на доступ до своїх даних, право на виправлення та видалення інформації, право на обмеження обробки, право на передачу даних та право на заперечення проти обробки даних.

GDPR вимагає від компаній вживати заходів щодо захисту даних, включаючи реалізацію відповідних технічних та організаційних заходів для забезпечення безпеки даних. Він також передбачає санкції та штрафи за порушення правил та неправомірну обробку персональних даних.

Загальний регламент захисту даних (GDPR) є важливим інструментом, що забезпечує захист приватності та контроль за персональними даними громадян Європейського Союзу, а також підвищує відповідальність компаній щодо обробки даних.

В Україні також затверджено "Закон України про захист персональних даних" [8]. Він встановлює принципи та правила обробки персональних даних, а також визначає права та обов'язки суб'єктів даних та операторів, які збирають та обробляють ці дані.

Закон зобов'язує операторів забезпечувати конфіденційність та безпеку персональних даних, зібраних у користувачів. Він передбачає згоду суб'єкта на збирання та обробку його персональних даних, а також вимагає інформувати суб'єктів про цілі збору, обробки та використання їх даних.

Важливим аспектом українського законодавства є обов'язок операторів надавати суб'єктам дані, зібрані про них, на запит. Це означає, що користувачі мають право запросити доступ до своїх персональних даних, а також вносити виправлення або видаляти інформацію за потреби.

У сучасному цифровому світі компанії збирають і зберігають величезну кількість даних про своїх користувачів: уподобання, звички, поведінка в мережі, покупки та багато іншого. Але скільки з них справді усвідомлюють, що закони про захист даних, такі як GDPR, дають право доступу до цих даних? Необхідно використовувати це право, запитати у компаній усі зібрані дані. Це відкриває двері до розуміння того, як компанії бачать користувачів та використовують їхні дані. Дослідження та аналіз цих даних можуть стати сильним інструментом для покращення та оптимізації людських життів, виявити цікаві закономірності, тренди та взаємозв'язки.

1.4 Використання статистичного аналізу

Статистичний аналіз відіграє важливу роль у розумінні складних взаємозв'язків та прийнятті поінформованих рішень, широко використовується у різних галузях, включаючи соціальні науки, бізнес, медицину. Методи статистичного аналізу дозволяють досліджувати зв'язки між змінними, виявляти закономірності, передбачати результати, виявляти вплив факторів на різні аспекти життя та перевіряти гіпотези.

Аналіз даних постійно розвивається разом із прогресом технологій та доступністю великих обсягів даних. Сучасні методи аналізу дозволяють визначити складніші взаємозв'язки між змінними, враховувати безліч факторів та використовувати більш точні та ефективні моделі для передбачення та інтерпретації даних.

Існує велика різноманітність методів та підходів, специфіка застосування яких залежить від цілей дослідження та типу даних, з якими працює науковець. Кожен метод має свої особливості та передбачає певні передумови.

Використання графічних зображень, зокрема діаграм, графіків та інших наочних посібників, для аналізу та інтерпретації інформації – це потужний інструмент для виявлення тенденцій і закономірностей, які можуть бути не відразу очевидними при перегляді необроблених даних.

Кореляційний аналіз використовується для вимірювання ступеня зв'язку між двома чи більше змінними. Він дозволяє визначити, чи є статистично значущий зв'язок між змінними та який характер цього зв'язку (позитивний, негативний чи його відсутність).

Загалом, статистичний аналіз дозволяє дослідникам та приймаючим рішенням зрозуміти складні взаємозв'язки та зробити висновки. Він відіграє важливу роль у розробці оптимальних моделей поведінки та політик, які

сприяють покращенню життя окремих людей та суспільства загалом. На основі результатів аналізу можна розробляти програми та ініціативи, спрямовані на вирішення соціальних проблем, оптимізацію бізнес-процесів, покращення послуг та багато іншого.

2 ПОСТАНОВКА ЗАДАЧІ ТА ПРОЕКТУВАННЯ

2.1 Постановка задачі

Спочатку необхідно визначити множину різних джерел даних, на основі яких проводитиметься аналіз. Для кожного джерела, що розглядається, знайти спосіб отримати доступ до шуканих даних, завантажити їх в оптимальному форматі з доступних варіантів. Визначити які саме показники відповідають характеру проводимого аналізу, провести вручну всі необхідні маніпуляції для переходу до наступної програмної обробки.

Дослідження буде проводитись на даних у вигляді кількісних та якісних показників за кожен окремий день на часовому проміжку з 01.01.2023 до 20.05.2023 (разом 140 днів). Усі зібрані дані належать одному анонімному користувачеві.

Вибрані джерела та показники:

1. Chronology Google Maps (дата, місто знаходження користувача).
2. Погода (дата, день тижня, час сходу сонця, час заходу сонця, середня температура за день, середній тиск за день, середня вологість за день, погодній статус).
3. Фази місяця (дата, фаза місяця).
4. Flo (дата, фаза менструального циклу, кількість здійснених дій у додатку за день).
5. TikTok (дата, кількість вподобаних відео за день, кількість переглянутих відео за день).
6. Telegram (дата, кількість відправлених повідомлень за день, кількість отриманих повідомлень за день).
7. Discord (дата, сумарна тривалість дзвінків за день).

8. Privat24.ua (дата, кількість витрачених коштів за день, кількість отриманих коштів за день).
9. Google Fit (дата, тривалість активності за день, кількість спалених калорій, дистанція переміщення за день, кількість зроблених кроків за день).
10. Google Chrome (дата, кількість відвіданих веб-сторінок за день).

Оптимальним вибором для подальшої обробки та аналізу є мова програмування Python [9], завдяки наявності усіх необхідних модулів для обробки, візуалізації даних та реалізації аналізу.

Кожен з отриманих файлів зчитати та виконати попередню обробку даних: привести до зручного формату, перетворити окремі записи у кількісні показники за день, відсортувати, уніфікувати формат дат, відсікти зайві показники за межами обраного проміжку або за межами інтересів дослідження.

За допомогою модуля Pandas [10] створити з цих файлів датафрейми.

За допомогою модуля Numpy [11] обчислити основні статистичні метрики для кожного з датафреймів: середнє арифметичне, медіана, мода, дисперсія, стандартне відхилення.

За допомогою модулів Matplotlib та Seaborn [12] побудувати графіки та діаграми для візуального огляду та аналізу даних кожного датафрейму.

Провести кореляційний аналіз усіх показників.

На основі виявлених кореляцій між активністю суб'єкта в інтернет-просторі, його станом та явищами реального світу оцінити отримані результати та зробити припущення щодо інших можливих значущих факторів впливу, створити базу для подальших досліджень.

2.2 Кореляційний аналіз

Кореляційний аналіз [13] є потужним інструментом статистичного аналізу, який використовується для вивчення взаємозв'язків між змінними. Він дозволяє визначити наявність, щільність, силу, форму та спрямованість зв'язку між двома або більше змінними, а також передбачати значення однієї змінної на основі іншої чи кількох інших змінних.

Коефіцієнт кореляції – це числова міра, яка показує ступінь залежності між двома змінними (позитивний, негативний або рівний нулю). Існують різні типи коефіцієнтів кореляції, такі як коефіцієнт Пірсона, коефіцієнт Спірмена та коефіцієнт Кендалла.

Коефіцієнт Пірсона - це один з найбільш поширених коефіцієнтів кореляції, що використовуються в статистичному аналізі. Він вимірює лінійну залежність між двома неперервними змінними, позначається символом "r" і може приймати значення від -1 до +1.

Значення +1 свідчить про позитивний лінійний зв'язок, тобто, коли обидва показники змінюються в одному напрямку. Значення -1 свідчить про негативний лінійний зв'язок, тобто, коли показники змінюються у протилежних напрямках. Значення 0 вказує на відсутність лінійного зв'язку між змінними.

Розрахунок коефіцієнта Пірсона ґрунтується на сумах добутків відхилень змінних від своїх середніх значень:

$$r = \sum_{i=1}^{n} \frac{(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum_{j=1}^{n}(X - \bar{X})^2}\sqrt{\sum_{j=1}^{n}(Y - \underline{Y})^2}},$$

де X та Y - значення змінних, $\bar{X}$ та $\bar{Y}$ - середні значення змінних, n - кількість спостережень

Коефіцієнт Спірмена - це статистичний показник, що використовується для вимірювання монотонного зв'язку між двома змінними, позначається символом "ρ" (ро) і може набувати значень від -1 до

+1, так само як і коефіцієнт Пірсона. Він ґрунтується на рангових значеннях змінних, тобто на їх порядкових позиціях у впорядкованому наборі даних.

Коефіцієнт Спірмена чутливий до порядкових змін показників і може бути використаний для оцінки зв'язку між різними типами змінних, включаючи номінальні, рангові та інтервальні, і дозволяє оцінити рівень зв'язку між змінними, коли дані не відповідають вимогам лінійної кореляції.

Розрахунок коефіцієнта Спірмена включає такі кроки:

1. Ранжуються значення кожної змінної від найменшого до найбільшого. Для однакових значень ранг дорівнює середньому числу їхніх позицій в порядку зростанні величини.
2. Розраховуються різниці у рангах кожної пари спостережень:

$$d_i = x_i - y_i$$

3. Коефіцієнт Спірмена обчислюється за формулою:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n\ (n^2 - 1))},$$

де d - різниця рангів пари спостережень, n - кількість спостережень.

Коефіцієнт Кендалла - це статистичний показник, що використовується для вимірювання ступеня узгодженості або залежності між двома ранжованими змінними, позначається символом "τ" (тау) і може приймати значення від -1 до +1, аналогічно іншим коефіцієнтам кореляції. Він не є параметричним методом оцінки кореляції і заснований на порівнянні пар рангових значень двох змінних. Коефіцієнт Кендалла підходить для оцінки залежностей, які не обов'язково лінійні і можуть бути монотонними (тобто значення змінних рухаються в одному напрямку, але не обов'язково з постійною швидкістю). Він також вимагає нормального розподілу даних і може бути застосований до різних типів змінних.

Розрахунок коефіцієнта Кендалла включає наступні кроки:

1. Ранжуються значення кожної змінної від найменшого до найбільшого.

2. Розраховуються різниці у рангах кожної пари спостережень.
3. Визначаються пари узгоджених (коли різниці у рангах мають однаковий знак) та неузгоджених (коли різниці у рангах мають різні знаки).
4. Обчислюється коефіцієнт Кендалла за формулою:

$$\tau = \frac{s_1 - s_2}{\frac{1}{2}n\,(n-1))},$$

де s_1 - кількість узгоджених пар, s_2 - кількість неузгоджених пар, n - кількість спостережень

Визначення статистичної значущості у кореляційному аналізі дозволяє сказати, наскільки ймовірно знайдений зв'язок між змінними є реальним, а не результатом випадковості. Зазвичай використовується перевірка гіпотези про рівність нулю коефіцієнта кореляції. Результати статистичного тесту виражаються через p-значення, яке вказує на ймовірність отримання таких або більш екстремальних результатів, якщо нульова гіпотеза (відсутність зв'язку) правильна. Якщо p-значення нижче встановленого рівня значущості (зазвичай 0,05), можна вважати, що зв'язок є статистично значущим.

2.3 Мова програмування Python та використані модулі

Python - це високорівнева інтерпретована мова програмування. Python має зрозумілий та виразний синтаксис, який робить його доступним для новачків у програмуванні. Він дозволяє розробникам писати чистий код. Python підтримує як процедурне, так і об'єктно-орієнтоване програмування та має можливості для асинхронного програмування.

Python має величезний набір стандартних бібліотек (модулів), які надають широкий спектр функцій та можливостей. Крім того, є безліч сторонніх бібліотек, розроблених спільнотою, які дозволяють вирішувати різноманітні завдання, від наукових обчислень до веб-розробки.

Pandas - це бібліотека мови програмування Python, яка надає потужні та гнучкі інструменти для аналізу та обробки даних, високопродуктивні структури, такі як DataFrame (табличний формат) та Series (одномірний формат), які дозволяють легко працювати з даними.

За допомогою pandas можна завантажувати та об’єднувати дані з різних джерел, таких як CSV-файли, бази даних або веб-сервіси. Цей модуль дозволяє виконувати операції з фільтрації, сортування, групування та агрегації даних. Також pandas пропонує широкий набір функцій для обробки пропущених значень, перетворення типів даних та роботи з датами та часом. Ключовою особливістю також є інтеграція з іншими бібліотеками Python, такими як NumPy та Matplotlib, для більш складного аналізу та візуалізації даних.

Matplotlib – це бібліотека для візуалізації даних у мові програмування Python. Вона надає широкий набір функцій та інструментів для створення графіків, діаграм та візуальних уявлень даних, включаючи лінійні графіки, стовпчасті діаграми, кругові діаграми, гістограми та багато іншого. Вона надає гнучкі можливості налаштування зовнішнього вигляду графіків, такі як кольори, шрифти, розміри та підписи осей.

Matplotlib надає потужні інструменти для візуалізації даних, дозволяючи дослідникам та аналітикам подавати свої результати у наочній та зрозумілій формі.

Seaborn [14] – це бібліотека для візуалізації даних у мові програмування Python, яка є надбудовою над бібліотекою Matplotlib. Вона надає більш високорівневі та зручні функції для створення статистичних графіків та візуального аналізу даних, пропонує простий та інтуїтивно зрозумілий інтерфейс для створення складних графіків, таких як графіки розподілу, ящики з вусами, точкові графіки, теплові карти та інші.

Seaborn також має вбудовані функції для роботи зі статистичними моделями та проведення аналізу взаємозв'язків між змінними. Вона надає

зручні інструменти для візуалізації регресійних моделей, кореляційних матриць, категоріальних даних та багато іншого.

Також у роботі були використані інші модулі:

1. numpy - бібліотека для роботи з масивами та виконання математичних операцій над ними.
2. urllib - модуль, який надає інструменти для роботи з мережевими запитами та обробки URL-адрес.
3. statsmodels - бібліотека, яка надає потужні інструменти для статистичного моделювання, включаючи оцінку параметрів, перевірку гіпотез, побудову регресійних моделей, часових рядів.
4. BeautifulSoup - бібліотека, яка полегшує вилучення даних з HTML та XML документів, спрощуючи парсинг та навігацію за структурою документа.
5. datetime - модуль, який надає функціональність для роботи з датами, часом та операціями над ними, дозволяючи виконувати маніпуляції з датами, форматувати та аналізувати часові дані.
6. geopy - модуль, який надає інструменти для геокодування, тобто перетворення адрес у географічні координати і навпаки, що корисно при роботі з місцезнаходженням та геоданими.

3 ЗБІР, ОГЛЯД, ПІДГОТОВКА ТА ОБРОБКА ДАНИХ

3.1 Використані методи програмної обробки

Для формату JSON: читаємо дані з файлу та за допомогою методу json.loads() записуємо у структуру. Цей метод використовується щоб розібрати коректний json рядок (даний у якості аргументу) та конвертувати у структуру Python Dictionary, яку він повертає у результаті свого виконання. З усіх даних, записаних у створеній структурі, вилучаємо тільки необхідні показники. Ініціалізуємо змінну типу List (впорядковані змінювані колекції об’єктів довільних типів), яким надаємо значення - відповідну частину структури.

Для формату CSV: читаємо дані з файлу за допомогою методу pandas.read_csv() записуємо у структуру типу pandas.DataFrame(). Цей метод у бібліотеці Pandas дозволяє читати дані з CSV-файлу (даного у якості аргументу) та повертає двовимірну структуру з позначеними осями - DataFrame, забезпечуючи зручну роботу з табличними даними у Python.

Для формату TXT: читаємо дані з файлу, ініціалізуємо змінну типу List та, обробляючи кожен рядок даних згідно їхнього формату, вилучаємо необхідні показники, записуючи у створений список.

Оптимізація: створення та обробка датафреймів при такій великій кількості даних може займати певний час, що відчутно уповільнить тестування програми в умовах її численних запусків. Щоб уникнути втрат часу, кожен створений датафрейм необхідно записати у файл з розширенням .csv. Після початкового запуску для створення файлів, надалі будемо використовувати метод pandas.read_csv().

Обробка часових даних: в процесі обробки доведеться неодноразово зіткнутися з необхідністю перетворення форматів дат. Для цього застосовано datetime - модуль мови програмування Python, який надає функціональність для роботи з датами, часом та операціями на них, дозволяючи виконувати маніпуляції з датами, форматувати та аналізувати часові дані.

Метод datetime.strftime() використовується для перетворення об'єкта datetime у рядок з певним форматом (даним у якості аргументу). Він дозволяє форматувати дату у вигляді рядка, використовуючи спеціальні символи для подання року, місяця та дня у потрібному порядку. Цей метод корисний для виведення дати у певному форматі або під час створення строкового представлення дати для збереження або відображення.

Метод datetime.strptime() використовується для перетворення дати у форматі рядка (даного у якості першого аргументу) на об'єкт datetime з певним форматом (даного у якості другого аргументу).

Метод datetime.fromtimestamp(timestamp) використовується для створення об'єкта datetime на основі часової мітки (timestamp), яка представляє кількість секунд, що минули з півночі 1 січня 1970 (формат UnixTime). Цей метод дозволяє конвертувати часові мітки у зручний для роботи формат дати та часу у Python.

Метод pandas.DatetimeIndex.normalize() конвертує часовий компонент (яким можна знехтувати при проведенні цього аналізу) дати й часу на північ, тобто 00:00:00. Отже, формат “YYYY:MM:DD HH:MM:SS” конвертується у “YYYY-MM-DD”, що значно полегшить майбутні маніпуляції.

Перетворення даних у вигляді записів на шукані кількісні показники. За допомогою методу pandas.unique(), знайдемо усі унікальні дати у колонці “Date” для створених датафреймів. Власність pandas.DataFrame.loc[] дає доступ до групи заданих у ключі рядків та

колонок. Розмір цієї групи по ключу унікальної дати і буде шуканим кількісним показником.

Графічний аналіз: Для візуального огляду та графічного аналізу доречно буде зобразити оброблені дані у вигляді графіків та/або стовпчастих (стовпчик відповідатиме одному дню) чи кругових діаграм (сектор відповідатиме одному варіанту значення якісного показника), в залежності від формату. Для цього було використано наступні методи:

1. matplotlib.pyplot.subplots() - створює простір для графіків;
2. matplotlib.ticker.MultipleLocator() - встановлює позначку біля кожного цілого числа, кратного аргументу в інтервалі перегляду;
3. matplotlib.axis.xaxis.set_major_locator() - встановляє локатор головного тікера;
4. matplotlib.pyplot.xticks() - отримує або встановлює поточні місця позначок і мітки на осі x;
5. matplotlib.pyplot.yticks() - отримує або встановлює поточні місця позначок і мітки на осі y;
6. matplotlib.pyplot.title() - встановлює назву для осей графіку;
7. matplotlib.pyplot.show() - відображає графік;
8. Та інші.

Для того, щоб додати на деякі графіки лінію тренду використано методи numpy.polyfit() та numpy.poly1d(). Метод numpy.polyfit() у бібліотеці NumPy використовується для підгонки поліноміальної регресії до набору даних, повертаючи коефіцієнти полінома. Метод numpy.poly1d() створює об'єкт полінома на основі заданих коефіцієнтів, дозволяючи виконувати операції з поліномами, такі як обчислення значення полінома у певній точці або виконувати арифметичні операції з поліномами.

Аналіз часових рядів. Для побудови функції автокореляції для часового ряду в Python, використано функцію statsmodels.plot_acf(). Метод statsmodels.plot_acf() з модуля tsaplots бібліотеки statsmodels

використовується для побудови автокореляційної функції часового ряду. Він дозволяє візуалізувати залежність між значеннями часового ряду та його лагами, що допомагає в аналізі сезонності та визначенні оптимального параметра моделі часового ряду.

Робота з посиланнями. Для відкриття URL-адреси використано функцію urllib.request.urlopen() в модулі urllib.request. Вона виконує запит до вказаної в аргументі URL-адреси та повертає відповідь, отриману від сервера - об'єкт HTTPResponse. Цей об'єкт можна використовувати для читання даних, отриманих від сервера. Наприклад, можна викликати метод read(), щоб отримати вміст сторінки у вигляді байтового рядка, або метод decode(), щоб декодувати байтовий рядок у текстовий формат.

Функція quote у бібліотеці urllib використовується для кодування рядка з метою включення її до URL-адреси. При передачі рядка, що містить спеціальні символи, такі як пробіли, символи пунктуації або кирилицю, в URL-адресі, потрібно правильно закодувати ці символи. Функція quote виконує це завдання, замінюючи спеціальні символи на їхні відсоткові коди, які можуть бути безпечно використані URL.

BeautifulSoup() - це виклик конструктора класу BeautifulSoup із бібліотеки BeautifulSoup. Він приймає HTML-код як аргумент і створює об'єкт BeautifulSoup, який представляє розібраний HTML-документ.

Після створення об'єкта BeautifulSoup можна використовувати його для вилучення даних з HTML-структури, пошуку певних тегів, вилучення тексту, атрибутів та інших елементів сторінки. BeautifulSoup полегшує процес парсингу та обробки HTML-коду, надаючи зручні методи для навігації та вилучення потрібних даних.

3.2 Google

Для отримання доступу до архівів даних сервісів Google необхідно перейти на веб-сторінку Google Архіватору [15].

При створенні експорту можна вибрати зі списку продуктів Google (Google Calendar, Gmail, Google Chrome, Google Classroom, YouTube та багато інших), що використовувалися на акаунті, усі необхідні для дослідження набори даних.

Для деяких наборів даних наявний вибір серед декількох форматів експорту, але загалом результатом експорту даних кожного продукту є набір із багатьох файлів різних форматів. Наприклад, дані сервісу Gmail будуть експортовані у два файли: архів електронних листів з розширенням .mbox та файл налаштувань користувача з розширенням .json.

Після вибору всіх підлеглих експорту даних та їх формату, треба обрати спосіб отримання архіву (за посиланням / додати на Google Drive / додати на Dropbox / та інші), частоту (одноразовий експорт / регулярний експорт кожні 2 місяці протягом року), тип (ZIP / TGZ) та максимальний розмір (від 1 до 50 гігабайт).

Після залишення запиту процес експорту може тривати кілька годин або навіть днів. Коли все буде готове, на пошту користувача надійде електронний лист про завершення.

3.2.1 Chronology Google Maps

Результатом експорту даних історії місцезнаходжень будуть наступні файли:

1. Маркери видалення даних у форматі CSV.
2. Налаштування користувача у форматі JSON.

3. Семантичні дані із історії місцезнаходжень у форматі JSON.
4. Записи місцезнаходжень у форматі JSON, розбиті на папки за роками та місяцями.

Для подальшої обробки було вивантажено файли записів місцезнаходжень за місяці, що належать досліджуваному проміжку. Кожен із них має близько десяти тисяч рядків та містить інформацію про сегменти активності переміщення користувача (локація початку та закінчення, тривалість, дистанцію, ймовірний метод переміщення та ступінь впевненості у цьому передбаченні, вірогідність кожного методу переміщення, множина дорожніх точок переміщення, спрощений необроблений шлях) та відвідані місця (локація, тривалість, впевненість у правильності запису, інші вірогідні варіанти і т.д.).

Програмна обробка включала наступні кроки:

1. Прочитати файл записів місцезнаходжень за місяць, декодувати зміст за допомогою методу json.loads() та зберегти у структуру Python, з якої вилучити необхідні записи та зберегти у List.
2. Для кожного дню на досліджуваному проміжку знайти хоч один запис про відвідане у цей день місце. З запису вилучити широту та довготу локації та методом geolocator.reverse() виконати геокодування, отримуючи адресу за координатами. З отриманої інформації вилучити назву міста. В рідкісних випадках, коли за день немає жодного запису про відвідане місце, вважатимемо, що місто перебування користувача не змінилось з минулого дня.
3. Із отриманої інформації створити структуру типу pandas.DataFrame з колонками “Date” та “City”.
4. Повторити аналогічну обробку для кожного місяця, прочитати створені файли, записати у дата фрейми, об’єднати та зберегти у результуючий файл формату CSV, що містить записи про кожен день на досліджуваному проміжку - дату та місто знаходження.

За допомогою методу pandas.groupby() дані в датафреймі групуються по стовпцю 'City'. Потім метод sum() застосовується до кожної групи, підсумовуючи значення інших числових стовпців у кожній групі. Методами бібліотеки matplotlib, створено графіки та діаграми для візуального огляду оброблених даних:

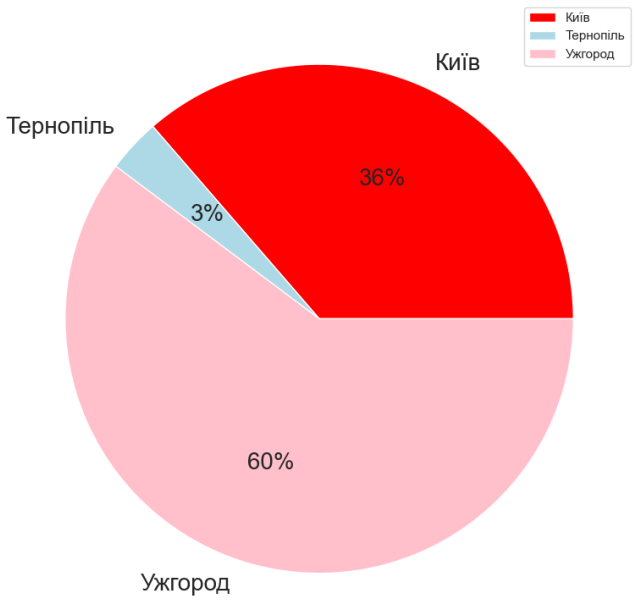

Рисунок 3.2.1.1 - Кругова діаграма локації знаходження користувача

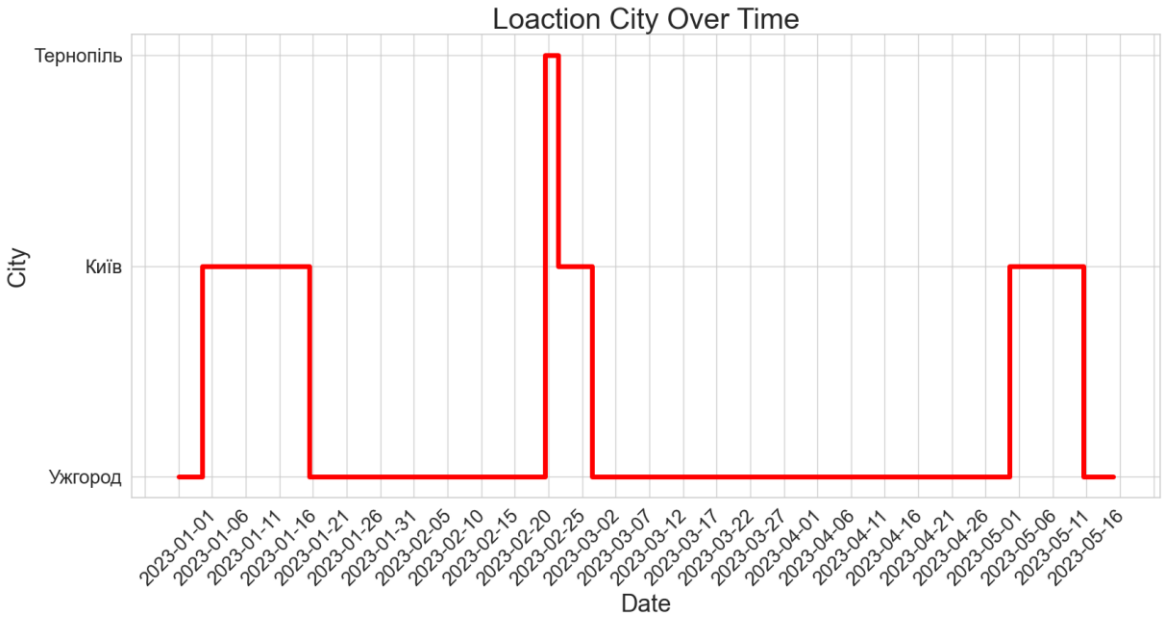

Рисунок 3.2.1.2 - Графік локації знаходження користувача

3.2.2 Google Fit

Результатом експорту даних сервісу Google Fit будуть наступні файли:

1. Записи про тренування та активність, що автоматично відстежується, наприклад біг або поїздки на велосипеді, у форматі TXC.
2. Записи щоденні підсумкові показники активності у форматі CSV.
3. Усі дані Google Fit, згруповані по джерелам, у форматі JSON.
4. Записи про сеанси використання сервісу у форматі JSON.

Для подальшої обробки було вивантажено файл записів про щоденні підсумкові показники активності користувача. Він складається із 610 записів на часовому проміжку з 9 вересня 2021 року по 21 травня 2023 року та містить інформацію про тривалість активності у хвилинах, кількість спалених калорій, подолану дистанцію, кількість "кардіо-очків", кількість "кардіо-хвилин", середній пульс, максимальный пульс, мінімальний пульс, середню швидкість, максимальну швидкість, мінімальну швидкість, кількість зроблених кроків, середню вагу, максимальну вагу, мінімальну вагу та інші показники за день.

Програмна обробка включала наступні кроки:

1. За допомогою методу pandas.read_csv() завантажити дані із файлу в об’єкт Dataframe.
2. Для кожного дню на досліджуваному проміжку знайти запис про підсумкові показники активності цього дня. Вилучити та зберегти у список показники про тривалість активності користувача у хвилинах, кількість спалених калорій, подолану дистанцію та кількість зроблених кроків.
3. Із отриманої інформації створити структуру типу pandas.DataFrame з колонками 'Date', 'Activity Minutes', 'Calories', 'Distance' та 'Steps'.

4. Створений дата фрейм зберегти у результуючий файл формату CSV.

Для знаходження статистичних метрик використаємо методи з програмної бібліотеки Numpy.

Таблиця 3.2.2.1 - Статистичні метрики для даних Google Fit

Метод	Метрика	Activity (хв)	Calories (ккал)	Distance (м)	Steps
numpy.mean()	Середнє арифметичне	58.806	1335.845	2329.842	3921.957
numpy.median()	Медіана	-	1317.948	1811.728	3098.000
numpy.var()	Дисперсія	2321.624	6517.022	4673122.918	12081718.555
numpy.std()	Стандартне відхилення	48.183	80.728	2161.741	3475.877

За допомогою методів бібліотеки matplotlib створено графіки та діаграми для візуального огляду оброблених даних:

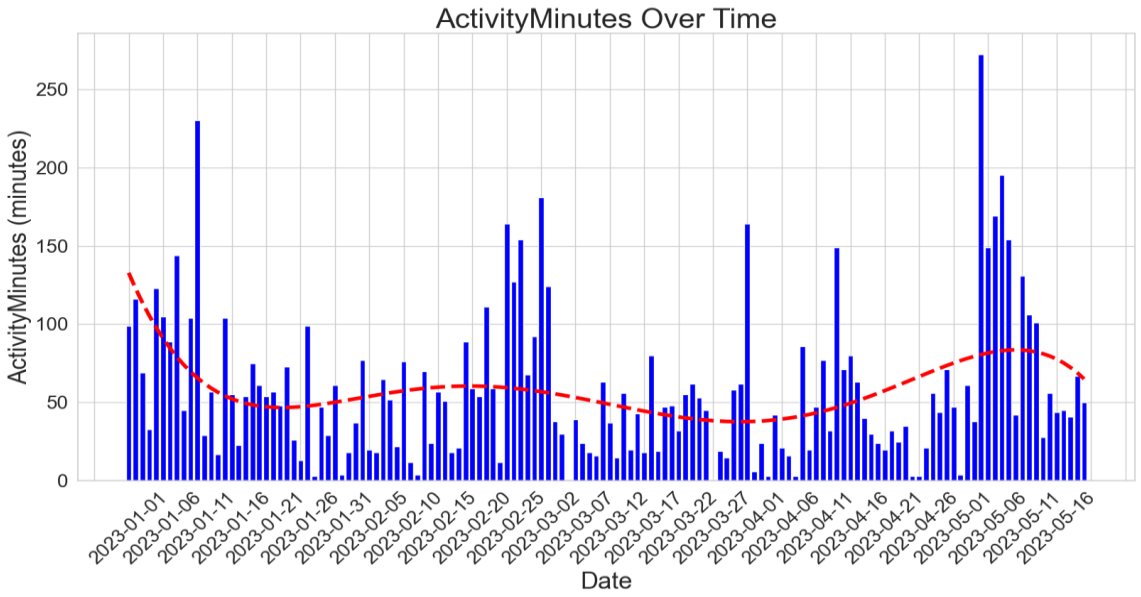

Рисунок 3.2.2.1 - Стовпчаста діаграма тривалості активності

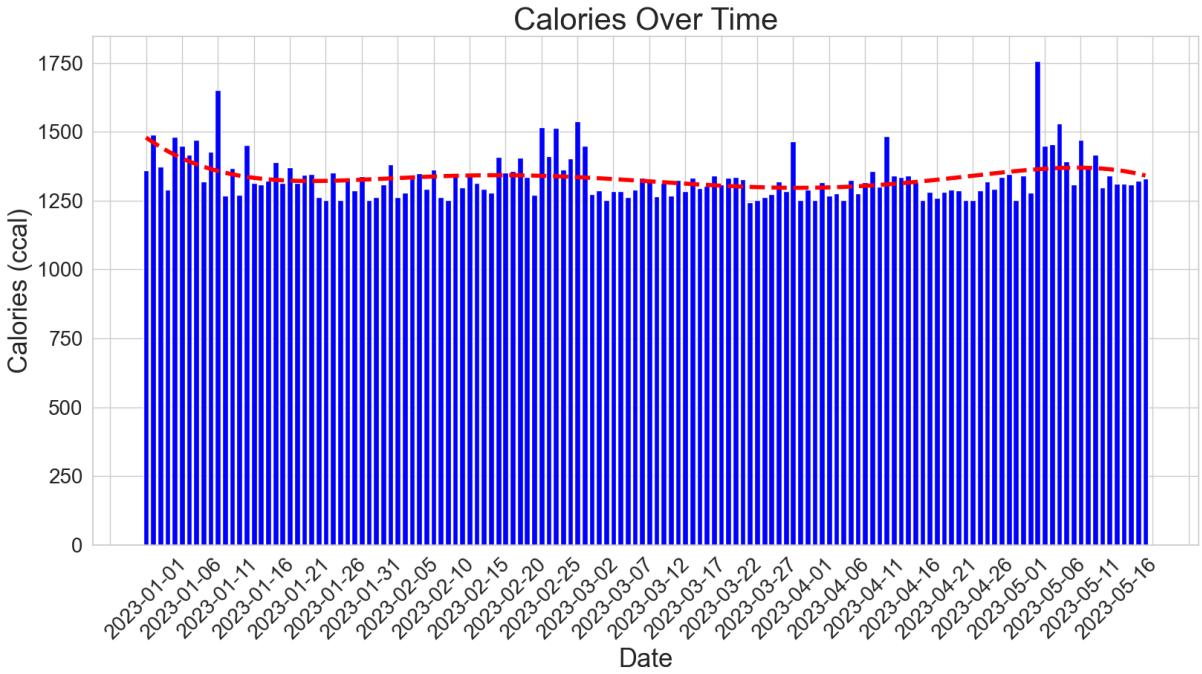

Рисунок 3.2.2.2 - Стовпчаста діаграма кількості спалених калорій

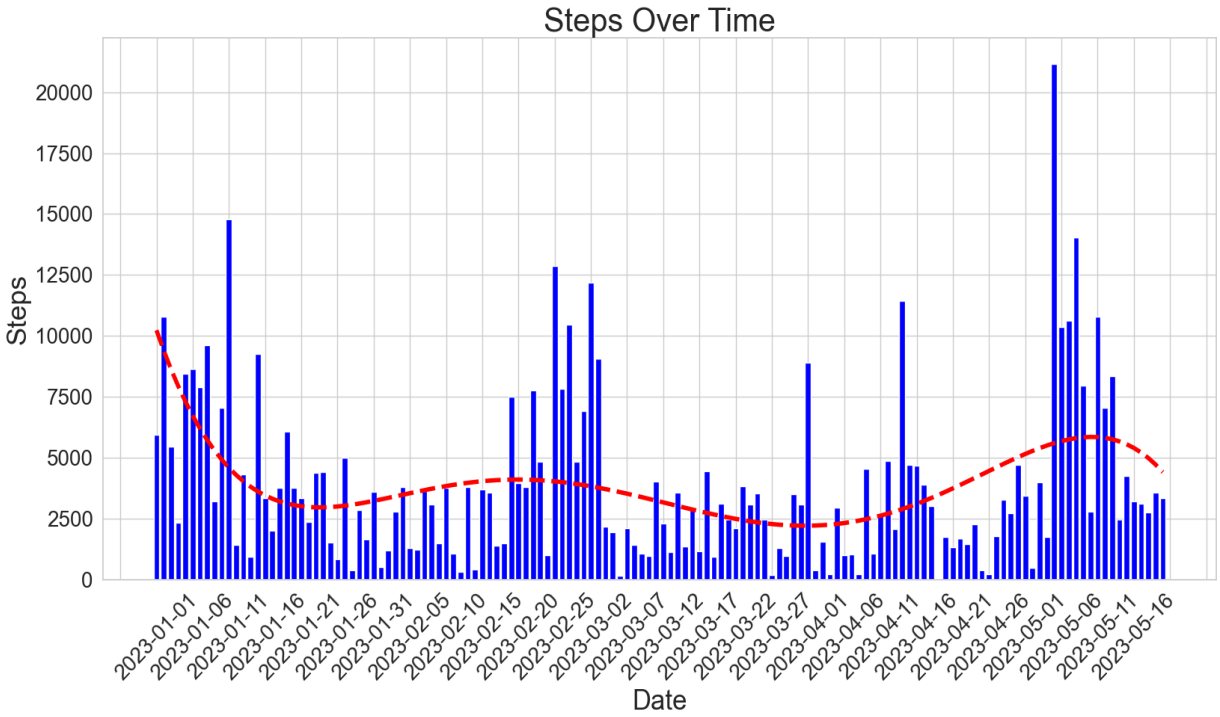

Рисунок 3.2.2.3 - Стовпчаста діаграма кількості зроблених кроків

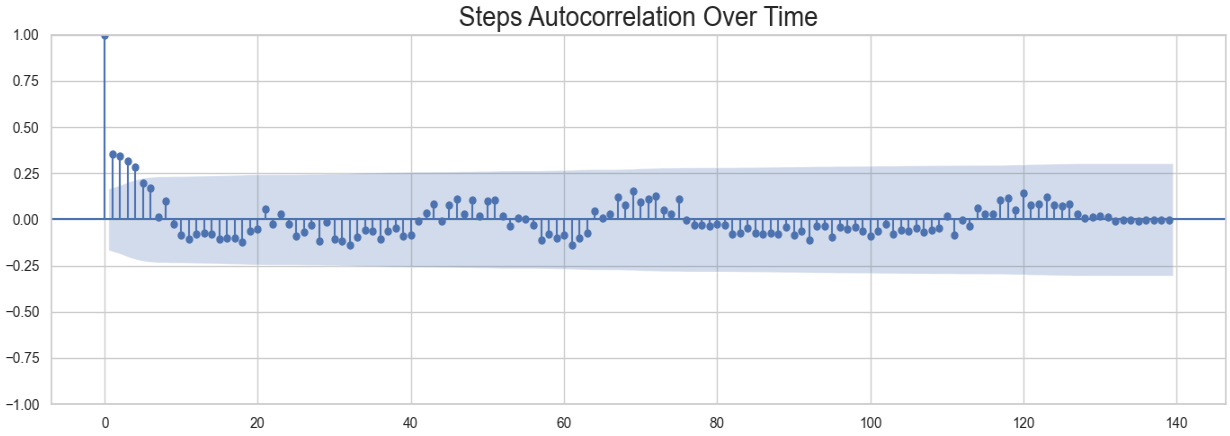

Рисунок 3.2.2.4 - Графік автокорреляцій кількості кроків

3.2.3 Google Chrome

Результатом експорту даних сервісу Google Chrome будуть наступні файли:

1. Дані автозаповнення, у форматі JSON.
2. Закладки, у форматі HTML.

3. Історія браузеру, у форматі JSON.
4. Словник, у форматі CSV.
5. Розширення, у форматі JSON.
6. Пошукові системи, у форматі JSON.
7. Налаштування синхронізації, у форматі JSON.

Для подальшої обробки було завантажено файл історії браузеру користувача. Він складається із близько 30 тисяч записів на часовому проміжку з 21 травня 2022 року по 21 травня 2023 року (1 рік) та містить записи про кожну відвідану інтернет-сторінку (спосіб переходу на сторінку, назву сторінки, посилання, дату).

Програмна обробка включала наступні кроки:

1. Прочитати файл історії браузеру, декодувати зміст за допомогою методу json.loads та зберегти у структуру Python, з якої вилучити необхідні записи та зберегти у змінну типу List.
2. Створити список дат, вилучаючи часові мітки із кожної відвіданої сторінки и перетворюючи їх на об’єкт datetime.
3. Для кожної дати на досліджуваному часовому проміжку знаходимо кількість входжень в створений список.
4. Із отриманої інформації створити структуру типу pandas.DataFrame з колонками "Date" та "PageCount".
5. Зберегти датафрейм у результуючий файл формату CSV, що містить записи про кожен день на досліджуваному проміжку - дату та кількість сторінок, відвіданих користувачем у цей день.

Для знаходження статистичних метрик використаємо методи з програмної бібліотеки Numpy.

Таблиця 3.2.3.1 - Статистичні метрики для даних Google Chrome

Метод	Метрика	PageCount
numpy.mean()	Середнє арифметичне	49.936
numpy.median()	Медіана	32.500
numpy.unique() + numpy.argwhere()	Мода	0 у кількості 15
numpy.var()	Дисперсія	3409.832
numpy.std()	Стандартне відхилення	58.394

За допомогою методів бібліотеки matplotlib створено графіки та діаграми для візуального огляду оброблених даних:

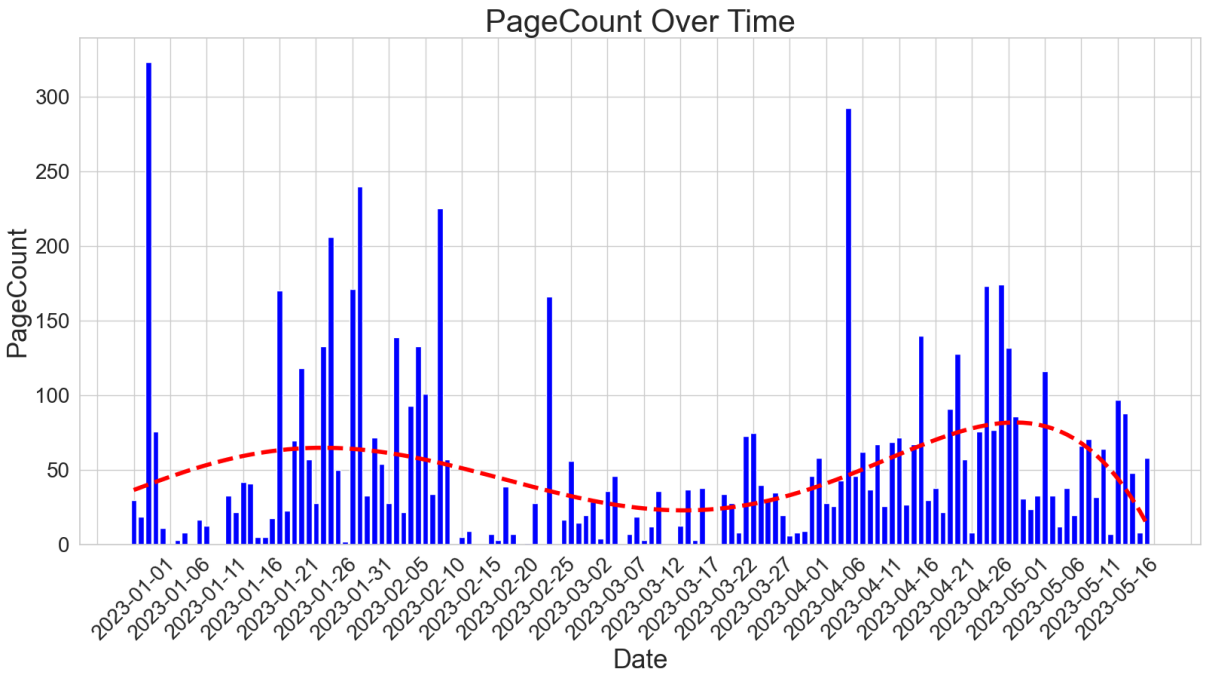

Рисунок 3.2.3.1 - Стовпчаста діаграма кількості відвіданих сторінок

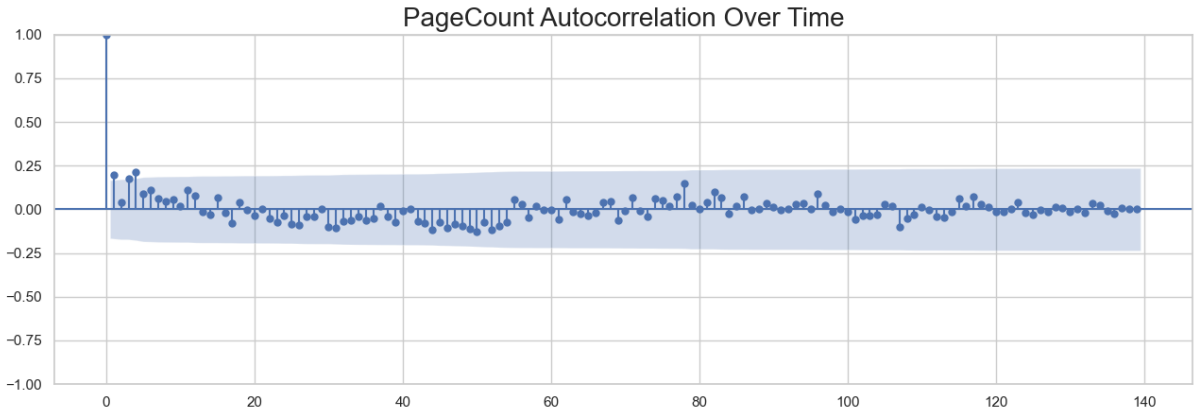

Рисунок 3.2.3.2 - Графік автокорреляцій кількості відвіданих веб-сторінок

3.3 Дані про погоду у містах знаходження користувача

Для доступу до даних про погоду в певному місті на певному часовому проміжку зазвичай використовується один із наступних варіантів:

1. Використання публічних API, таких як OpenWeatherMap, Weather Underground та інші, які надають доступ до даних про погоду через HTTP-запити. Можна отримати ключ API і використовувати його для отримання даних про погоду для конкретного міста на вказаний часовий проміжок.
2. Використання аналізу даних із джерел із відкритим доступом. Деякі організації та університети надають публічні набори даних про погоду, які можна завантажити та використати для досліджень. Наприклад, NOAA (Національне управління океанічних і атмосферних досліджень) надає безліч відкритих даних про погоду.

Однак, доступ як до API, так і до архівів даних про погоду в містах перебування користувача, визначених у розділі 3.2.1, виявився виключно закритим або платним. Внаслідок чого, були вирішено використати більш нестандартний підхід.

Скрапінг веб-сайтів [16] - це процес автоматизованого вилучення даних із веб-сторінок. За допомогою програмного коду, можна отримати

доступ до HTML-коду веб-сторінки, отримати потрібні дані та зберегти їх для подальшого аналізу або використання. Для скрапінгу веб-сайтів можна використовувати спеціальні бібліотеки мови програмування Python, такі як BeautifulSoup, Scrapy, Selenium та інші. Ці інструменти надають зручні функції для парсингу HTML-коду, пошуку та вилучення конкретних елементів сторінки, таких як текст, посилання, зображення та інші дані.

Застосуємо цей метод для вилучення необхідних даних про погоду з веб-сайтів, які надають таку інформацію.

Програмна реалізація методу включала наступні кроки:

1. Скласти URL-адресу із даних про дату, місто та базовий вигляд посилання використовуваного веб-сайту.
2. Відкрити з'єднання зі складеною URL-адресою та записати відповідь сервера у об'єкт файлового потоку. Зчитати HTML-код сторінки в байтовому форматі та декодувати у рядковий формат з кодуванням UTF-8.
3. Для полегшення процесу парсингу сторінки, перетворити HTML-код на об’єкт BeautifulSoup().
4. Вилучити інформацію про шукані погодні показники. Порахувати середнє значення для показників температури, тиску та вологості, що коливаються протягом дня.
5. Сформувати вилучені та уніфіковані погодні дані у список.
6. Повторити запит для кожного дня на досліджуваному часовому проміжку - викликати функцію пошуку погодних даних, вказуючи у аргументах інформацію про місто знаходження користувача у цей день (визначену у розділі 3.2.1).
7. Із отриманої інформації створити структуру типу pandas.DataFrame з колонками "Date", "DayWeek", "SunriseTime", "SunsetTime", "TemperatureMean", "PressureMean", "HumidityMean" та "WeatherStatus".

8. Зберегти датафрейм у результуючий файл формату CSV, що містить записи про кожен день на досліджуваному проміжку - дату та погодні показники у місті передування користувача у цей день.

Для знаходження статистичних метрик використаємо методи з програмної бібліотеки Numpy.

Таблиця 3.3.1 - Статистичні метрики для даних про погоду

Метод	Метрика	Temperature (°C)	Pressure (mm)	Humidity (%)
numpy.mean()	Середнє арифметичне	5.912	751.630	71.872
numpy.median()	Медіана	5.875	751.625	73.188
numpy.var()	Дисперсія	37.941	55.791	237.668
numpy.std()	Стандартне відхилення	6.160	7.469	15.416

За допомогою методів бібліотеки matplotlib створено графіки та діаграми для візуального огляду оброблених даних.

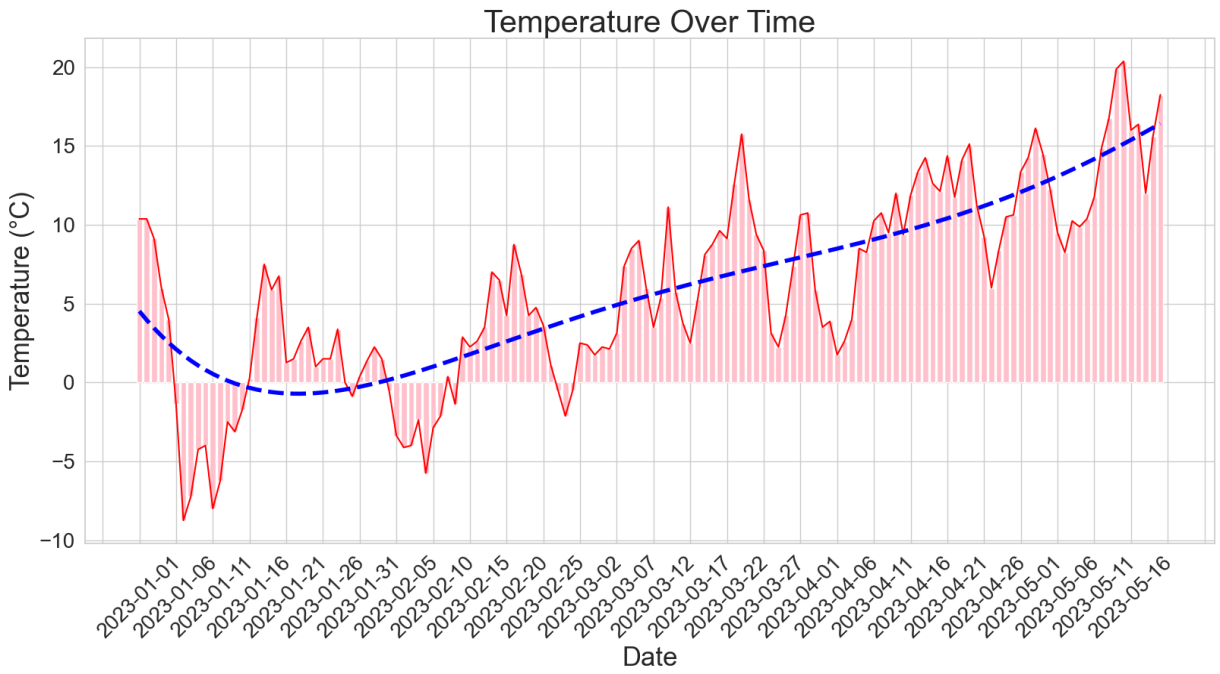

Рисунок 3.2.3.1 - Графік температури

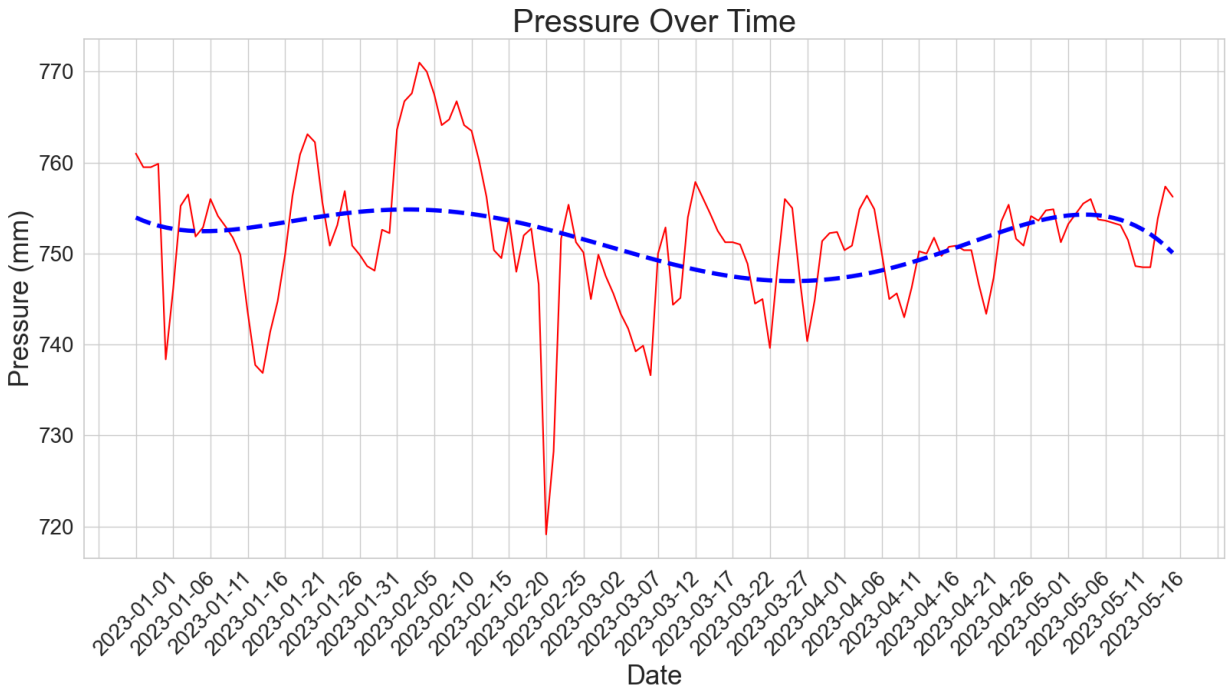

Рисунок 3.2.3.1 - Графік атмосферного тиску

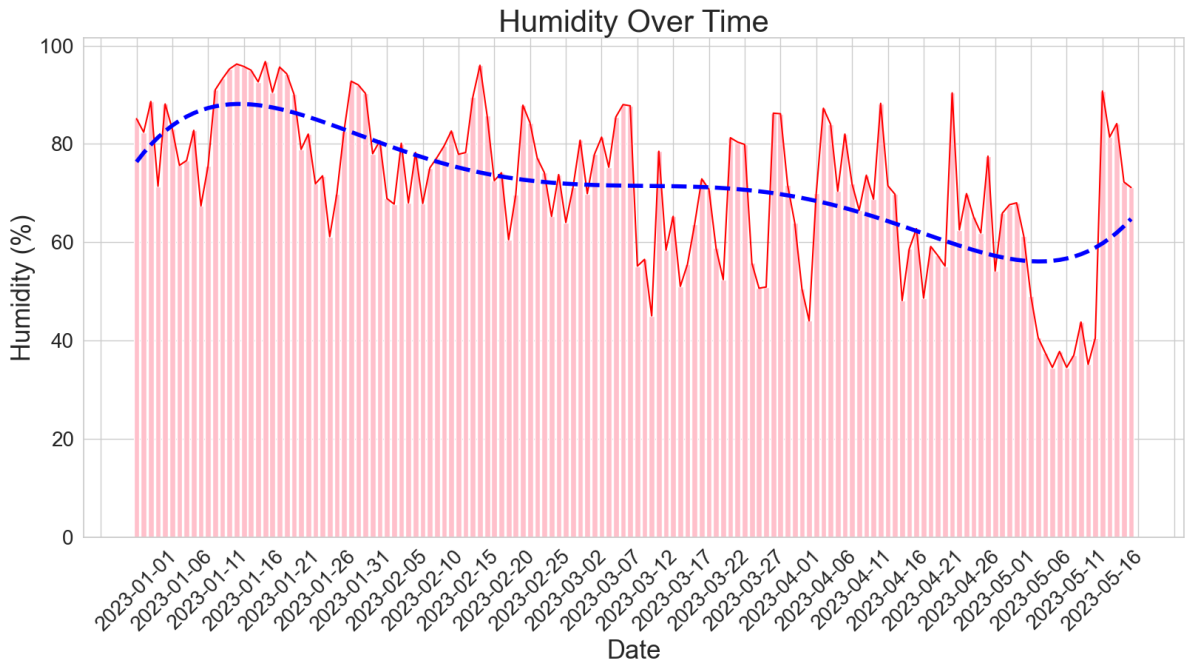

Рисунок 3.2.3.1 - Графік вологості

3.4 Фази місяця

Враховуючи розмір досліджуваного часового проміжку, оптимальним і найшвидшим методом отримати файл, що містить інформацію фази місяця, буде сформувати його вручну, на основі доступної інформації [17].

Для подальшої обробки створено текстовий документ, який містить дати переходу місяця у нову фазу (новий місяць / перша чверть / повний місяць / остання чверть).

Програмна обробка включала наступні кроки:

1. Прочитати текстовий файл і з кожного рядку вилучити інформацію про дату та фазу.
2. Для кожної дати на досліджуваному часовому проміжку визначити поточну фазу та із отриманої інформації створити структуру типу pandas.DataFrame з колонками 'Date' та 'MoonPhase'.

3. Зберегти датафрейм у результуючий файл формату CSV, що містить записи про кожен день на досліджуваному проміжку - дату та фазу місяця у цей день.

Методами бібліотеки matplotlib, створено графік для візуального огляду оброблених даних:

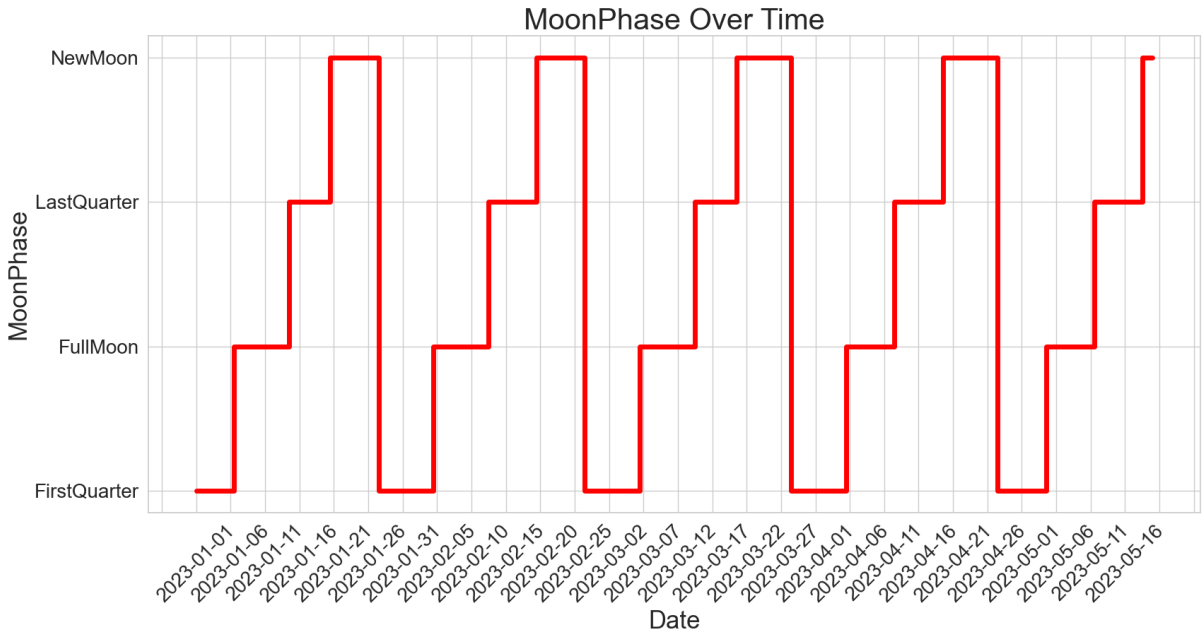

Рисунок 3.4.1 - Графік фаз місяця

3.5 Flo

Для отримання доступу до архівів даних мобільного додатку відстеження менструального циклу Flo [18] необхідно у додатку перейти по наступному шляху: "Меню" (символ у верхньому лівому кутку) – "Допомога" – "Зв'язатися з нами". У відкритому вікні можна залишити запит на експорт даних користувача.

Після залишення і підтвердження запиту процес експорту може тривати до 30 днів. Коли все буде готове, на пошту користувача надійде електронний лист про завершення.

Результатом експорту даних сервісу Flo є безпечний zip-архів, захищений паролем. Цей zip-архів міститиме два окремі файли:

1. Файл з даними аккаунту у форматі TXT.
2. Файл з даними аккаунту у форматі JSON.

Для подальшої обробки було вивантажено другий файл. Він має більше ста тисяч рядків та містить інформацію про особливості та налаштування аккаунту, записи про сесії і дії у додатку, записи про цикли, симптоми та примітки.

Програмна обробка включала наступні кроки:

1. Прочитати файл даних аккаунту, декодувати зміст за допомогою методу json.loads та зберегти у структуру Python, з якої вилучити необхідні записи та зберегти у змінну типу List.
2. Для кожної дати на досліджуваному часовому проміжку знайти кількість записів про дії користувача у додатку, а також визначити поточну фазу менструального циклу.
3. Також для кожного дню визначити його порядковий номер у поточному циклі, а також його нормалізоване значення від 0 до 1 відносно тривалості поточного циклу .
4. Із отриманої інформації створити структуру типу pandas.DataFrame з колонками "Date", "FloPhase", "FloDay", "FloPercentage" та "ActivityCount".
5. Зберегти датафрейм у результуючий файл формату CSV, що містить записи про кожен день на досліджуваному проміжку.

Для знаходження статистичних метрик використаємо методи з програмної бібліотеки Numpy.

Таблиця 3.5.1 - Статистичні метрики для даних Flo

Метод	Метрика	ActivityCount
numpy.mean()	Середнє арифметичне	7.336
numpy.median()	Медіана	0
numpy.unique() + numpy.argwhere()	Мода	0 у кількості 119
numpy.var()	Дисперсія	822.866
numpy.std()	Стандартне відхилення	28.686

За допомогою методів бібліотеки matplotlib створено графіки та діаграми для візуального огляду оброблених даних:

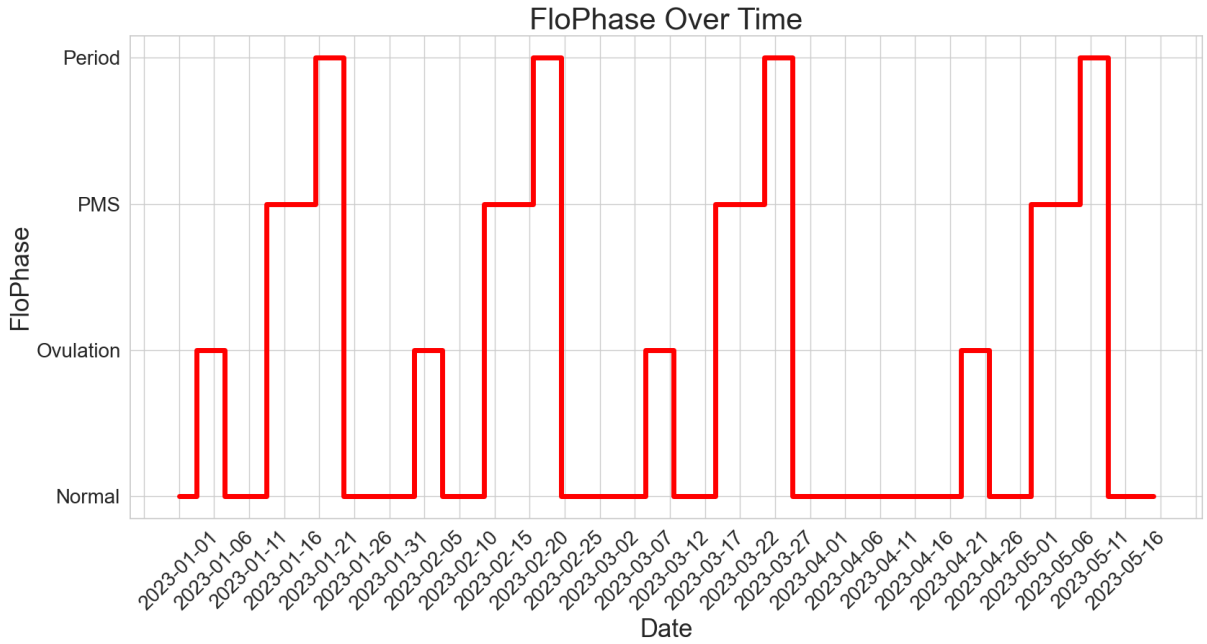

Рисунок 3.5.1 - Графік фаз менструального циклу користувача

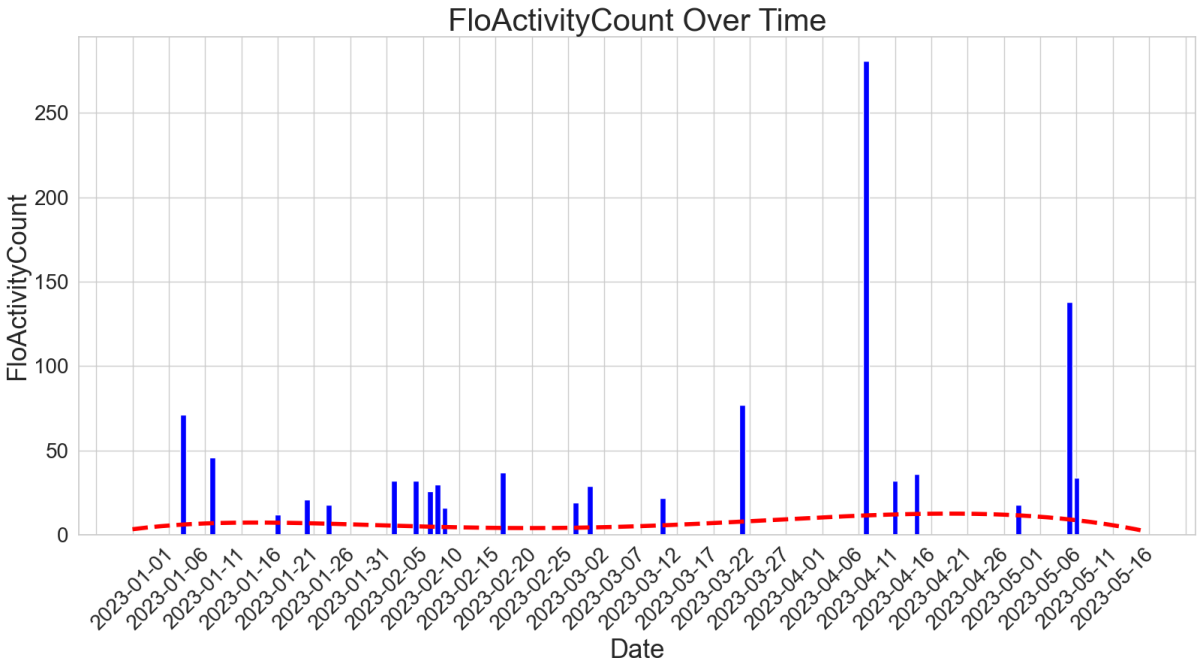

Рисунок 3.5.2 - Стовпчаста діаграма активності користувача у додатку Flo

3.6 Tiktok

У додатку TikTok [19] необхідно перейти по наступному шляху: вкладка "Профіль" – символ меню у правому верхньому кутку – "Налаштування та конфіденційність" – "Аккаунт" – "Завантаження ваших даних". Далі обрати формат файла JSON (також доступний альтернативний формат TXT) та натиснути "Запитати дані". Обробка запиту може тривати декілька днів, а після її закінчення, про яке буде сигналізувати відповідне сповіщення, відведено 4 дні, щоб завантажити файл у вигляді zip-архіву. Якщо відправити новий запит, попередній файл стане недоступний.

Результатом експорту даних сервісу TikTok буде один файл з даними аккаунту у форматі JSON.

Для подальшої обробки було вивантажено експортований файл. Він має більше трьохсот тисяч рядків та містить інформацію про "улюблені" ефекти, хештеги, звуки та відео (дата та посилання), список підписників та підписок (дата та нік користувача), список використаних хештегів (назва,

дата), список вподобаних відео (дата, посилання), список переглянутих відео за останні півроку (дата, посилання), історію сесій в додатку (дата, ip-адреса, модель пристрою, система пристрою, тип мережі, оператор мережі), історію транзакцій у додатку, історію пошуку, історію поширень (дата, тип поширюваного контенту, посилання, метод), налаштування, історію коментарів, історію повідомлень і т.д .

Програмна обробка включала наступні кроки:

1. Прочитати файл даних аккаунту TikTok, декодувати зміст за допомогою методу json.loads та зберегти у структуру Python, з якої вилучити списки записів про вподобані та переглянуті відео та зберегти у змінні типу List.
2. Для кожного дню на досліджуваному часовму проміжку за допомогою власності pandas.DataFrame.loc[] отримати доступ до групи рядків та колонок за заданою у ключі датою. Розмір цієї групи у списках по ключу унікальної дати і буде шуканою кількістю вподобаних/переглянутих відео.
3. Із отриманої інформації створити структуру типу pandas.DataFrame з колонками "Date", "LikedCount" та "HistoryCount".
4. Зберегти датафрейм у результуючий файл формату CSV, що містить записи про кожен день на досліджуваному проміжку - дату та кількість вподобаних і переглянутих користувачем відео цього дня.

Для знаходження статистичних метрик використаємо методи з програмної бібліотеки Numpy.

Таблиця 3.6.1 - Статистичні метрики для даних TikTok

Метод	Метрика	LikedCount	HistoryCount
numpy.mean()	Середнє арифметичне	7.500	457.364
numpy.median()	Медіана	5.000	413.000
numpy.unique() + numpy.argwhere()	Мода	3 у кількості 20	292 у кількості 4
numpy.var()	Дисперсія	43.164	69885.417
numpy.std()	Стандартне відхилення	6.570	264.359

За допомогою методів бібліотеки matplotlib створено графіки та діаграми для візуального огляду оброблених даних:

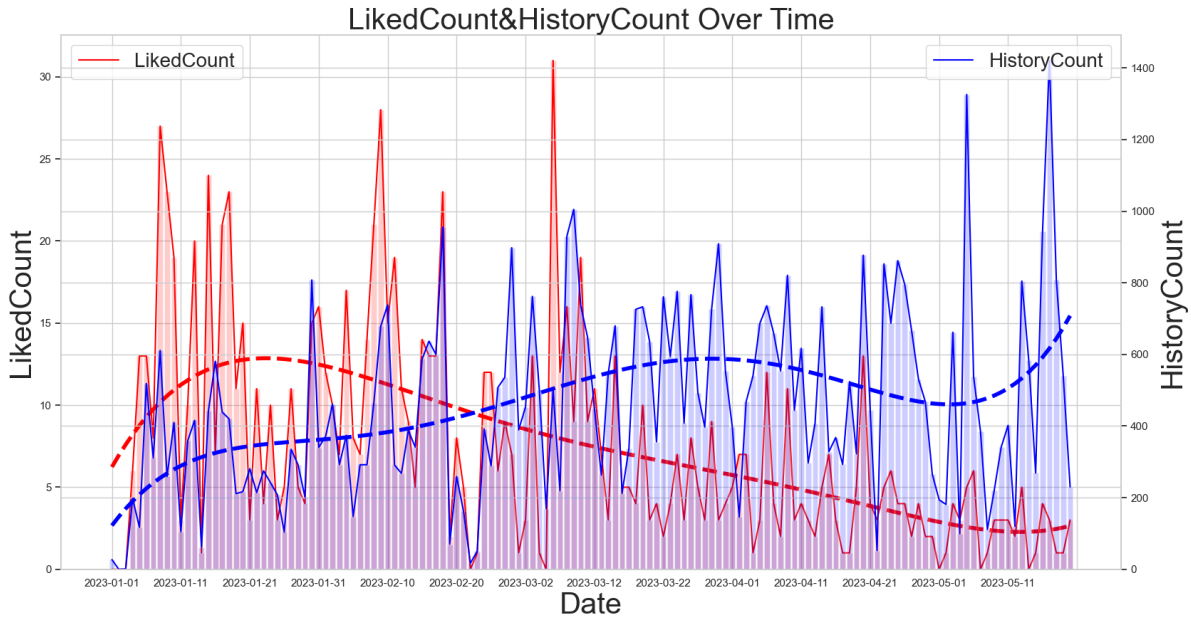

Рисунок 3.6.1 - Графіки вподобаних та переглянутих відео

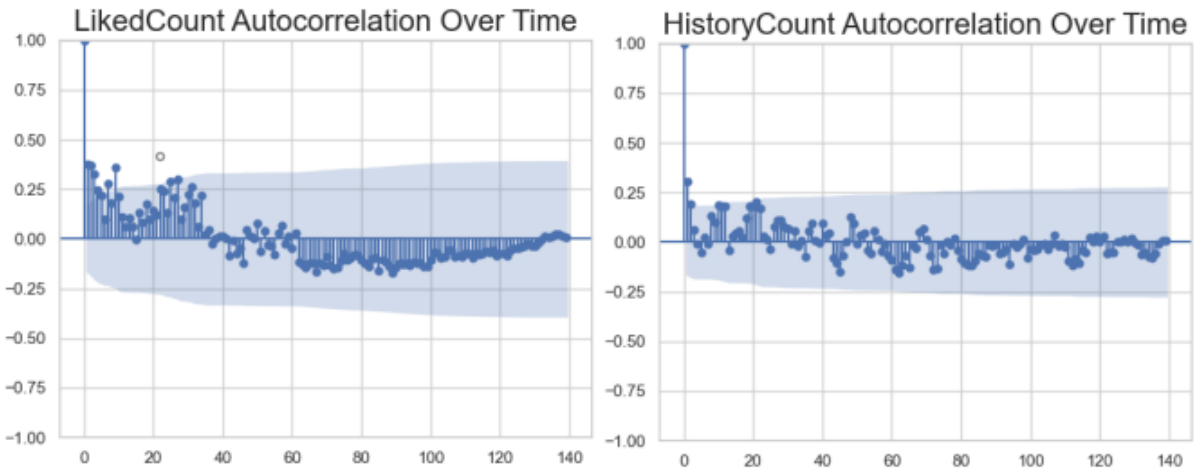

Рисунок 3.2.2.5 - Графіки автокорреляцій вподобаних та переглянутих відео

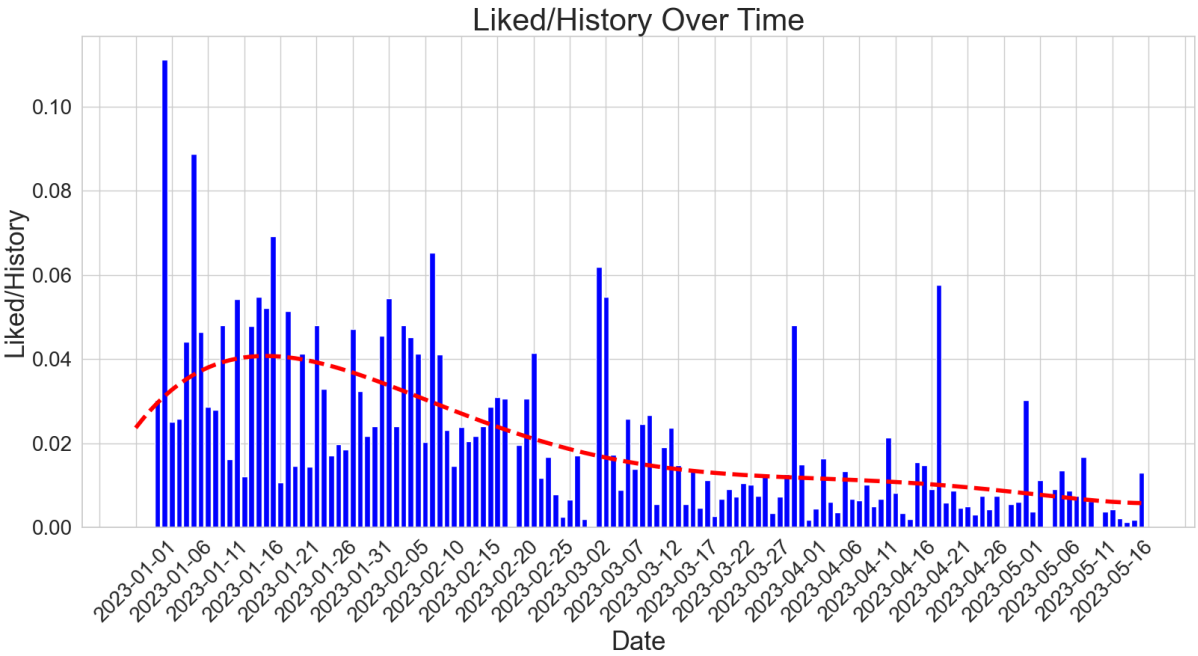

Рисунок 3.2.2.5 - Стовпчаста діаграма співвідношення кількості вподобаних та переглянутих відео

3.7 Telegram

У додатку Telegram [20] необхідно перейти по наступному шляху: Меню (символ у лівому верхньому кутку) – "Налаштування" – "Просунуті

налаштування" – "Експорт даних Telegram". При створенні експорту можна вибрати зі списку даних (інформація про акаунт, список контактів, персональні чати, публічні канали, фото, відео, голосові повідомлення, активні сесії, тощо), що стосуються акаунту, усі необхідні для дослідження набори даних. Доступні формати: HTML для людського і JSON для машинного читання. Обробка запиту може тривати довгий час, в залежності від обсягу експортованих даних, а після її закінчення, файл буде знаходитись за вказаним, у процесі налаштування експорту, шляхом.

Результатом експорту даних сервісу Telegram буде папка з медіа-файлами та один файл з даними аккаунту у форматі JSON.

Для подальшої обробки було вивантажено експортований файл. Він має більше восьми мільйонів рядків та містить інформацію про всі чати (тип, індекс) та повідомлення (індекс, тип, дата, дата у форматі часової мітки, ім’я відправника, id відправника, вкладені файли, тип змісту, текст повідомлення, тощо) користувача.

Програмна обробка включала наступні кроки:

1. Прочитати файл даних аккаунту Telegram, декодувати зміст за допомогою методу json.loads та зберегти у структуру Python, з якої вилучити список чатів у змінну типу List.
2. Для кожного повідомлення у кожному чаті вилучити інформацію про дату відправки та ім’я відправника. Кожне повідомлення, що належить досліджуваному часовому проміжку зберегти у відповідний список (отримані/відправлені повідомлення).
3. Для кожного дню на досліджуваному часовому проміжку підрахувати кількість отриманих та відправлених повідомлень.
4. Із отриманої інформації створити структуру типу pandas.DataFrame з колонками "Date", "SendCount" та "ReceivedCount".
5. Зберегти датафрейм у результуючий файл формату CSV, що містить записи про кожен день на досліджуваному проміжку.

Для знаходження статистичних метрик використаємо методи з програмної бібліотеки Numpy.

Таблиця 3.7.1 - Статистичні метрики для даних Telegram

Метод	Метрика	SendCount	ReceivedCount
numpy.mean()	Середнє арифметичне	50.821	72.636
numpy.median()	Медіана	37.500	53.000
numpy.unique() + numpy.argwhere()	Мода	6, 8, 12, 33, 38, 46 у кількості 4	21 у кількості 5
numpy.var()	Дисперсія	2302.975	4955.803
numpy.std()	Стандартне відхилення	47.989	70.397

За допомогою методів бібліотеки matplotlib створено графіки та діаграми для візуального огляду оброблених даних:

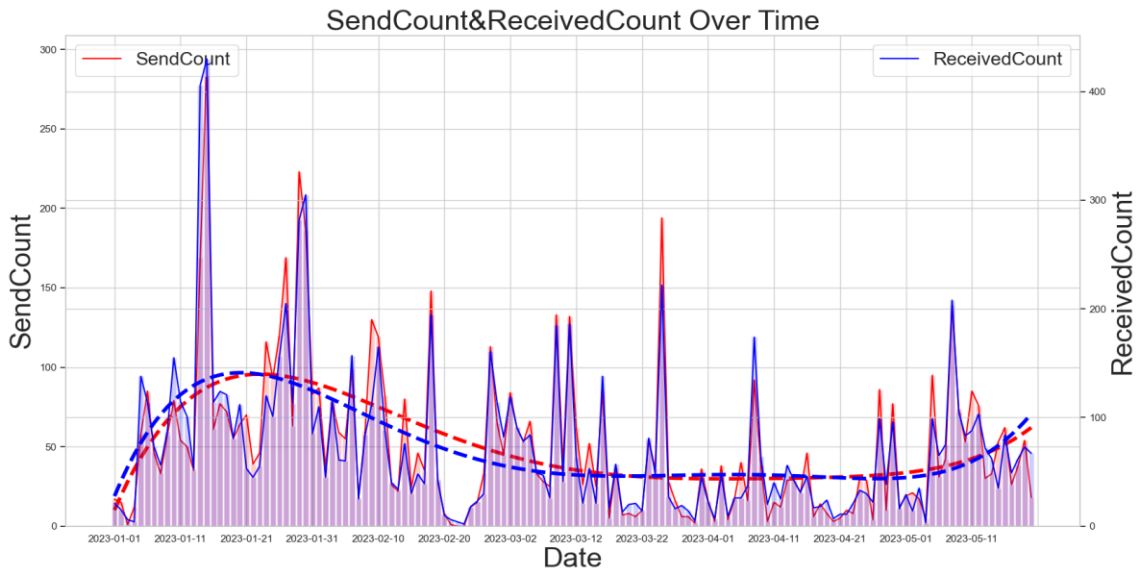

Рисунок 3.7.1 - Графіки кількості надісланих та отриманих повідомлень

3.8 Discord

Для того, щоб отримати копію особистих даних, наданим сервісу Discord [21] за час використання облікового запису, необхідно перейти по наступному шляху: Налаштування Користувача – Конфіденційність – "Запросити всі мої дані". Ця кнопка є у всіх версіях Discord (настільний додаток, браузерна та мобільна версія).

Після залишення запиту процес експорту може тривати кілька годин або навіть днів. Коли все буде готове, на пошту акаунту надійде електронний лист про завершення.

Результатом експорту буде пакет даних, у якому знаходяться декілька файлів у форматах JSON та CSV, загальною вагою близько 400 мегабайт. Вони містять дані про обліковий запис (список друзів, налаштування, ім'я користувача, адресу електронної пошти, аватар та історія всіх операцій по Nitro), сервери, повідомлення, відомості про використання програми, тощо.

Однак, шуканий показник серед даних із цього джерела – сумарна тривалість дзвінків, експортовані файли погано відформатовані та перевантажені зайвою інформацією. Оптимальним і найшвидшим методом отримати файл, що містить необхідні дані, буде сформувати його вручну, на основі доступної у додатку інформації.

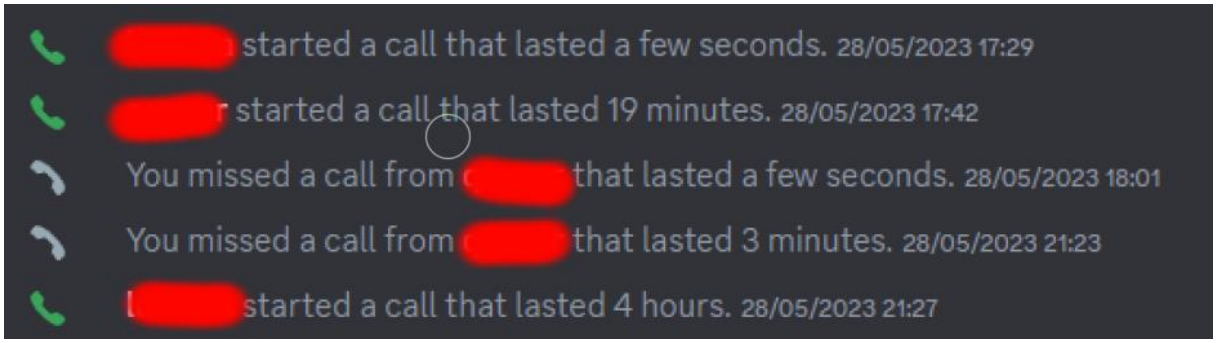

Рисунок 3.8.1 - Шукані дані на знімку екрану у Discord

Програмна обробка включала наступні кроки:

1. Прочитати текстовий файл і з кожного рядку вилучити інформацію про дату та тривалість дзвінку.
2. Для кожної дати на досліджуваному часовому проміжку визначити сумарну тривалість дзвінків за день у хвилинах та із отриманої інформації створити структуру типу pandas.DataFrame з колонками "Date" та "CallsDuration".
3. Зберегти датафрейм у результуючий файл формату CSV, що містить записи про кожен день на досліджуваному проміжку.

Для знаходження статистичних метрик використаємо методи з програмної бібліотеки Numpy.

Таблиця 3.8.1 - Статистичні метрики для даних Discord

Метод	Метрика	CallsDuration (хв)
numpy.mean()	Середнє арифметичне	193.493
numpy.median()	Медіана	180.000
numpy.unique() + numpy.argwhere()	Мода	0 у кількості 51
numpy.var()	Дисперсія	36869.264
numpy.std()	Стандартне відхилення	192.014

Методами бібліотеки matplotlib, створено графік для візуального огляду оброблених даних:

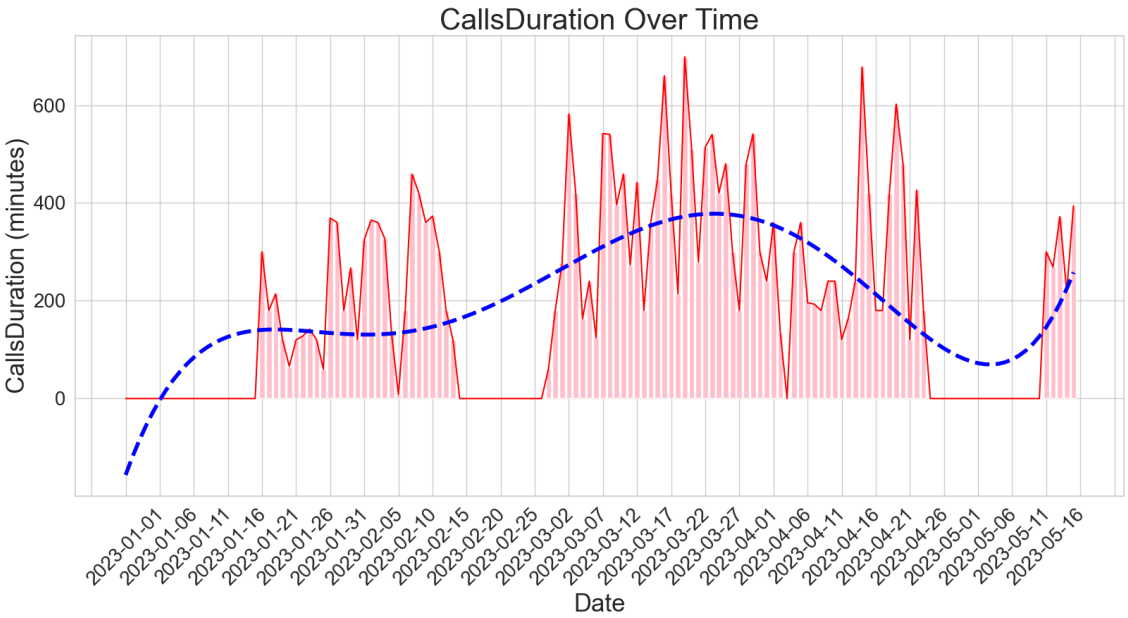

Рисунок 3.8.2 - Графіки тривалості дзвінків Discord

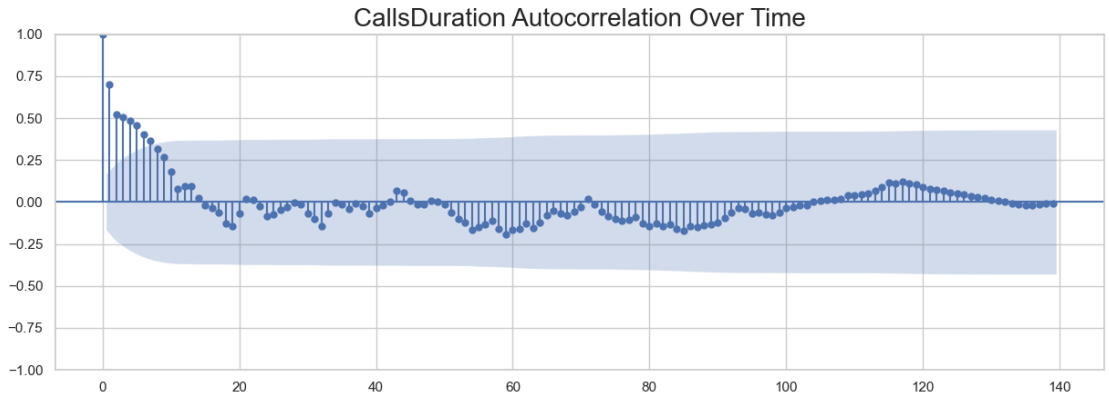

Рисунок 3.2.2.5 - Графік автокорреляцій тривалості дзвінків Discord

3.9 Privat24

Для отримання доступу до архівів даних сервісу Privat24 необхідно на веб-сторінці Privat24 - [22], перейти на вкладку "Гаманець", вибрати досліджувану картку та часовий інтервал (до 90 днів). Після цього натиснути на іконку "Експорт у XLS".

Результатом експорту буде таблиця, що містить інформацію про всі проведені на картці операції (дата, час, категорія, картка, опис, сума у

валюті картки, сума у валюті транзакції, залишок) за обраний інтервал часу. Оскільки досліджуваний часовий період має більше за 90 днів, експорт таблиці необхідно здійснити двічі та об’єднати, після чого видалити зайві стовпчики і завантажити у форматі CSV.

Програмна обробка включала наступні кроки:

1. За допомогою методу pandas.read_csv() завантажити дані із файлу в об’єкт Dataframe.
2. Для кожного дня на досліджуваному проміжку знайти всі записи про здійснені транзакції. Підрахувати суми транзакцій у якості одержувача та відправника за день.
3. Із отриманої інформації створити структуру типу pandas.DataFrame з колонками 'Date', 'MoneySend' та 'MoneyReceived'.
4. Створений дата фрейм зберегти у результуючий файл формату CSV.

За допомогою методів бібліотеки matplotlib створено графіки та діаграми для візуального огляду оброблених даних:

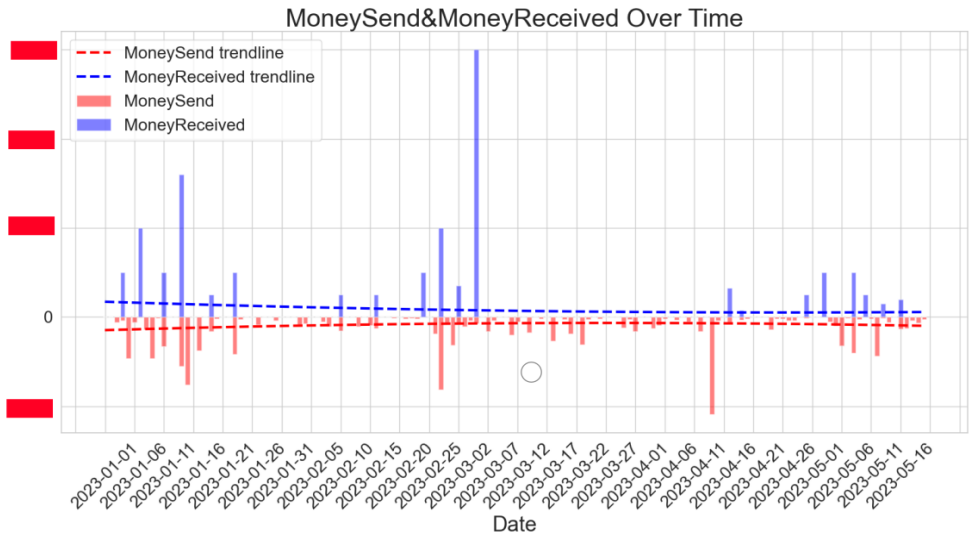

Рисунок 3.9.1 - Стовпчасті діаграми обсягу грошових витрат та надходжень

4 АНАЛІЗ ДАНИХ, РЕЗУЛЬТАТИ РОБОТИ

4.1 Аналіз статистичних метрик

Підраховані статистичні метрики навіть без подальших графічних відображень є дуже цінною інформацією. На їх основі інтуїтивно чи у порівнянні з даними інших користувачів можна зробити певні висновки, хоч і не розраховуючи на високу точність.

Google Fit. За даними аналізованого користувача середня кількість зроблених кроків за день дорівнює близько 4 тысячам, що згідно з [23], класифікується як низький рівень активності і не дотягує навіть до мінімального рекомендованого значення в 7 тисяч. Це свідчить про необхідність збільшення фізичної активності.

Медіана нижча за середнє значення вказує на незначну кількість викидів або незвичайних подій, які вплинули на загальну статистику.

Співвідношення стандартного відхилення до середнього арифметичного кількості кроків ~89% свідчить про кардинальну несистемність та нерегулярність фізичної активності користувача.

Google Chrome. Згідно з дослідженням [24], проведеним GlobalWebIndex у 2019 році, середній користувач Інтернету відвідує близько 40-50 унікальних веб-сторінок на день. Середня кількість відвіданих піддослідним користувачем інтернет-сторінок за день ~50, що належить множині типових значень.

Медіана нижча за середнє значення вказує на незначну кількість викидів або незвичайних подій, які вплинули на загальну статистику.

Співвідношення стандартного відхилення до середнього арифметичного кількості відвіданих сторінок ~116% свідчить про

кардинальну несистемність активності користувача у використанні браузеру.

Tiktok. На аналізованому акаунті середнє число переглянутих відео за день дорівнює близько 500. Навіть без порівнянь з іншими користувачами можна з упевненістю сказати про надмірність використання соціальної мережі. Адже при середній тривалості відео 15 секунд (згідно за даними з TikTok та інших джерел [25], що аналізували платформу), час проведений у додатку становить 125 хвилин на день (середнє значення по світу - 52 хвилини). Навіть не враховуючи інші активності доступні на мобільних девайсах, це значення перевищує дві години, які вважаються прийнятним лімітом для здорового використання телефону.

Медіана нижча за середнє значення вказує на незначну кількість викидів або незвичайних подій, які вплинули на загальну статистику.

Співвідношення середньої кількості вподобань за день до середньої кількості переглянутих відео близько 1.6%. У порівнянні із середнім рівнем залучення рівним 6.72% аналізованого користувача можна вважати більш перебірливим у вподобаннях відео.

Співвідношення стандартного відхилення кількості переглядів до середньої кількості переглядів ~58% свідчить про відносну регулярність використання додатку.

Telegram. Згідно зі статистикою [26], середня кількість текстових повідомлень на день серед людей віком від 18 до 24 років - 128. Середня кількість відправлених піддослідним користувачем повідомлень за день ~50, що значно менше типового значення.

Медіана нижча за середнє значення вказує на незначну кількість викидів або незвичайних подій, які вплинули на загальну статистику.

Співвідношення стандартного відхилення до середнього арифметичного кількості відправлених/отриманих повідомлень ~97% свідчить про несистемність використання Telegram.

Discord. Згідно з [27], середній користувач Discord у США витрачає 4,67 години або 280,6 хвилин на місяць, що становить 9 хвилин щодня. Середня сумарна тривалість дзвінків піддослідного користувача становить 194 хвилини за день, що катастрофічно перевищує типове значення і свідчить про необхідність більшої помірності у використанні сервісу.

Медіана нижча за середнє значення вказує на незначну кількість викидів або незвичайних подій, які вплинули на загальну статистику.

Співвідношення стандартного відхилення до середнього арифметичного кількості відвіданих сторінок ~99% свідчить про кардинальну несистемність та нерегулярність активності у Discord.

4.2 Графічний аналіз

Отримані графіки та діаграми дозволяють не тільки зробити висновки щодо тенденцій та закономірностей активності користувача, дані якого були використані, а також припустити їх причини та наслідки. Більше того, досвід отриманий у процесі дає можливість відкрити широкий діапазон потенційних областей та об’єктів для аналогічних досліджень.
Chronology Google Maps.

Google Fit. На діаграмах можна помітити багато аномальних піків активності користувача, немає поміних трендів до скорочення чи зростання. Розподіл даних нормальний. На автокореляційному графіку можна побачити певну сезонність.

Google Browser. На діаграмі можна помітити багато аномальних піків активності користувача, немає помітних трендів до скорочення чи зростання. Розподіл даних нормальний. Автокореляційний графік показує випадковий розподіл значень близько нуля на всіх лагах, що може вказувати на відсутність сезонності або залежності між поточними та минулими значеннями.

Погода. На діаграмах відсутні аномальні піки, помітний тренд до зростання температури та спадання вологості.

Flo. На діаграмі можна помітити один аномальний пік активності користувача, немає помітних трендів до скорочення чи зростання.

TikTok. На діаграмах можна помітити багато аномальних піків активності, а також загальний тренд до зростання кількості переглянутих відео та скорочення кількості вподобаних відео. Більш яскраво полярність трендів демонструє графік співвідношеннь цих показників з явним трендом на скорочення. Можна запропонувати два можливих пояснення таких результатів: аналізований користувач став більш перебірливим за даний проміжок часу, або алгоритми підбору відео у особистій стрічці змінювалися, погіршуючи передбачення інтересів цього користувача.

Telegram. На діаграмі можна помітити багато аномальних піків активності користувача, наявний помітний тренд до скорочення як відправлених, так і отриманих повідомлень. Розподіл даних нормальний. Автокореляційний графік показує переважну слабку негативну залежність між поточними та минулими значеннями.

Discord. На діаграмі можна помітити багато аномальних періодів відсутності активності користувача, помітний тренд до зростання сумарної тривалості дзвінків. Автокореляційний графік показує переважну слабку негативну залежність між поточними та минулими значеннями.

Privat24.ua. На діаграмі можна помітити багато аномальних піків, наявні незначні тренди до скорочення як надходжень, так і витрат.

4.3 Кореляційний аналіз

Метод pandas.dataframe.corr() використовується для обчислення кореляційної матриці між стовпцями DataFrame. Кореляційна матриця є квадратною таблицею, де кожен елемент на перетині рядка і стовпця

представляє коефіцієнт кореляції між відповідними стовпцями даних - ступінь залежності, що варіюється від -1 до 1.

Метод дозволяє розраховувати коефіцієнти кореляції Пірсона, Кендалла і Спірмена, залежно від значення параметру "method", встановленого під час виклику.

Усі зібрані та попередньо оброблені дані було об’єднано в єдиний датафрейм. Використовуючи метод pandas.dataframe.map(), якісні дані було відображено в новий стовпець у кількісному вигляді. Наприклад, у результуючому стовпцю, що відображає дані про місто знаходження користувача, значенню "Ужгород" відповідатиме "0", "Київ" – "1", "Тернопіль" - "2". Надалі метод необхідно викликати зі значенням параметру numeric_only="True", щоб обчисленням підлягали тільки стовпці з цілим, дійсним або логічним типом даних.

Спочатку для побудови кореляційної матриці було використано коефіцієнт Пірсона. Більшість зібраних даних виміряються дійсними або цілими числами і підлягають аналізу з допомогою цього методу.

Для більш наочного відображення даних використано функцію seaborn.heatmap() із бібліотеки Seaborn. Вона використовується для візуалізації матриці даних у вигляді теплової мапи. В якості аргументів функція приймає двовимірний масив або квадратну матрицю та інші параметри для налаштування створюваного графічного відображення.

	CityId	ActivityMinutes	...	MoneySend	MoneyReceived
CityId	1.000000	0.566750	...	-0.337513	0.138866
ActivityMinutes	0.566750	1.000000	...	-0.389407	0.063419
Calories	0.577176	0.954455	...	-0.402250	0.080936
Distance	0.606563	0.968375	...	-0.411431	0.076180
Steps	0.600400	0.974212	...	-0.406531	0.074914
PageCount	-0.201341	-0.059445	...	0.025837	-0.017265
WeekDayId	0.033315	0.113223	...	0.045506	0.075023
TemperatureMean	-0.186919	-0.018269	...	0.100788	-0.156653
PressureMean	-0.223696	-0.092767	...	0.109491	0.016634
HumidityMean	-0.094175	-0.157114	...	-0.038275	0.045182
MoonPhaseId	0.238067	0.067591	...	-0.148034	-0.010950
FloPhaseId	0.149381	0.081920	...	-0.030157	0.033205
FloActivityCount	0.034020	0.006267	...	0.037006	0.035165
FloDay	0.140062	0.125504	...	-0.068838	0.001831
FloPercentage	0.119785	0.072008	...	-0.065466	0.004807
LikedCount	0.159730	-0.082209	...	-0.065878	-0.140282
HistoryCount	-0.071478	-0.052571	...	0.011843	-0.167558
SendCount	0.239181	0.133838	...	-0.280009	0.128826
ReceivedCount	0.337274	0.139205	...	-0.363109	0.206927
CallsDuration	-0.570715	-0.442521	...	0.171876	-0.163457
MoneySend	-0.337513	-0.389407	...	1.000000	-0.223021
MoneyReceived	0.138866	0.063419	...	-0.223021	1.000000

[22 rows x 22 columns]

Рисунок 4.3.1 - Корреляційна матриця

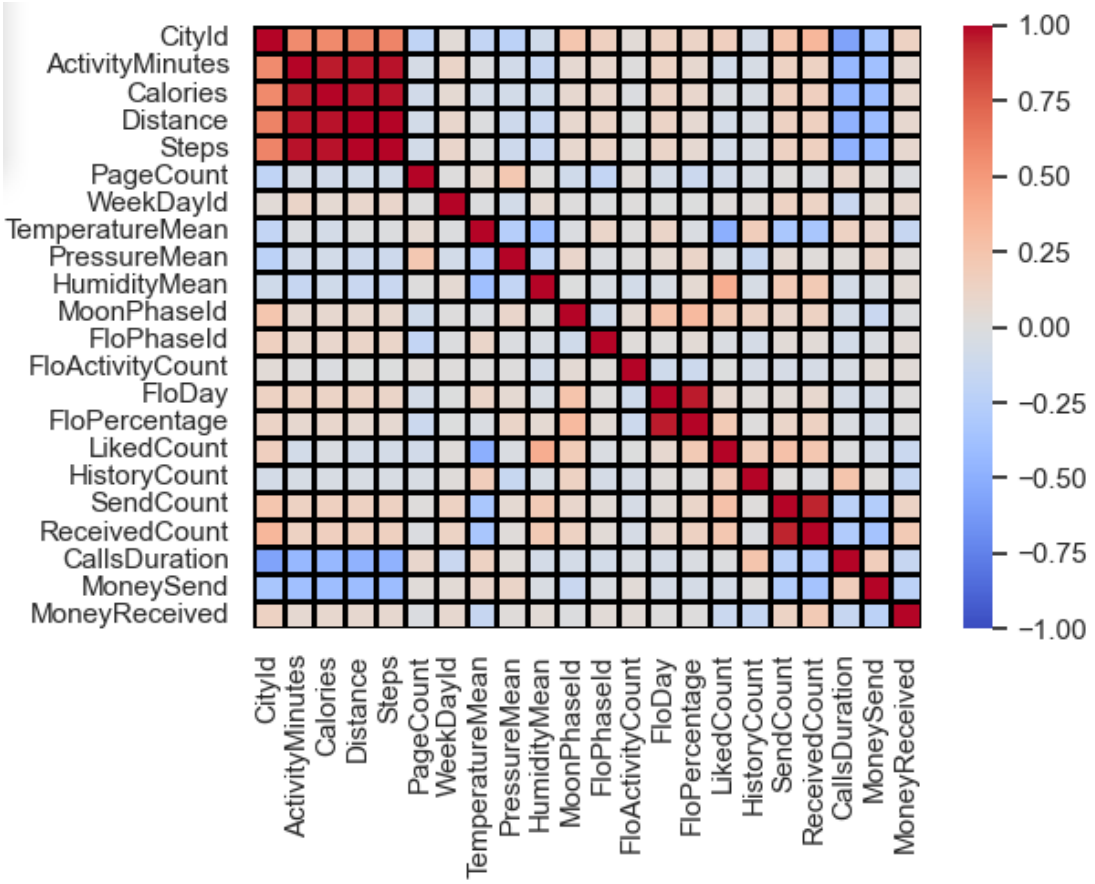

Рисунок 4.3.2 - Графічне відображення кореляційної матриці у вигляді теплової мапи

Коефіцієнт кореляції між показниками ActivityMinutes (тривалість активності) / Calories (кількість спалених калорій) / Distance (подолана відстань) / Steps (кількість здійснених кроків), FloDay(порядковий номер у поточному циклі) / FloPercentage (нормалізований порядковий номер у поточному циклі відносно його довжини), SendCount (кількість відправлених повідомлень) / ReceivedCount (кількість отриманих повідомлень) дорівнює >0.94, що свідчить про дуже високу ступінь кореляції. У випадку даних з джерел Google Fit та Flo подібна залежність передбачувана та її причини очевидні, і для майбутніх досліджень було знехтувано деякими показниками і використано тільки один стовпчик із цих наборів даних. А що стосується дуже високої щільності зв'язку між кількістю відправлених та отриманих повідомлень, цей результат було запам'ятовано та враховано під час аналізу результатів та створення припущень.

Після здійснення оголошених модифікацій, кореляційну матрицю було розраховано знову, з використанням різних методів обчислення коефіцієнтів кореляції.

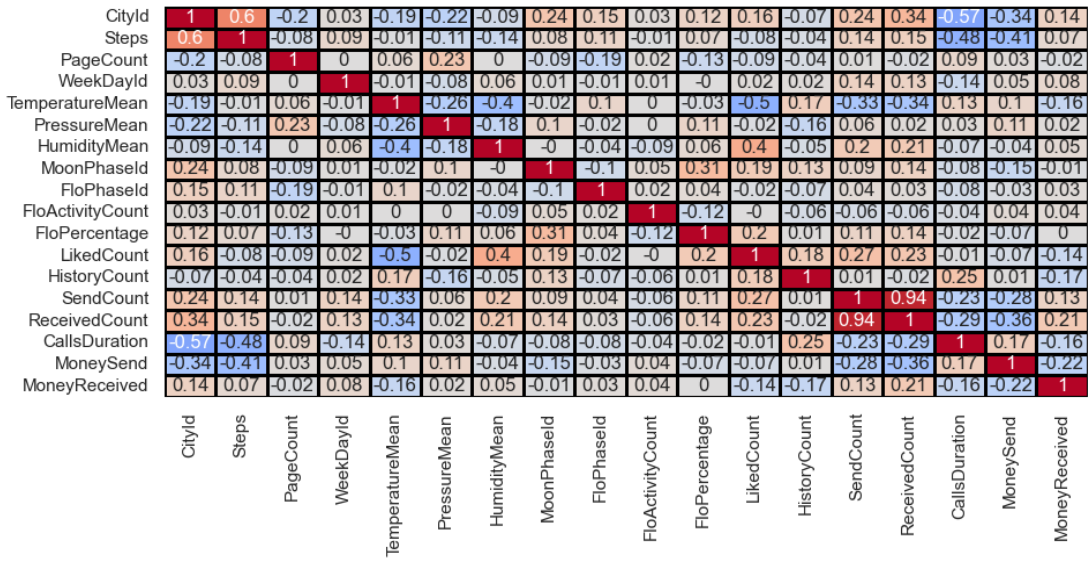

Рисунок 4.3.3 - Графічне відображення кореляційної матриці, обчисленої методом Пірсона, у вигляді теплової мапи

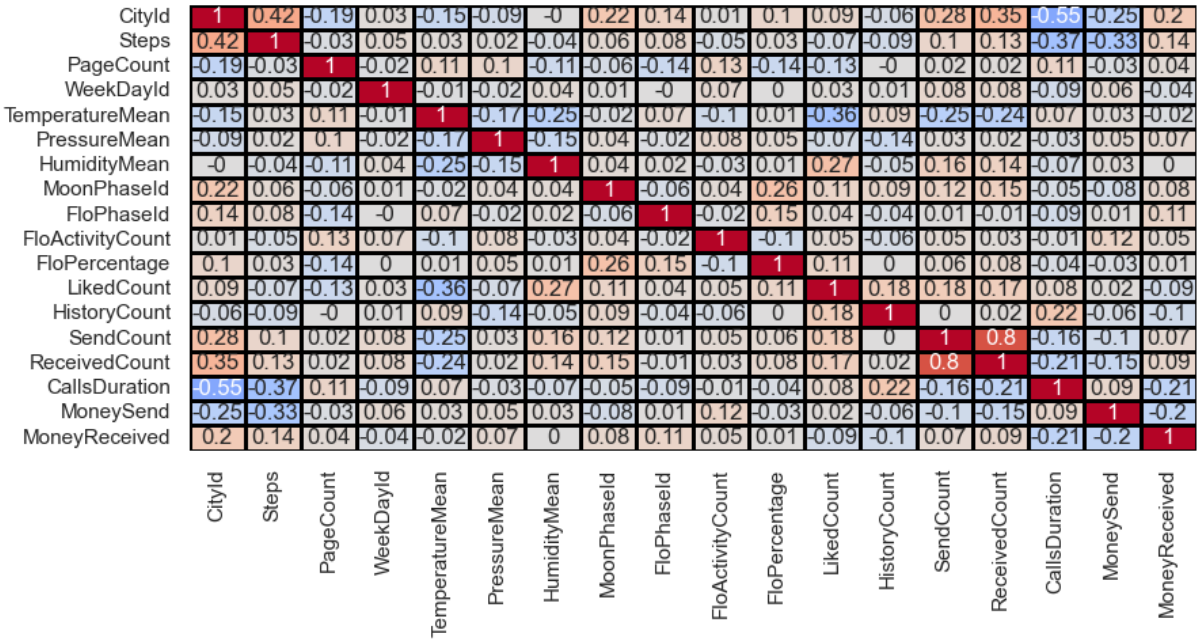

Рисунок 4.3.4 - Графічне відображення кореляційної матриці, обчисленої методом Кендалла, у вигляді теплової мапи

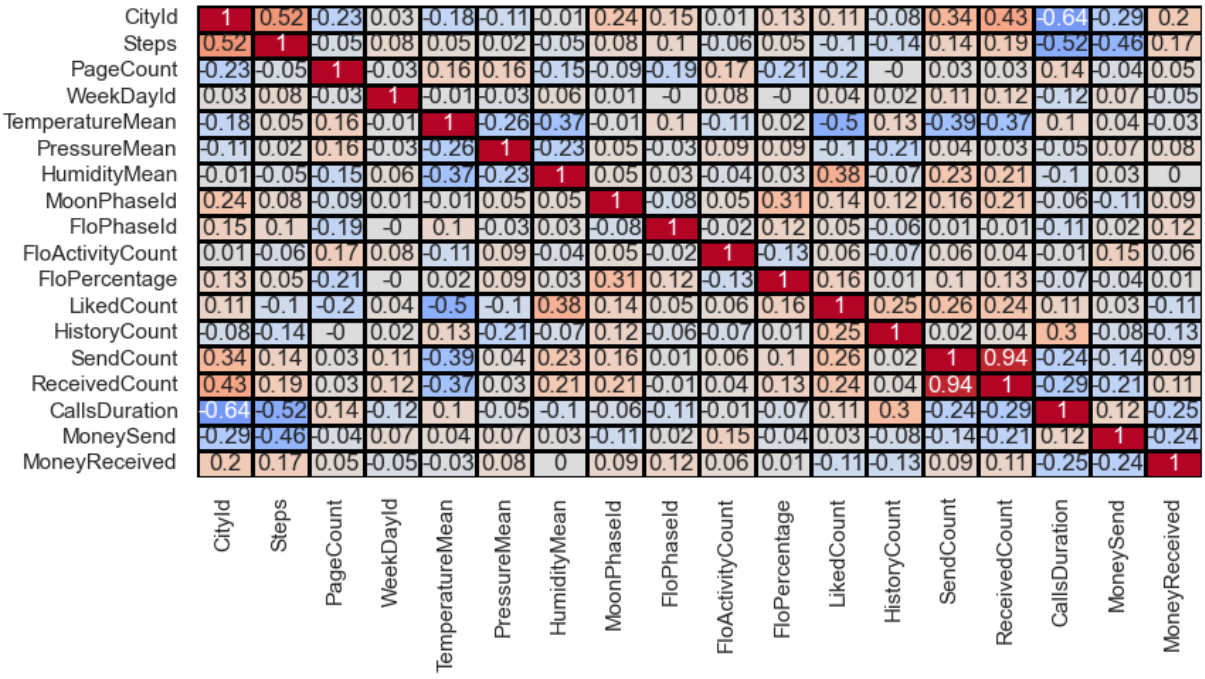

Рисунок 4.3.5 - Графічне відображення кореляційної матриці, обчисленої методом Спірмена, у вигляді теплової мапи

Отримані значення об’єднано у датафрейм, що містить найбільші показники кореляції між кожною парою стовпців та рядків, серед отриманих різними методами.

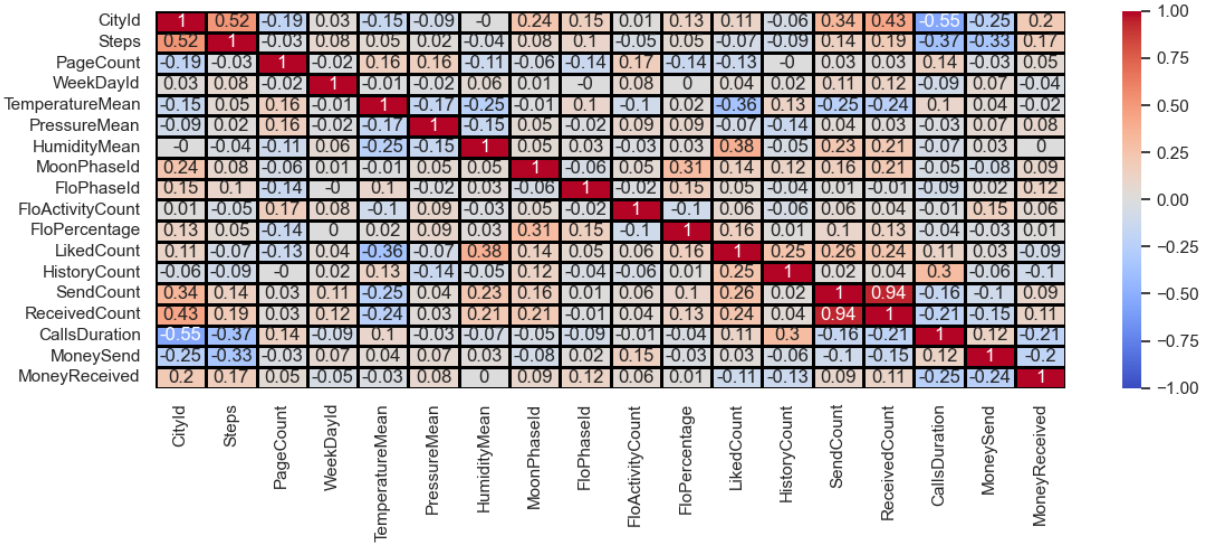

Рисунок 4.3.3 - Графічне відображення результуючої кореляційної матриці у вигляді теплової мапи

Результати оцінки ступеня статистичної значимості обчислених корреляційних зв’язків представлено у таблиці.

Таблиця 4.1 - Оцінка статистичної значимості обчислених корреляційних зв’язків

Абсолютне значення коефіцієнта кореляції по модулю	Щільність (сила) кореляційного зв'язку	Пари показників з відповідною мірою залежності
більше 0.9	дуже висока (статистично значущий зв'язок)	SendCount / ReceivedCount
від 0.7 до 0.9	висока (статистично значущий зв'язок)	-

від 0.5 до 0.7	помітна (статистично не значущий зв'язок)	CityId / Steps CityId / CallsDuration
від 0.3 до 0.5	помірна (статистично не значущий зв'язок)	CityId / SendCount CityId / ReceivedCount Steps / CallsDuration / Steps / MoneySend TemperatureMean / LikedCount HumidityMean / LikedCount MoonPhaseId / FloPercentage HistoryCount / CallsDuration
менше 0.3	слабка (статистично не значущий зв'язок)	всі решта

Порахуємо також суми коефіцієнтів кореляції для кожного показника із іншими, щоб визначити найпотужніші фактори впливу.

```
MoneyReceived =  1.54
WeekDayId =  1.65
FloPhaseId =  2.0
FloActivityCount =  2.03
PressureMean =  2.22
HistoryCount =  2.36
FloPercentage =  2.37
MoneySend =  2.42
MoonPhaseId =  2.48
HumidityMean =  2.5
PageCount =  2.51
TemperatureMean =  2.88
SendCount =  3.01
Steps =  3.02
LikedCount =  3.22
CallsDuration =  3.25
CityId =  3.81
```

Рисунок 4.3.3 - Результат обчислення сумарної сили факторів

Отже, було виявлено найсильніший фактор - місто локації користувача. Він особливо сильно впливає(або залежить) на фізичну активність (в позитивному напрямку), обсяг обміну повідомленнями та тривалість дзвінків Discord (в напрямку зменшення, що є позитивним).

Цікавим та важливим відкриттям було виявлення дуже слабкого рівня впливу фази менструального циклу на життєві показники та інтернет-активність користувача. Незважаючи на те, що суб'єктивне самовідчуття та настрій піддослідної могли значно змінюватись і перебувати в сильній залежності від фази циклу, однак це майже не відображається на виміряних об’єктивних показниках.

4.4 Припущення та шляхи для розвитку

Основні потенційні шляхи розвитку та поглиблення даного дослідження.

Збільшення досліджуваного часового проміжку. Замість 140 днів, можна розглянути більш тривалий період спостережень, наприклад, декілька років. Це дозволить отримати більш повне уявлення про залежності, тенденції, сезонність та зміни в даних. Також це дозволить уникнути виявлення хибних кореляцій, таких як залежність кількості вподобаних у TikTok відео від температури, що є результатом впливу специфіки обраного короткого проміжку.

Розширення вибірки піддослідних. Для отримання більш репрезентативних результатів варто залучити більше учасників дослідження. Це дозволить проведення порівняльного аналізу, що може допомогти як краще визначати індивідуальні особливості та потреби кожного користувача на тлі інших піддослідних, так і робити узагальнені висновки, уникаючи спотворення результатів через ці особливості. Можна виявити кореляції для людей різної міри наближеності, знайти міру впливу піддослідних на одне одного, а також впливу на них спільних зовнішніх факторів.

Розширення множини джерел даних. Для більш повного розуміння взаємозв'язків і факторів впливу можна долучити у подальші дослідження

такі джерела, як: щоденник емоційного стану (наприклад Moodpress), сервіс для прослуховування музики (наприклад Apple Music), розумний годинник, додатки для замовлення доставок, тощо. Це надасть більш детальну інформацію про інтернет-активність та показники у різних сферах життя користувача.

Покращення точності даних. Для отримання більш достовірних результатів важливо покращити точність зібраних даних. Це можна зробити шляхом використання більш точних сенсорів та вимірювальних пристроїв. Також слід звернути увагу на правильність збору та обробки даних, а також на видалення аномалій або помилок, що можуть впливати на результати.

Перехід до коротших і точніших інтервалів. Замість аналізу щоденних показників можна, наприклад, перейти до погодинних інтервалів. Такий підхід дозволить виявити патерни, залежності і динаміку змін у даних на більш детальному рівні. Також стане можливим дослідити активність користувача в залежності від числа місяця та часу доби і, наприклад, робити припущення щодо графіку сну піддослідного на основі цих даних.

Також важливе подальше перетворення інструменту у формат додатку, більш дружнього для користувачів, з простим і придатним для розуміння інтерфейсом.

Хоча для більшості людей пропозиція поділитися усією перерахованою конфіденційною інформацією викличе тільки жах і відторгнення, знайдется чимало тих, для кого користь переважить занепокоєння.

ВИСНОВКИ

Метою роботи була розробка інструменту для здійснення збору, огляду, підготовки, візуалізації та аналізу даних інтернет-активності користувача та показників у сферах його життя.

Для успішного виконання поставленої мети було проведено дослідницьку роботу з метою визначення безлічі джерел даних, які підлягають аналізу. Це включало ідентифікацію різних платформ, програм або систем, які можуть надати інформацію про інтернет-активність користувача та показники в різних аспектах його життя. Після визначення цих джерел було отримано доступ до кожного з них, використовуючи унікальний для кожного джерела підхід та алгоритм дій. Далі був виконаний процес збору даних з кожного джерела та їх уніфікація, щоб забезпечити єдиний формат та структуру даних для подальшого аналізу та порівняння, створено основу для аналізу та візуалізації.

У ході роботи було виконано обчислення статистичних метрик, які дозволили отримати кількісне уявлення про дані та їх характеристики. Це включало розрахунок середніх значень, медіани, стандартного відхилення та інших показників, які допомогли зрозуміти загальну картину та розподіл даних. Для візуалізації та наочного подання результатів були побудовані графіки та діаграми, які допомогли виявити тренди, взаємозв'язки та особливості даних.

Далі було проведено кореляційний аналіз усіх показників, щоб виявити ступінь взаємозв'язку між різними змінними. Це дозволило визначити наявність, направленість та силу залежності між показниками.

Було проведено оцінку та аналіз отриманих результатів. Були зроблені висновки про інтернет-активність користувача та показники у різних сферах

його життя. Також були сформульовані припущення та гіпотези, які можуть бути подальшими об'єктами дослідження та аналізу.

Було сформульовано шляхи для розвитку інструменту, сформовано базу для подальших досліджень. Це відкриває перспективи для більш глибокого та всебічного аналізу інтернет-активності та її впливу на різні аспекти життя користувачів.

Незважаючи на обмежену (у розмірі досліджуваного часового проміжку та кількості піддослідних) вибірку даних, за результатами аналізу було зроблено численні придатні для використання на практиці висновки та варті перевірки припущення.

Потенційно розроблений інструмент можна використати у сфері соціологічних досліджень, бізнес-аналітиці для соціальних мереж та інших цифрових застосунків. У роботі зазначено можливі шляхи розширення функціоналу та сфер аналізу, кількісного та якісного розвитку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Rambam Maimonides Medical Journal, 2010: "Has human evolution stopped?".
2. Reis, Joe; Housley, Matt (2022) "Fundamentals of Data Engineering". O'Reilly Media, Inc. ISBN 9781098108304.
3. Pew Research Center, April 2021, "Social Media Use in 2021".
4. Denissen, J. J. A., Butalid, L., Penke, L., & van Aken, M. A. G. (2008). The effects of weather on daily mood: A multilevel approach.
5. International Journal of Psychophysiology, Volume 94, Issue 3, December 2014, Pages 351-357.
6. Public Health Nutrition, Volume 2, Supplement 3a, March 1999. - Режим доступу: https://doi.org/10.1017/S1368980099000567.
7. General Data Protection Regulation, GDPR; Regulation (EU) 2016/679.
8. Закон України “Про захист персональних даних” від 01.06.2010 № 2297-VI. - Режим доступу: https://zakon.rada.gov.ua/go/2297-17.
9. Python language. – Режим доступу: https://www.python.org/.
10. Pandas open-source tool. – Режим доступу: https://pandas.pydata.org/.
11. Numpy package. – Режим доступу: https://numpy.org/.
12. Matplotlib library. – Режим доступу: https://matplotlib.org/.
13. Бахрушин В.Є. Методи аналізу даних : навчальний посібник для студентів / В.Є. Бахрушин. – Запоріжжя : КПУ, 2011.
14. Seaborn library. – Режим доступу: https://seaborn.pydata.org/.
15. Google Архіватор. – Режим доступу: https://takeout.google.com/.
16. Веб-скрапінг за допомогою Python: покроковий посібник. – Режим доступу: https://encr.pw/fri42.
17. Календар фаз місяця. - Режим доступу: https://www.spaceweatherlive.com/uk/kalendar-misyachnih-faz.html.
18. Flo. - Режим доступу: https://flo.health/contact-us.

19. TikTok Support. - Режим доступу: https://support.tiktok.com/en/.
20. Telegram. – Режим доступу: https://telegram.org/.
21. Discord. - Режим доступу: https://support.discord.com/hc/en-us.
22. Privat24. - Режим доступу: https://next.privat24.ua/.
23. Paluch AE et al. JAMA Network Open, 2021. - Режим доступу: https://encr.pw/zUYql.
24. FLAGSHIP REPORT, "GlobalWebIndex’s flagship report on the latest trends in social media", 2020 - Режим доступу: https://encr.pw/rzkVH.
25. TikTok User Statistics. - Режим доступу: https://backlinko.com/tiktok-users.
26. Michael Wise, 15 Text Messaging Statistics Every Business Should Know. - Режим доступу: https://encr.pw/Jwyfk.
27. Statistics report about Discord. - Режим доступу: https://www.statista.com/study/133074/discord/.