Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних Наук
(повна назва)
Кафедра Програмної інженерії
(повна назва)

КВАЛФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
Дослідження методів аналізу даних та їх застосування у системі банківського фінансового менеджменту
(тема)

Виконав:
Студент 2 курсу, групи ІПЗм-22-2
Новосельцев І. І.
(прізвище, ініціали)
Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Керівник проф. Руткас А. Г.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри : _____ З. В. Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий(магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
зав. каф. ПІ, к.т.н., проф.
_____ Зоя Дудар
(підпис)
«_____» _____20 _____р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Новосельцеву Іллі Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи: «Дослідження методів аналізу даних та їх застосування у системі банківського фінансового менеджменту»
затверджена наказом від «29» березня 2024 р. № 250 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії _____ 202 р.

3. Вхідні дані до роботи *технічне завчання, календарний план, методичні вказівка. Технології: PostgreSQL Server, середовище розробки Intelij Idea, Java (Spring framework), React.js.*

4. Перелік питань, що потрібно опрацювати в роботі: *вступ, аналіз предметної області, дослідження методів аналізу даних, формування вимог до програмної системи, архітектура та проектування програмного продукту, висновки, перелік джерел посилань.*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін	Відмітки про виконання
1	Аналіз предметної галузі	29.01.2024	Виконано
2	Створення специфікації ПЗ	08.02.2024	Виконано
3	Проєктування та розробка ПЗ	11.02.2024	Виконано
4	Тестування та дослідна експлуатація ПЗ	22.03.2024	Виконано
5	Написання пояснювальної записки	11.06.2024	Виконано
6	Перевірка пояснювальної записки	17.06.2024	Виконано
7	Оцінка роботи рецензентами	19.06.2024	Виконано
8	Здача роботи в електронний архів	22.06.2024	Виконано
9	Попередній захист	23.06.2024	Виконано
10	Допуск до захисту у зав. кафедри	24.06.2024	Виконано
11	Захист кваліфікаційної роботи	25.06.2024	Виконано

Дата видачі завдання «23» січня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Руткас А. Г.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 79 стор., 20 рис., 29 джерел.

ДОСЛІДЖЕННЯ, МЕТОДИ АНАЛІЗУ ДАНИХ, ЗАСТОСУВАННЯ МЕТОДІВ АНАЛІЗУ ДАНИХ, ФІНАНСОВИЙ МЕНДЖМЕНТ, JAVA, PYTHON, SPRING FRAMEWORK, REACT.JS

Об'єкт дослідження – застосування методів аналізу даних у системі банківського фінансового менеджменту.

Метою є проведення дослідження застосування методів аналізу даних у сучасних фінансових системах банківського менеджменту та оцінка їх ефективності.

Методи розробки базуються на середовищі розробки СКБД PostgreSQL Server, середовище розробки Intelij Idea, технології Java (Spring framework), Python, React.js.

У результаті роботи було досліджено методи аналізу даних та їх застосування в сучасних системах банківського менеджменту. Створено систему, яка демонструє приклади використання методів із поставленої задачі з відповідною документацією , а також була виконана підготовча робота для подальших досліджень.

RESEARCH, DATA ANALYSIS METHODS, USAGE OF DATA ANALYSIS METHODS, FINANCIAL MANAGEMENT, JAVA, PYTHON, SPRING FRAMEWORK, REACT.JS

The object of research - a usage of data analysis methods in a banking management system .

The aim is to make a research of a usage of data analysis methods in a modern banking management systems and estimate of their efficiency.

Development methods are based on the PostgreSQL Server database development environment, Intelij Idea development environment, Java technologies (Spring framework), React.js.

As a result there was made a research of data analysis methods and their usage in modern banking management systems. Was made a system that represents a usage of methods from aim research with corresponding documentation and also there was made a preparation work for the future research.

Я, Новосельцев Ілля Ігорович, студент гр. ІПЗМ-22-2, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів аналізу даних та їх застосування у системі банківського фінансового менеджменту», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

У сучасному банківському світі, де величезні обсяги даних постійно наростають, аналіз цих даних стає ключовим елементом успішного управління та прийняття стратегічних рішень. І саме ця потреба в глибокому аналізі даних для досягнення оптимального банківського менеджменту є основою для даної дослідницької роботи.

Завдання аналізу даних стає ще більш важливим у контексті банківського менеджменту, де правильний аналіз може визначити конкурентну перевагу, забезпечити ефективне управління ризиками та забезпечити інноваційні стратегії обслуговування клієнтів.

Точність в аналізі клієнтських патернів, здатність передбачати ризики та спроможність класифікувати дані стають ключовими компонентами для розбудови стійкої та конкурентоспроможної фінансової системи. Ця дослідницька робота має на меті розглянути, як саме аналіз даних може стати ключовим фактором для модернізації фінансового сектору та підвищення якості послуг. Ми вдаватимемося у детальне обговорення чотирьох основних напрямків: аналіз клієнтських даних, прогнозування ризиків, класифікація та кластеризація даних, прогнозування та оптимізація.

Фінансові установи активно використовують аналіз клієнтських даних для створення індивідуальних підходів до обслуговування клієнтів та розвитку нових фінансових продуктів. Це дозволяє побудувати більш гнучкі та персоналізовані стратегії обслуговування.

Прогнозування ризиків виступає як важливий інструмент управління ризиками та забезпечення фінансової стабільності. Засоби прогнозування допомагають ідентифікувати потенційні ризики та вчасно реагувати на них.

Класифікація та кластеризація даних є невід'ємною частиною аналітичного процесу в банківському секторі. Ці методи допомагають у виявленні схожих груп

клієнтів та визначенні унікальних потреб кожної з них.

Прогнозування та оптимізація виступають як ключові елементи в управлінні фінансовими ресурсами та оптимізації стратегій банку для досягнення максимальної ефективності.

Ця дослідницька робота ставить за мету вивчити значення та потенціал аналізу даних у фінансовому секторі, звертаючи увагу на те, як ці методи можуть сприяти вдосконаленню управління та реагуванню на виклики галузі.

Для розробки необхідно використати середовище розробки Intellij Idea, мову програмування JavaScrip для фронтенд частини, Java, Spring, Python для бекенд частини та СКБД PostgreSQL.

1 ОПИС ПРОБЛЕМНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Фінансова сфера переживає період значних змін, починаючи від модернізації традиційних банківських послуг до виникнення інноваційних технологій та конкуренції зі сторони фінтех-стартапів. Розширення цифрових технологій, величезний обсяг генерованих даних та зростаючі очікування споживачів ставлять перед фінансовими установами завдання адаптуватися та вдосконалювати свої сервіси.

Ринок фінансових послуг перетворюється завдяки нововведенням, що пропонуються фінтех-компаніями. Ці компанії, часто використовуючи передові технології та аналітичні методи, здатні надавати конкурентоспроможні послуги та пропонувати новаторські рішення у фінансовій сфері. Це змушує традиційні банки шукати способи оптимізації та модернізації своїх процесів.

Щодо можливих досліджень в цій сфері, варто звернути увагу на кілька ключових аспектів, що пов'язані з темою. Аналіз клієнтських даних, прогнозування ризиків та оптимізація фінансових процесів є головними напрямками, які привертають увагу дослідників. Дослідницькі роботи в цих сферах можуть спрямовуватися на розробку нових алгоритмів аналізу даних, виявлення нових патернів у клієнтському поведінці та створення унікальних стратегій управління ризиками [1].

Таким чином, дослідження в сфері фінансового менеджменту та аналізу даних відкриває широкий спектр можливостей для розвитку нових методів, що дозволяють банкам оптимізувати свою діяльність та забезпечувати високий рівень обслуговування клієнтів.

Якщо казати більш конкретно при користь методів аналізу даних на практиці, то це можна описати наступним чином. Аналіз клієнтських даних дозволяє створювати персоналізовані стратегії обслуговування, відповідати на потреби

клієнтів та підвищувати рівень задоволеності. Це важливо для збільшення вірогідності утримання клієнтів та розвитку довгострокових відносин. Прогнозування ризиків на основі аналітичних методів дозволяє фінансовим установам передбачати можливі проблеми та реагувати на них напередодні. Аналіз класифікації та кластеризації даних може сприяти вдосконаленню внутрішніх процесів, забезпечуючи більш ефективне розподілення ресурсів. Прогнозування та оптимізація допомагає управляти фінансовими ресурсами та знаходити оптимальні інвестиційні шляхи для досягнення максимального результату [2].

Ці методи не лише надають можливості для аналізу даних, а й створюють основу для прийняття управлінських рішень, які базуються на об'єктивних фактах та реальних даних. Такий підхід сприяє підвищенню ефективності, зниженню ризиків та забезпеченню стратегічного розвитку у фінансовій сфері.

1.2 Необхідність використовування методів аналізу даних у фінансовому секторі

Необхідність використання методів аналізу даних у банківському менеджменті визначається широким спектром можливостей, які вони пропонують для оптимізації фінансових процесів та вдосконалення стратегічного управління. Аналітичні методи стали ключовим інструментом для вирішення складних завдань управління ризиками, оптимізації клієнтського досвіду та розвитку нових продуктів.

Серед ключових аспектів варто звернути увагу на управління ризиками та безпекою. Банки активно використовують аналіз даних для прогнозування ризиків кредитування та виявлення фінансових аномалій. Наприклад, застосування алгоритмів машинного навчання для аналізу кредитної історії допомагає точніше визначати ризик неплатежів та кредитних зобов'язань, що забезпечує більш ефективне управління ризиками [3].

Крім того, аналіз даних є ключовим інструментом для підвищення ефективності бізнес-процесів. Він допомагає банкам оптимізувати ділові процеси,

прогнозувати попит на фінансові продукти, а також адаптувати їх до змін у вимогах ринку. Наприклад, за допомогою аналітики клієнтських даних банки можуть ідентифікувати вигідні сегменти клієнтів та вдосконалити стратегії обслуговування, що призводить до збільшення задоволеності клієнтів та збільшення віддачі від інвестицій.

Також, розвиток нових продуктів є важливим аспектом використання аналітичних методів у банківській сфері. Аналіз даних дозволяє ідентифікувати нові ринкові можливості та створювати інноваційні фінансові продукти, такі як персоналізовані фінансові послуги чи інвестиційні рішення, що відповідають потребам сучасних клієнтів.

1.3 Вплив методів аналізу даних на фінансові установи

Методи аналізу даних суттєво впливають на фінансові установи у різних аспектах їхньої діяльності. Ці методи мають ключове значення для оптимізації процесів та управління ризиками, дозволяючи зробити більш обґрунтовані рішення та підвищити ефективність роботи фінансових установ.

Аналітика даних надає можливість управляти ризиками, оцінювати ризики кредитування, прогнозувати та оптимізувати інвестиційні портфелі. Вона також допомагає у виявленні та мінімізації ризиків, пов'язаних з кредитами, інвестиціями та змінами на фінансових ринках.

Додатково, аналітика даних полегшує вироблення та запровадження стратегій взаємодії з клієнтами, персоналізуючи пропозиції та послуги для поліпшення клієнтського досвіду. Також вона допомагає виявляти аномальні патерни, які можуть свідчити про шахрайство чи кібератаки, забезпечуючи безпеку операцій та запобігаючи фінансовим втратам. В цілому, методи аналізу даних сприяють підвищенню ефективності роботи фінансових установ, полегшуючи прийняття рішень та знижуючи ризики [4].

Розглянемо доцільність методів аналізу даних на прикладах компаній та їх

систем в реальному світі.

HSBC (Hongkong and Shanghai Banking Corporation) – один із найбільших банків у світі, що володіє глобальною мережею та надає фінансові послуги різних типів. Використання методів аналізу даних у HSBC просувається через кілька напрямків:

- ризиковий аналіз та управління ризиками – HSBC активно використовує аналітику даних для оцінки та керування ризиками. Вони аналізують клієнтські дані для виявлення потенційно шахрайських та ризикованих операцій;
- покращення обслуговування клієнтів – банк використовує аналіз даних для створення персоналізованих пропозицій та рішень для клієнтів, що сприяє підвищенню їхнього задоволення;
- маркетинг та реклама – HSBC застосовує аналіз даних для вдосконалення стратегій маркетингу та реклами, зорієнтованих на потреби різних клієнтських груп;
- фінансове планування та прогнозування – аналітика даних використовується для прогнозування фінансових тенденцій, допомагаючи банку приймати обґрунтовані інвестиційні рішення;
- боротьба зі злочинністю та відмиванням грошей – HSBC використовує аналітику даних для виявлення підозрілих операцій та запобігання відмиванню грошей.

JPMorgan Chase – один з найбільших фінансових конгломератів у світі. Вони впроваджують методи аналізу даних у різних напрямках своєї діяльності:

- ризиковий аналіз та управління ризиками – JPMorgan використовує аналітику даних для оцінки ризиків інвестицій, кредитів та фінансових операцій. Це допомагає їм приймати обґрунтовані рішення щодо управління ризиками;
- технології блокчейну та криптовалюта – JPMorgan вивчає дані блокчейну

та ринок криптовалют для розробки нових фінансових продуктів та послуг, що допомагає їм залишатися попереду у сфері інновацій;

- аналіз клієнтських даних – банк використовує аналіз даних для розуміння поведінки та потреб клієнтів, створюючи персоналізовані фінансові рішення для кожного клієнта;
- торговельні стратегії – JPMorgan використовує аналітику даних для розробки стратегій торгівлі на фінансових ринках, що дозволяє їм впевненіше приймати рішення щодо інвестицій та торговельних операцій;
- боротьба зі злочинністю та фінансовими шахрайствами – штучний інтелект та аналітика даних використовуються для виявлення підозрілих фінансових операцій та шахрайства.

American Express – фінансова компанія, яка спеціалізується на фінансових послугах та платіжних системах. Вони активно використовують методи аналізу даних у багатьох аспектах своєї діяльності:

- персоналізовані пропозиції для клієнтів – American Express використовує аналіз даних для створення персоналізованих пропозицій та послуг для своїх клієнтів. Вони аналізують покупки та інші дані, щоб надавати індивідуальні рекомендації;
- боротьба зі шахрайством – штучний інтелект та аналітика даних використовуються для виявлення підозрілих транзакцій та запобігання шахрайству;
- прогнозування та управління ризиками – American Express використовує аналіз даних для прогнозування фінансових ризиків та управління ними, що допомагає їм раціонально керувати кредитуванням;
- розробка нових продуктів – компанія використовує дані для розробки нових фінансових продуктів та програм, орієнтованих на потреби своїх клієнтів.

Goldman Sachs – міжнародний банк, що надає різноманітні фінансові послуги та консультації. Вони активно використовують методи аналізу даних для

покращення своєї діяльності та різних аспектів фінансового управління:

- торговельні стратегії – Goldman Sachs використовує аналіз даних для розробки торговельних стратегій та управління інвестиціями. Вони використовують складні алгоритми та моделі для прийняття рішень у фінансових операціях;
- управління портфелем та ризиками – аналітика даних використовується для аналізу фінансових ризиків та управління портфелем інвестицій, дозволяючи зменшити ризики та оптимізувати стратегії;
- інвестиційні рішення – Goldman Sachs використовує аналіз даних для прийняття обґрунтованих інвестиційних рішень. Штучний інтелект допомагає їм розуміти ринкові тенденції та прогнозувати подальший розвиток ситуації;
- стратегії управління активами – компанія використовує аналітику даних для розробки стратегій управління активами своїх клієнтів, допомагаючи їм оптимізувати фінансові плани та досягати своїх цілей.

1.4 Виклики та перешкоди у застосуванні методів аналізу даних в банківській сфері

Застосування методів аналізу даних у свою чергу дуже нетривіальна задача, особливо в банківській сфері, вона має свої виклики та перешкоди. Впровадження аналітики потребую значних змін в сфері менеджменту, програмної розробки системи та змін у відділі кадрів. Розглянемо основні проблеми впровадження аналізу даних:

Конфіденційність та захист даних, що підлягають аналізу. Банки працюють з величезними обсягами конфіденційної інформації про клієнтів. Забезпечення безпеки цих даних під час аналізу відіграє критичну роль і стає важливим аспектом в процесі використання аналітичних методів. Бували випадки, коли під час процесів аналізу, здійснювалися зловмисні хакерських атаки на банківський системи,

наслідком чого ставав витік персональних даних, таких як номери карт, телефонів тощо.

Недостатня якість даних. Підготовка даних для аналізу є складним завданням. Часто зустрічаються проблеми зі збором, структуруванням та якістю даних, що може негативно впливати на точність аналітичних моделей. Також бувають випадки, коли досить не точні дані давали неправильну та нереальну аналітику, що впливала на прийняття невірних рішень у подальшому розвитку компанії.

Неоднорідність даних та їх обмеження може бути досить серйозною проблемою. Різноманітність джерел даних у банківській галузі може створювати проблеми в їх інтеграції та аналізі. Обмеження доступу до певних видів ключових даних також ускладнює процес аналізу.

Супровідний персонал та кадрові обмеження. Навички у сфері аналітики не завжди доступні у фінансових установах, що може призвести до нестачі кваліфікованих кадрів, які вміють працювати з аналітикою даних в банківській сфері. Особливо ця проблема торкається малобюджетного сегменту фінансового ринку.

Регулювання та відповідність. Фінансові установи підлягають строгим правилам та вимогам щодо збереження та обробки даних. Це може призводити до складнощів у впровадженні певних аналітичних підходів через потребу відповідності законодавству та регуляціям.

Масштабування та інфраструктура. Обробка великих обсягів даних в реальному часі може стати викликом для існуючої інфраструктури банків. Масштабування систем та їх готовність до зростання обсягів даних важливі для успішного застосування аналітики. Це є проблемою, бо більшість сучасних банків були створені ще багато років потому, їхня кодова система та рішення потребують значних змін та оновлень, а коли ми кажемо про такі великі системи як банкінг, то впровадження цих змін є дуже великою та складною перешкодою.

У розв'язання цих проблем може допомогти вдосконалення технологічних

платформ, залучення кваліфікованих кадрів, розробку стратегій захисту даних та постійне вдосконалення методів обробки та аналізу інформації.

1.5 Веб-орієнтована система

Передумовами для створення проєкту стала необхідність зробити програмну систему, що використовуватиме методи аналізу даних для детального аналізу фінансових транзакцій по картках людини, задля створення моделі, яка прогнозуватиме майбутні витрати та допомагатиме оптимізувати бюджет на основі цих прогнозів.

На виході повинна бути повноцінна робоча система з відтвореним повним головним функціоналом.

Оскільки кожна людина завжди щось купує, буквально робить це щоденно, їй буде корисна аналітика своїх затрат. Система надасть можливість автоматично розділити транзакції на групи з подібними характеристиками. Наприклад, одна група може включати транзакції, пов'язані з продуктами харчування, інша - з покупками домогосподарства, і так далі. Потім ці групи будуть проаналізовані на основі історичних даних та будуть створені прогнозові дані витрат по цих групах для користувача.

Правильний аналіз допоможе у менеджменті фінансів людини, дозволить краще розуміти, куди йдуть кошти та де можна зробити зміни для економії. Також індивідуальний аналіз витрат на може допомогти пропонувати персоналізовані фінансові поради або плани, зокрема, що стосується розподілу бюджету або підвищення збережень.

1.6 Постановка задачі

Метою кваліфікаційної роботи є дослідження існуючих сучасних методів аналізу даних у системі банківського фінансового менеджменту та розробка веб-орієнтованої системи, що демонструє використання цих методів на практиці.

Вимоги до проєкту полягають у дослідженні існуючих методів аналізу даних, що використовуються у системах банківського фінансового менеджменту та створенні програмної системи, яка демонструю на практиці, як ці методи можуть бути застосовані. Основними функціональними ідеями для реалізації такої системи є прогнозування майбутніх витрат та оптимізація бюджету на основі прогнозів.

Досягнути поставленої цілі буде робитися шляхом використання моделей машинного навчання, які аналізують історичні дані транзакцій користувача для прогнозування майбутніх витрат із врахуванням різноманітних факторів, таких як тип транзакції, категорія витрат тощо, для точнішого прогнозування. Також використання алгоритмів прогнозових моделей для автоматичної оптимізації бюджету користувача із врахуванням його фінансових цілей та обмежень бюджету, для генерації рекомендацій щодо оптимального розподілу коштів.

Відмінністю даної системи повинна бути легкість і зручність у застосуванні та велика ефективність для людини, яка має на меті ефективно контролювати свої фінансові витрати. Людині не потрібно буде тримати все в голові та завантажувати себе думками про це.

Система повинна використовувати методи аналізу даних для:

- категоризації своїх фінансових транзакцій (розділяти та категоризувати транзакції);
- управління фінансами (аналіз витрат за різними категоріями, що дозволить краще розуміти, куди йдуть кошти);
- прогнозування витрат (індивідуальний аналіз, що буде прогнозувати витрати для кращого розподілу бюджету або підвищення збережень).

Ціллю роботи є створення розумного та зручного сервісу для менеджменту фінансових транзакцій. Головними критеріями успіху є збільшення кількості користувачів та їх висока оцінка, яка має бути досягнена легкістю та зручністю інтерфейсу, а також результатом, отриманим впродовж користування, тобто покращення управління персональними фінансами. Проєкт орієнтований на групу

людей дуже великого вікового діапазону, від 14-80 років. Як показують дослідження, людина починає взаємодіяти з банківськими картками, за допомогою гаджетів приблизно з 14 років, також можна зустріти людей, які не закінчують користуватися ними і в поважному віці.

Тож розроблена в результаті система підійде людям багатьох вікових категорій якім потрібен персоналізований аналіз та прогнозування витрат за допомогою телефонного або комп’ютерного додатку.

Інформаційна система повинна задовільнити наступні вимоги:

– полегшити та структурувати інформацію по транзакціям;
– підвищити розуміння та покращити планування витрат;
– зробити процес аналізу транзакцій зручним для користувачів;

Критеріями успіху можуть бути наступні фактори:

– набути великої зацікавленості серед людей;
– розвиток системи до міжнародного масштабу;
– інтеграція із різними банками та фінансовими додатками по типу PayPal.

Після досягнення даних критеріїв розробка системи може вважатися успішною.

В подальших версіях додатку повинен бути реалізований наступний функціонал:

– розробка додатку для IOS та Android;
– інтеграція з додатками найбільших банків та інших фінансових систем, спочатку українського ринку, а потім міжнародного.
– додавання інтерактивних та змагальних ігор для підвищення інтересу до запису транзакцій;
– створення системи досягнень;
– перехід на багатомовні версії;
– додавання можливості ділитись своєю «transaction history» з іншими користувачами.

В поточній версії буде відсутня можливість інтеграції із додатками банків України за довгострокові періоди, хоча це й полегшило б розуміння для багатьох людей. Головною метою на даний момент є чітке та правильне відтворення основного фунціоналу, а саме надання прогнозування та варіантів оптимізації витрат. Обмеження також присутні з боку збереження даних, тому що на даному етапі немає можливості мати належне віддалене сховище. Також виключеннями є реалізація додаткового функціоналу, який не є основним на даний момент, такого як створення багатомовної версії.

Розроблювана програмна система повинна відповідати приведеним вище критеріям.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДІВ АНАЛІЗУ

2.1 Аналіз існуючих методів аналізу даних

Система банківського фінансового менеджменту є дуже широкою в своєму розумінні, якщо казати детальніше, то вона поділяється на власні сегменти, які мають певну пріоритетність для цієї сфери, такі як: управління ризиками, планування і бюджетування, управління капіталом, управління кредитами тощо. Усі ці сегменти поєднує одна необхідність для успішного функціонування – ця необхідність є прогнозуванням фінансових показників. Бо прогнозування допомагає оцінити майбутні ризики, такі як кредитні ризики, ринкові ризики та операційні ризики, воно є основою для створення бюджетів і фінансових планів, та дозволяє оцінити майбутні потреби в капіталі, враховуючи регуляторні вимоги і фінансові потреби банку. Тому в процесі дослідження планується розглянути методи аналізу даних, що конкретно відносять до контексту планування фінансових показників.

Конкретніше, статистично майже в усіх банківських системах використовуються такі основні методи аналізу для прогнозування фінансових показників як:

- лінійна регресія – метод є одним з найпростіших і найпоширеніших методів прогнозування. Він використовується для моделювання залежності між однією або кількома незалежними змінними і залежною змінною. Лінійна регресія дозволяє побудувати прогнозні моделі для різних фінансових показників, таких як доходи, витрати, прибуток тощо;
- аналіз часових рядів (Time Series Analysis) – метод використовується для аналізу даних, які збираються через регулярні проміжки часу. Цей метод дозволяє виявити тренди, сезонність і циклічність у фінансових показниках, що допомагає будувати точні прогнози на майбутнє;
- метод Монте-Карло (Monte Carlo Simulation) – використовується для моделювання і аналізу впливу невизначеності на фінансові показники. Він

дозволяє створювати сценарії і прогнозувати майбутні результати на основі випадкових вибірок. Цей метод часто застосовується для оцінки ризиків і планування інвестицій;

- експоненційне згладжування – є методом прогнозування, який враховує як минулі дані, так і їхні тренди і сезонні компоненти. Він підходить для аналізу часових рядів і часто використовується для прогнозування продажів, витрат і інших фінансових показників;
- кластерний аналіз – базовий метод для всіх інших методів, бо він використовується для сегментації даних на групи з подібними характеристиками. Цей метод допомагає виявити групи клієнтів з подібною поведінкою або транзакціями, що дозволяє більш точно прогнозувати їхні майбутні витрати або доходи.

Для створення порівняльної характеристики було відкинуто такі методи як часові ряди та метод Монте-Карло із наступних причин. Для надання точних результату цим методам потрібні дуже великі об’єми даних різної сезонності та великої кількості характеристик. Такі дані відбираються банками багато років та зберігаються у великих базах даних. А в контексті мого дослідження майже неможливо знайти датасет, який якістю та кількістю свої даних відповідав би вимогам цих методів.

2.2 Огляд методу аналізу Лінійна регресія

Лінійна регресія застосовується для прогнозування значень однієї змінної на основі іншої або декількох інших. Це допомагає встановити залежності між змінними, прогнозування та оцінка впливу змінних на результати. Банки використовують лінійну регресію для прогнозування різних показників таких як витрати, кредитного ризик, тощо. Суть цього методу полягає в тому, щоб визначити та кількісно оцінити вплив різних факторів на показники інтересу, у нашому випадку показник інтересу це майбутні витрати. Лінійна регресія демонструє залежність між

однією залежною змінною (цільовою або показником інтересу) і однією або більше незалежними змінними (предикторами). У нашому випадку доцільнішим буде використання декількох незалежних змінних для точнішого прогнозування, нам знадобиться проаналізувати такі показники як: дохід, кількість транзакцій, типи витрат тощо. Такий підхід використання лінійної регресії називається множинна лінійна регресія [5]. Метод буде використаний для побудови моделі, що прогнозуватиме майбутні витрати клієнтів на основі історичних даних про транзакції, доходи та інші фінансові характеристики.

Розглянемо математичне представлення алгоритму. Лінійна регресія - це модель, що описати лінійну залежність між залежною змінною (y) та незалежним змінним (x_1, x_2, x_3 ...). Основна ідея полягає в тому, щоб побудувати модель, яка найкращим чином буде підходити до наданого набору даних [6].

Формула (2.1) моделі множинної лінійної регресії для використання у дослідженні (декілька незалежних змінних):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon \tag{2.1}$$

де y – залежна змінна, яку ми намагаємось прогнозувати,

x – незалежна змінні або предиктори, на основі яких ми робимо прогноз,

$\beta_1, \beta_2, \ldots, \beta_k$ – ваги предикторів визначають, як кожен предиктор впливає на залежну змінну,

β_0 – підставне значення або зсув (intercept) на вісі y, коли $x = 0$,

ϵ – помилка або похибка, яка відображає нев'язку моделі, тобто різницю між реальними та прогнозованими значеннями.

Значення параметрів $\beta_0, \beta_1, \beta_2, \ldots, \beta_k$ мінімізують відхилення між фактичними значеннями x і передбаченими значеннями $\hat{y}$. Для знаходження цих значень

використовується метод найменших квадратів (SSE – Sum of Squared Errors), який мінімізує суми квадратів залишкових похибок (сума квадратів різниць між фактичними та прогнозованими значеннями).

Математично метод можна описати за формулою (2.2):

$$SSE = \sum_{i=1}^{n} \epsilon_i^2 \tag{2.2}$$

Її можна переписати більше детальніше із використанням значень $\beta_0, \beta_1, \ldots, \beta_k$, які мінімізують функцію, формула (2.3):

$$S(\beta_0, \beta_1, \ldots, \beta_k) = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik}))^2 \tag{2.3}$$

Метод використовується для побудови лінійних моделей, які можуть прогнозувати значення величини на основі взаємозв'язку з іншими змінними. Після його виконання ми отримаємо певну прогнозуючу модель, яку можна використовувати у прогнозування потрібних даних, у контексті дослідження це витрати користувача.

Як приклад візуалізації роботи множинної лінійної регресії можна навести наступні графіки залежності розміру витрат від року та місяця (див. рис 2.1).

Сині точки на графіку з рисунку це фактичні дані датасету, де кожна точка представляє одне спостереження з трьома змінними: двома незалежними змінними (на осях X і Y) та однією залежною змінною (на осі Z). Червона зигзагоподібна лінія – це регресійна площина, яка показує прогнозовані значення залежної змінної (Z) на основі значень двох незалежних змінних (X і Y). Червона лінія показує, як змінюється прогнозоване значення залежної змінної при зміні незалежних змінних.

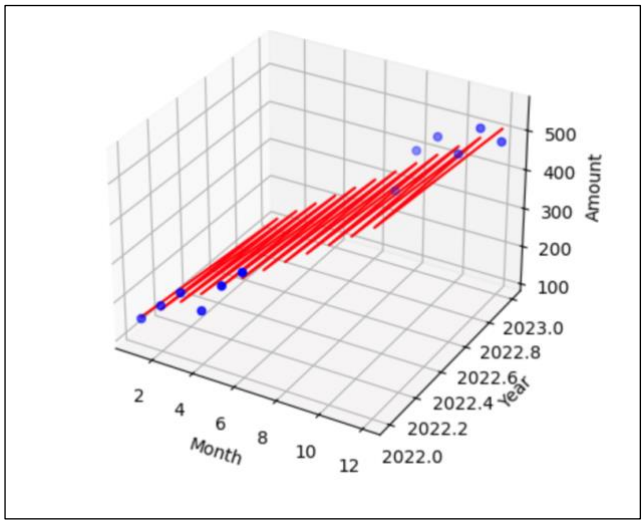


Рисунок 2.1 – 3D візуалізація залежностей (рисунок створено самостійно)

Далі розглянемо результати через візуалізацію на парних графіках (див. рис. 2.2).

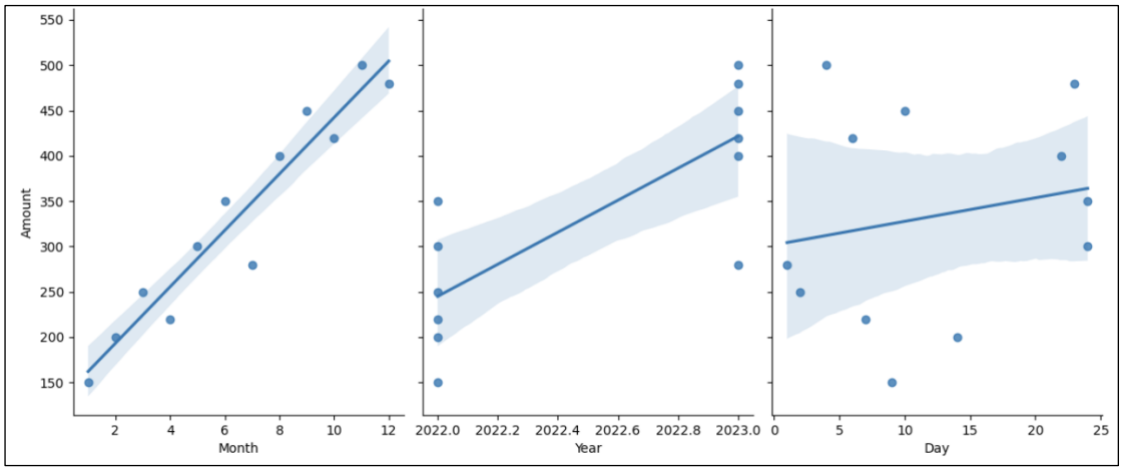


Рисунок 2.2 – Парні графіки залежностей (рисунок створено самостійно)

На графіках з рисунку можна побачити парні графіки, які показують взаємозв'язки між парами змінних у датасеті. Точки представляють окремі дані. Кожна точка на графіку відповідає одному рядку наданого датасету. Лінія на парному графіку показує лінійну регресію між двома змінними. Це означає, що лінія є найкращою прямою, яка приблизно проходить через точки, мінімізуючи відстані (залишки) між точками і лінією. Відстані між точками та лінією називаються

залишками (residuals). Вони представляють різницю між фактичними значеннями і прогнозованими значеннями на основі лінійної регресії.

Сильний кореляційний зв'язок – якщо лінія регресії добре відповідає точкам (точки близько до лінії), це свідчить про сильний лінійний зв'язок між змінними.

Слабкий кореляційний зв'язок – якщо точки сильно розкидані навколо лінії, це свідчить про слабкий або відсутній лінійний зв'язок.

Позитивна кореляція – якщо лінія регресії нахилена вгору, це свідчить про позитивну кореляцію.

Негативна кореляція – якщо лінія регресії нахилена вниз, це свідчить про негативну кореляцію.

2.3 Огляд методу Експоненційне згладжування

Експоненційне згладжування – це метод прогнозування тимчасових рядів, який надає більшу вагу більш свіжим спостереженням, поступово зменшуючи вагу старіших даних. Цей метод добре підходить для даних, які містять тренди або сезонні коливання.

Експоненційне згладжування працює шляхом обчислення зваженого середнього значення попередніх спостережень, де ваги зменшуються експоненційно з часом. Це означає, що більш нові спостереження отримують більше ваги, ніж старішіх [7].

Існує декілька видів цього методу, просте експоненційне згладжування та його розширена версія – метод Холта-Вінтерса.

Розглянемо для початку просте експоненційне згладжування. Така варіація підходить для даних без трендів або сезонних компонентів. Вона використовує одну параметричну константу α, яка визначає швидкість зменшення ваг.

Математичне представлення наведено за формулою (2.4):

$$S_t = a \cdot Y_t + (1 - a) \cdot S_{t-1} \quad (2.4)$$

де t – поточний період, який прогнозується,

S_t – прогнозоване значення на поточний період,

S_{t-1} – прогнозоване значення на минулий період,

a – згладжувальний параметр, що знаходиться в діапазоні $0 < a < 1$,

Y_t – фактичне спостереження в період t.

Такий метод підходить для ситуацій, коли є стабільний ряд без тренду і сезонності. В такому випадку метод ефективний для згладжування часового ряду, в якому немає очевидного тренду або сезонних коливань. Він підходить для даних, які коливаються навколо деякого середнього значення. Також такий варіант методу може бути використано для короткострокових прогнозів, де останні спостереження є більш важливими, ніж давніші. Як приклад такого використання можна навести щоденні продажі в магазині, де немає вираженого тренду або сезонності та короткострокові запаси товарів на складі [8].

Тепер давайте розглянемо детальніше розширену версію цього методу, його ще називають подвійним експоненційним згладжуванням. Цей метод підходить для часових рядів, які містять лінійний тренд. Він дозволяє враховувати і прогнозувати трендові зміни в даних. Також його можна використати у даних, які мають поступові зростання або зниження, але без сезонних коливань.

Математично метод розраховується у двох формулах. Розрахунок згладженого значення з урахуванням тренду наведено за формулою (2.5):

$$S_t = a \cdot Y_t + (1 - a) \cdot (S_{t-1} + T_{t-1}) \quad (2.5)$$

де T_t – оцінка тренду на момент часу t.

Розрахунок самого тренду наведено за формулою (2.6):

$$T_t = a \cdot (S_t - S_{t-1}) + (1 - a) \cdot T_{t-1} \quad (2.6)$$

Прикладом використання цього методу можуть бути щомісячні продажі, де спостерігається зростання або зниження, популяційні дані, що збільшуються з часом або маркетингові витрати, що мають свій тренд зростання чи зниження.

Тепер давайте розглянемо візуалізацію принципу роботи цього методу на прикладі прогнозування транзакційних даних людини. Для початку візуалізуємо початкові дані по з яких ми хочемо отримати прогнозовані результати (див. рис. 2.3).

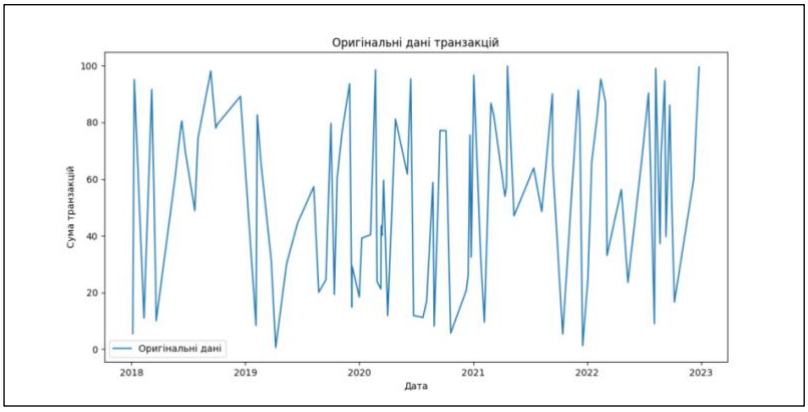


Рисунок 2.3 – Візуалізація початкових даних (рисунок створено самостійно)

Наступним кроком виконаємо прогнозування простим методом простого експоненційного згладжування (див. рис. 2.4).

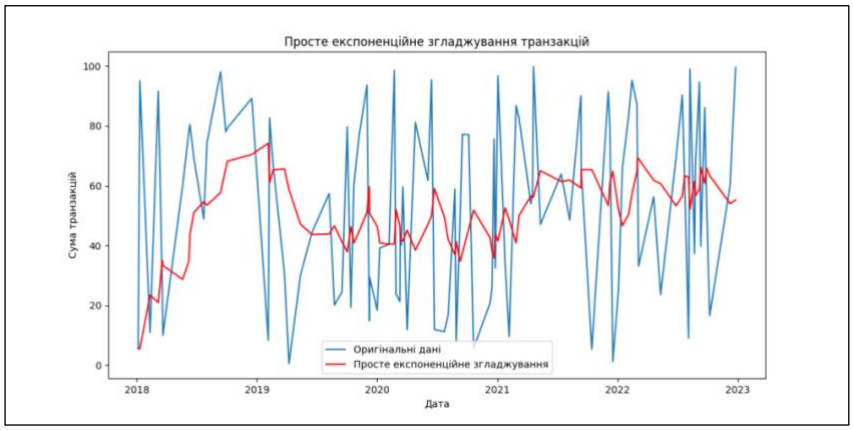


Рисунок 2.4 – Візуалізація результату виконання метода простого експоненційного згладжування (рисунок створено самостійно)

Червона лінія на рисунку представляє згладжені значення суми транзакцій, отримані за допомогою простого експоненційного згладжування. Цей метод добре підходить для даних без чіткого тренду.

Далі виконаємо прогнозування подвійним методом експоненційного згладжування (див. рис. 2.5).

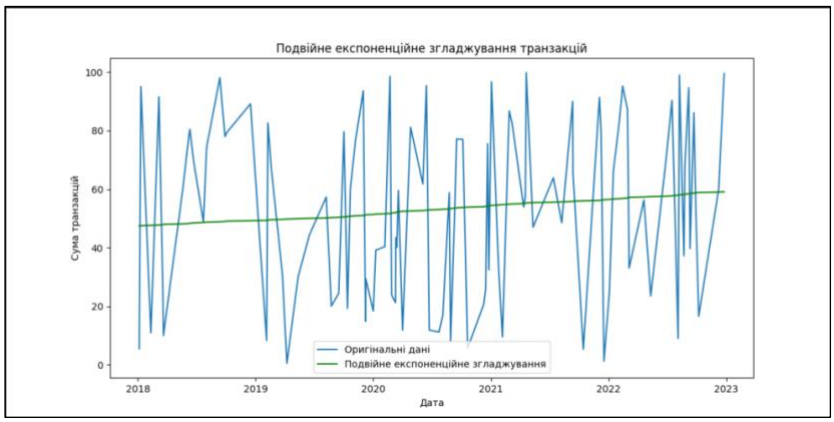


Рисунок 2.5 – Візуалізація результату виконання метода подвійного експоненційного згладжування (рисунок створено самостійно)

Зелена лінія представляє згладжені значення суми транзакцій, отримані за допомогою подвійного експоненційного згладжування. Цей метод враховує як тренд, так і згладжування коливань.

Останнім кроком ми порівняємо результати виконання обох методів, та побудуємо графік із об’єднаними результатами (див. рис. 2.6).

Рисунок 2.6 – Візуалізація об’єднаних результатів обох методів (рисунок створено самостійно)

На останньому графіку показані обидва методи для порівняння. На ньому можна побачити, як кожен метод поводиться з даними і який підходить краще для прогнозування витрат.

2.4 Огляд методу аналізу кластеризації даних K-means

Метод кластеризації даних – це процес групування схожих об'єктів разом у класи, які називаються кластерами. Основна мета полягає в тому, щоб об'єкти в одному кластері були більш схожими між собою, ніж з об'єктами з інших кластерів.

Одним з найпоширеніших методів кластеризації є метод k-середніх (K-Means Clustering). Він розділяє набір даних на k кластерів, де к - це попередньо визначена кількість кластерів [9].

Алгоритм K-Means використовується для мінімізації внутрішньо-кластерного варіанту і максимізації між-кластерного варіанту. Це досягається шляхом початкової випадкової ініціалізації k центрів кластерів, призначення кожного об'єкта до найближчого центру, перерахунку центрів кластерів як середнього розташування всіх призначених об'єктів, та повторення цих кроків до збіжності.

Принцип роботи методу можна також описати наступним чином:

- вибір кількості кластерів (K) – користувач задає кількість кластерів K, на які потрібно розбити дані;
- ініціалізація центроїдів – вибираються K випадкових точок як початкові центроїди кластерів;
- призначення точок кластерам – кожна точка даних призначається до найближчого центроїда, утворюючи кластери;
- оновлення центроїдів – центроїди обчислюються знову як середнє значення точок, що належать до кожного кластеру;
- повторення кроків 3 і 4 – кроки 3 і 4 повторюються до тих пір, поки центроїди не перестануть змінюватись (тобто, поки алгоритм не збіжиться).

Математично ці кроки описуються так:

– формування проблеми. Нехай $x = \{x_1, x_2, \dots, x_n\}$ – набір точок даних, де кожна точка $x_i \in R^2$. R – загальна кількість кластерів, яка оцінюється. Потрібно розбити точки на K кластерів, $C = \{C_1, C_2, \dots, C_k\}$, де C – набір кластерів, що включає, кожен за яких містить точки даних, які відносяться до цього кластера;

– цільова функція методу. Мінімізувати суму квадратів відстаней між точками і їхніми відповідними центроїдами, розрахунки проводяться за формулою (2.7):

$$J = \sum_{k=1}^{k} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \tag{2.7}$$

де J – значення цільової функції або функції витрат, яку алгоритм K-Means намагається мінімізувати,

μ_k – центроїд кластеру C_k.

– призначення точок кластерам. Кожна точка призначається до кластеру з найближчим центроїдом по формулі (2.8):

$$C^k = \left\{x_i : \|x_i - \mu_k\|^2 \leq \left\|x_i - \mu_j\right\|^2 \ \forall_j, 1 \leq j \leq K\right\} \tag{2.8}$$

– Оновлення центроїдів. Центроїд кожного кластеру оновлюється як середнє значення точок, що належать до цього кластеру:

$$\mu_k = \frac{1}{|C|} \sum_{x_i \in C_k} x_i \tag{2.9}$$

Далі ми вже отримаємо результат роботи методу, де ми бачимо серед точок даних створені кластери (точки різних кольорів) та червоні хрестики, що ілюструють центроїди цих кластерів (див. рисунок 2.3).

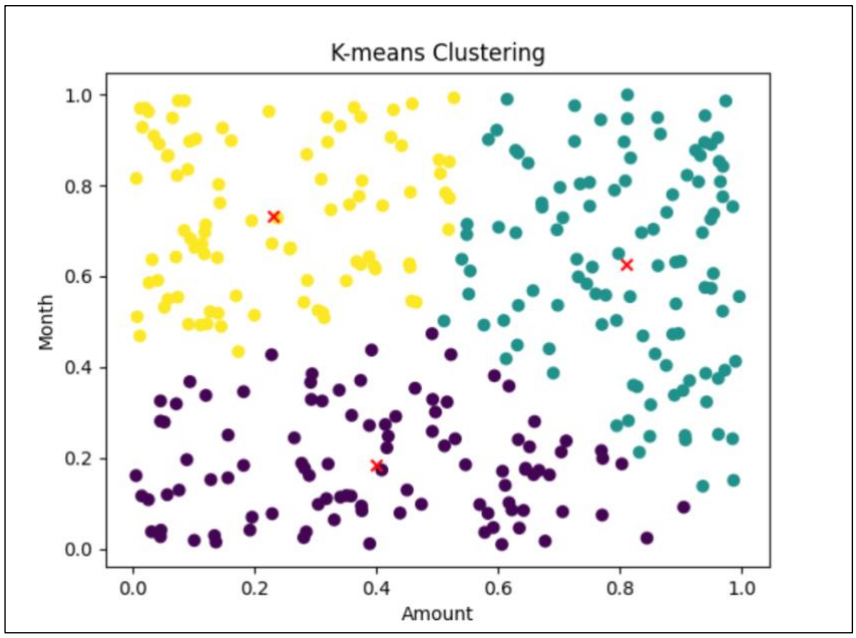


Рисунок 2.7 – Візуалізація результату методу K-means (рисунок створено самостійно)

2.5 Порівняльна характеристика за принципом Парето

Нашою метою є визначення найбільш оптимального метод аналізу даних для роботи фінансових даних по транзакціям з використанням багатокритеріального підходу, де кожен критерій відображає певний аспект корисності аналізу для користувача.

Критерії прийняття рішень:

- точність та ефективність – якість результатів, що отримуються від кожного методу, наприклад, чи забезпечують ці методи достатньо точний прогноз фінансових транзакцій;
- складність реалізації та використання – оцінює час та ресурси, потрібні для розгортання та підтримки кожного методу;
- масштабованість та адаптивність – перевіряє здатність методів

адаптуватися до змін у фінансових даних та їх обсягу;

– швидкість обробки даних – виявляє швидкість роботи, залежно від об’єму даних.

Обрані методи для порівняння: лінійна регресія, експоненційне згладжування, кластерний аналіз

Використання алгоритму лінійної регресії. Цей метод може бути застосований для прогнозування тенденцій чи визначення зв'язків між фінансовими даними, наприклад, прогнозування майбутніх транзакцій або встановлення залежностей між факторами та фінансовими результатами.

Використання алгоритму експоненційного згладжування. Цей метод допомагає виявити патерни чи тренди у часових рядах, зокрема, в аналізі часових рядів фінансових транзакцій.

Використання кластерного аналізу методом k-means. Цей метод може розділити транзакції на групи або кластери зі схожими характеристиками, що може допомогти в ідентифікації певних патернів або особливостей між різними групами транзакцій задля точнішого прогнозування.

Створимо шкали оцінювання для обраних альтернатив (див. табл. 2.1).

Таблиця 2.1 – формування школи оцінювання обраних алгоритмів (таблиця виконана самостійно)

	Використання лінійної регресії	*Використання експоненційного згладжування*	*Використання кластеризації даних*
Точність та ефективність	f_1x_1	f_1x_2	f_1x_3
Складність реалізації та використання	f_2x_1	f_2x_2	f_2x_3
Масштабованість та адаптивність	f_3x_1	f_3x_2	f_3x_3
Швидкість обробки даних	f_4x_1	f_4x_2	f_4x_3

Для нормалізації даних, за якими перевіряються алгоритми, використовується метод лінійної адитивної згортки з нормуючими множниками, який наведено за формулою (2.10):

$$x_{norm} = \frac{x - min\ (x)}{max(x) - min\ (x)} \tag{2.10}$$

де x_{norm} – нормалізоване значення,

x – початкове значення, яке потрібно нормалізувати,

$min(x)$ – мінімальне значення в наборі x,

$max(x)$ – максимальне значення в наборі x.

Цей метод також має векторний опис, його наведено у формулі (2.11):

$$U(x) = \max\ f_1(x) + f_2(x) + f_3(x) + f_4(x) \tag{2.11}$$

де $U(x)$ – нормалізоване значення, що об'єднує кілька показників,

$f_i(x)$ – функція корисності для i – го критерія.

Спочатку запишемо відповідні значення до обраних методів (див. табл. 2.2).

Таблиця 2.2 – Відповідні значення до обраних методів (таблиця виконана самостійно)

	Використання лінійної регресії	*Використання експоненційного згладжування*	*Використання кластеризації даних*
Точність та ефективність	87%	79%	92%
Складність реалізації та використання	Ні	Ні	Ні
Масштабованість та адаптивність	Так	Так	Так
Швидкість обробки даних	4 секунд	7 секунд	3 секунди

Наступним кроком ми нормалізуємо значення за вище наведеною формулою (див. табл. 2.3).

Таблиці 2.3 – Нормалізовані значення даних (таблиця виконана самостійно)

	Використання лінійної регресії	*Використання експоненційного згладжування*	*Використання кластеризації даних*
Точність та ефективність	0,84	0,72	0,94
Складність реалізації та використання	0	0	0
Масштабованість та адаптивність	1	1	1
Швидкість обробки даних	0,88	0,61	0,91

Принцип Парето показує, що загалом всі методи показують гарний результат. Ми бачимо, що експоненційне згладжування дає найменш ефективні результати, а 2 інших алгоритми показують гарний довільний результат. І якщо казати про важливість критерію у фінансовій сфері, точність оброки даних та швидкість має більший пріоритет.

Далі розпишемо фінальні результати та просумуємо їх (див. табл. 2.4).

Таблиця 2.4 – Фінальні складені результати (таблиця виконано самостійно)

	Використання лінійної регресії	*Використання експоненційного згладжування*	*Використання кластеризації даних*
Точність та ефективність	0,83	0,79	0,89
Складність реалізації та використання	1	0	1

Масштабованість та адаптивність	1	1	1
Швидкість обробки даних	0,91	0,61	0.94
Сума	3,74	2,4	3,83

Таким чином можна побачити, що найбільш відповідними до вимог дослідження та створення програмної системи є алгоритми лінійної регресії та кластерного аналізу. Вони добре підійдуть до вирішення задачі категоризації та прогнозування витрат.

2.6 Обґрунтування обраних методів аналізу для дослідження

Для роботи було обрано два найрозповсюджених методи аналізу даних у сфері банківського фінансового менеджменту: лінійна регресія та кластеризація даних методом k-means. Методи були обрані відштовхуючись від корисності однієї із найважливіших сфер банківського фінансового менеджменту – прогнозування витрат. Саме ця сфера є дуже важливою у будь-якій фінансовій установі, бо завдяки ній банки можуть вираховувати ризики по кредитах, планувати власний бюджет, допомагати планувати бюджет своїм клієнтам і напряму впливати на якість фінансових витрат – тобто, наскільки ефективно можна витратити або навпаки зберегти певний бюджет. Методи лінійної регресії та кластеризації чудово накладаються на такі цілі за свої певні ознаки.

Інноваційність обраних методів для проведення дослідження може бути охарактеризована декількома аспектами. По-перше, незважаючи на те, що окреме використання кластеризації даних та лінійної регресії є добре дослідженим, їх інтеграція для прогнозування та аналізу банківських транзакцій є достатньо новим та ще непоширеним підходом. По-друге, використання кластеризації надає можливість виявлення поведінкових патернів – створені групи-кластери із даних зі схожими характеристиками надають нову інформацію стосовно фінансових звичок

користувачів. Ця інформація в свою чергу може бути використана для розвитку теорій поведінкових фінансів та економіки. По-третє, аспект поєднання методів аналізу на практиці, для створення згрупованого прогнозованого аналізу у банківських додатках є доволі інноваційним на даний момент, оскільки сучасні банківські системи активно шукають способи впровадження методів аналізу для досягнення корисних цілей.

Лінійна регресія має простоту та інтерпретованість, вона є одним із найрозповсюджених і найзрозуміліших алгоритмів машинного навчання. Метод дозволяє легко інтерпретувати взаємозв'язки між незалежними змінними (наприклад, місяць, категорія витрат тощо) та залежною змінною (сума витрат). Лінійна регресія базується на добре розроблених математичних основах, таких як метод найменших квадратів, що забезпечує мінімізацію помилок прогнозу, також метод добре підходить для прогнозування кількісних змінних. У контексті дослідження метою метода є виявлення зв'язків між фінансовими показниками та ідентифікація факторів, що впливають на них. Він дозволить побудувати модель на основі минулих даних, яка описує лінійні залежності між різними змінними та здійснювати прогнози майбутніх витрат опираючись на цю модель.

Кластеризація даних також є дуже розповсюдженим та зрозумілим інструментом аналізу. Метод допомагає знаходити схожі дані та групувати їх у кластери. Це дозволяє виявити загальні тренди та поведінкові патерни витрат користувача. На основі цих кластерів можна створити більш точні та персоналізовані рекомендації щодо оптимізації бюджету, оскільки ми знатимемо, які категорії витрат найбільш актуальні для користувача. Також кластеризація добре накладається на сценарій її використання в комбінування із лінійною регресією, бо вона зменшує складність аналізу великих обсягів даних, розбиваючи їх на більш керовані групи. Таке групування у контексті дослідження може використовуватися у транзакціях для виявлення патернів витрат користувача та покращення розуміння структури цих витрат. Також це дає нам можливість ідентифікувати сегменти

операцій зі схожими властивостями для подальшого вивчення та управління.

Поєднання цих двох методів дозволяє досягти покращення точності прогнозів за рахунок більш комплексного аналізу. Це дає змогу отримати не лише загальну суму майбутніх витрат, але й розподілити їх за категоріями, що дозволяє більш детально спланувати бюджет.

Таким чином, вибір лінійної регресії для прогнозування та кластеризації даних для виявлення патернів витрат є оптимальним підходом для досягнення цілей дослідження.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис логіки роботи програмної системи

Програмна система на практиці повинна аналізувати історичні дані по транзакціях користувачів і створювати згрупований прогноз витрат на певний період часу у майбутньому, у випадку мого дослідження це буде наступний місяць. Результатом роботи програмної системи – є групи (кластери) витрат, об’єднані за схожими ознаками. Цим групам надана описова характеристика відштовхуючись від схожих ознак транзакцій в цьому кластері. Користувач, отримуючи прогнозовану структуровану аналітику витрат на майбутній період, має можливість використовувати її для оптимізації свого фінансового планування.

Принцип роботи програмної системи полягає в тому, що вона отримує на вхід датасет із даними о транзакціях у вигляді csv файлу, після цього файл завантажуються до системи, дані перевіряються і зберігаються у базі даних. Потім, у користувача з’являється можливість отримувати прогнозовану аналітику по цим даним – транзакції формуються у такий новий csv файл, передаються до сервісу python, де відбувається прогнозування і структуризація, і генеруються результат аналізу у вигляди прогнозованих даних, розбитих на групи-кластери. Ці дані передаються назад до java сервісу і вже можуть бути виведені на клієнті для користувача.

3.2 Вимоги до веб-застосунку

Користувач повинен мати наступні можливості:

- створювати транзакції та об’єднувати їх у спеціальні групи для зручного переглядання;
- отримувати автоматичну категоризацію транзакцій;
- отримувати статистику витрат по категоріям;
- отримувати індивідуальний аналіз та прогнозування витрат для планування

оптимізації витрат.

Клієнтський веб-застосунок повинен виконувати наступний функціонал:

– реєстрація та вхід користувачів;
– редагування профілю;
– перегляд транзакцій та груп транзакцій;
– редагування даних користувача (видалення, оновлення, створення);
– отримання аналітики по фінансовим транзакціям відповідно до бажання користувача;

Веб-застосунок повинен бути реалізований за допомогою сучасних інструментів для розробки сайтів React.js, Redux (для керування станом аплікації), Axios (для роботи з запитами до серверу), HTML5, CSS3, Bootstrap.

3.3 Вимоги до серверного застосунку

Серверна частина повинна реалізовувати наступний функціонал:

– реєстрація та авторизація користувачів;
– отримання даних від клієнтів;
– захист даних та шифрування;
– відправлення даних на клієнтську сторону;
– сортування даних;
– видалення, редагування та створення даних;
– створення транзакцій, груп транзакцій;
– аналіз даних по транзакціям користувача за допомогою методу кластеризації
– аналіз даних по транзакціям користувача за допомогою методу лінійної регресії;
– прогнозування витрат на наступний проміжок часу;
– структуроване відправлення даних залежно від аналітики;
– керування обліковими записами користувачів;

– обробка даних;

– валідація даних;

– конфігурація пристрою;

Серверний застосунок буде реалізовано за допомогою мови Java за залученням Spring Framework та мови Python із бібліотекою Pandas. Застосунок буде мікросервесним та складатиметься із одного сервісу на Java та сервісу на Python. На сьогодні такий підхід до архітектури застосунку вважається одним із найрозповсюдженіших та перевірених підходів до розробки систем із подібним функціоналом. База даних буде реалізована на сервері PostgreSQL.

3.4 UML проектування ПЗ

Першим кроком в проектування програмного забезпечення було створення Use Case діаграми, яка б відображала користувачів системи та дії, які вони можуть виконувати [10].

Діаграму Use Case можна побачити на рисунку 3.1.

Як видно з приведеної вище діаграми, користувач може зареєструватись в системі, або авторизуватись, якщо вже має акаунт. Також користувач може керувати транзакціями та групами, тобто він може створювати, змінювати або видаляти перелічені вище елементи. Користувач може отримувати потрібну аналітику транзакцій – побачити категорії витрат, сформувати аналіз по витратам, отримати прогнозування витрат. Користувач має можливість керувати профілем – змінювати стандартні дані або пароль.

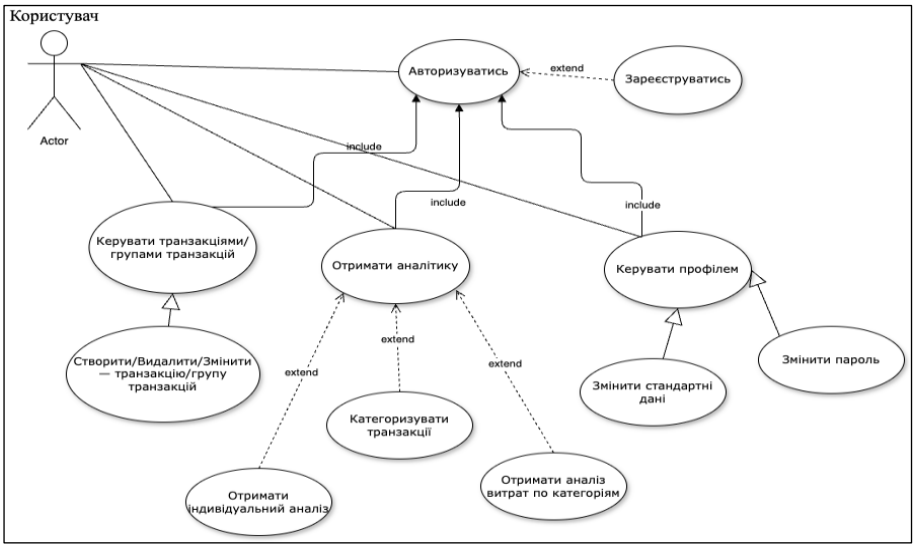


Рисунок 3.1 – Use Case діаграма користувача (рисунок створено самостійно)

Іншою діаграмою, яку було розроблено в процесі проектування системи, є діаграма діяльності, яку приведено на рисунку 3.2 [11]. Вона відображає увесь шлях, який може пройти користувач в системі від початку до кінця взаємодії, охоплюючи усі сценарії.

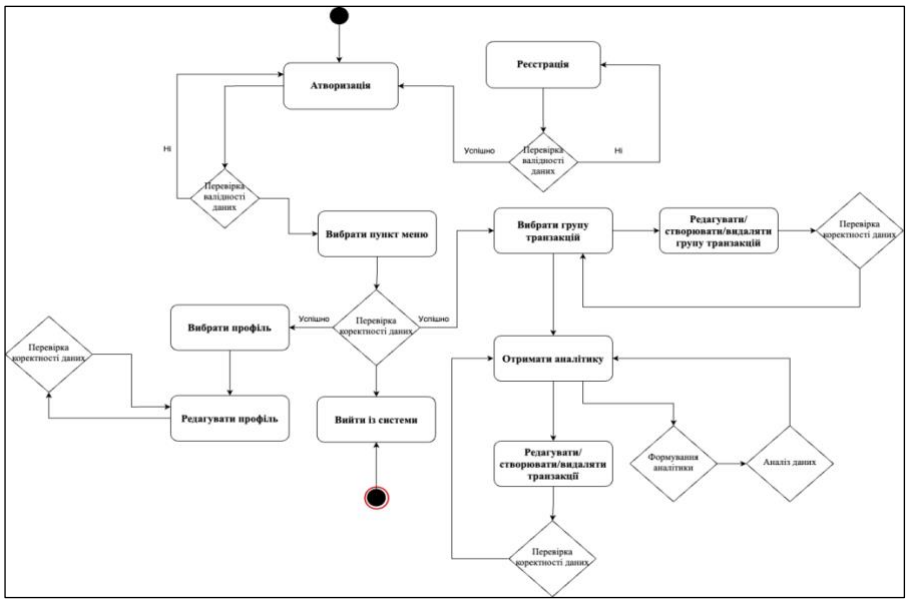


Рисунок 3.2 – Діаграма діяльності (рисунок створено самостійно)

3.5 Проєктування архітектури ПЗ

Серверна частина побудована за допомогою мови програмування Java із

залученням фреймворку Spring Framework та мови Python із використанням бібліотеки pandas.

Для більш детального розуміння архітектури системи можна подивитися на діаграму розгортання [12], яка приведена на рисунку 3.3.

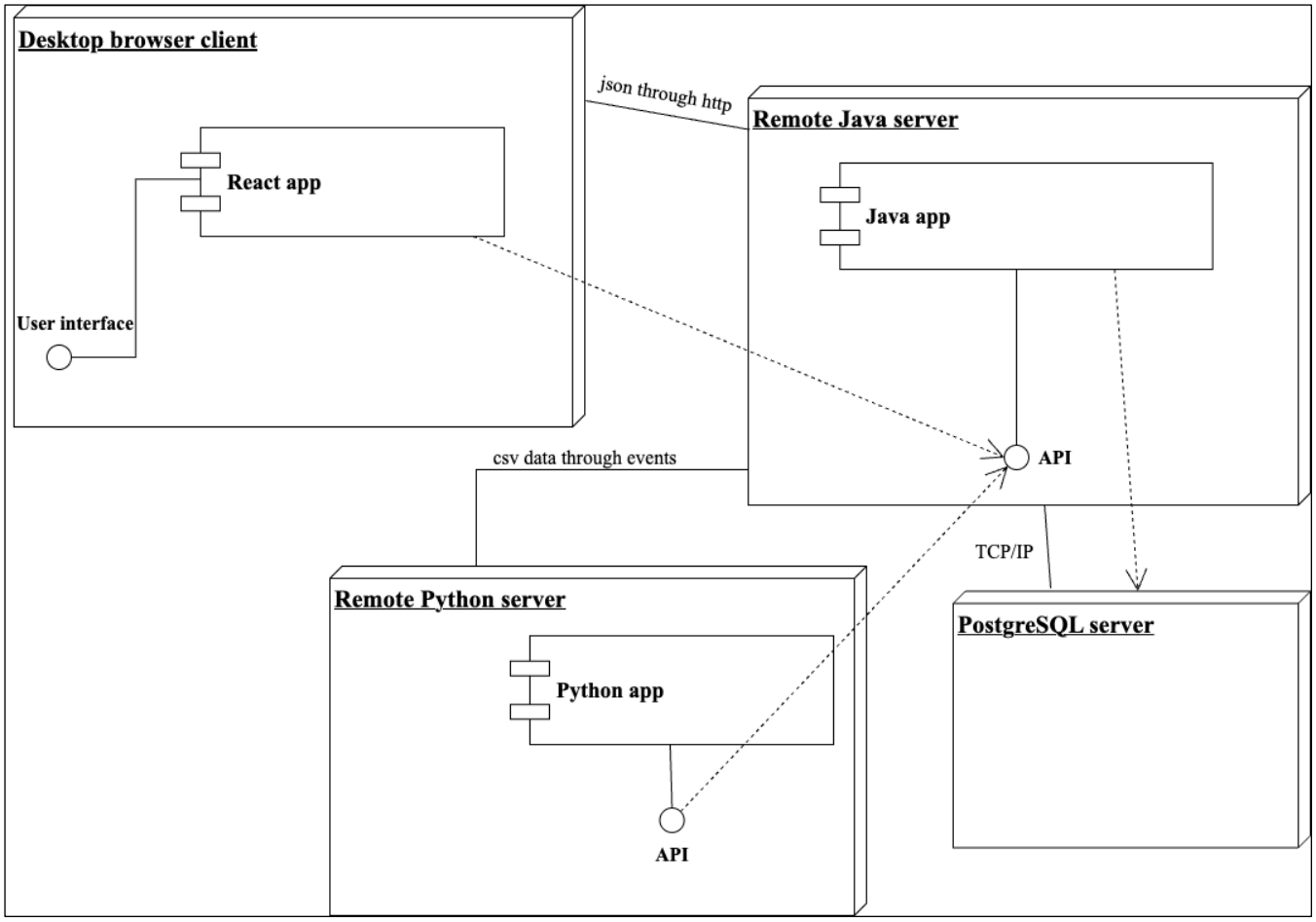


Рисунок 3.3 – Діаграма розгортання (рисунок створено самостійно)

Діаграма розгортання визначає фізичне обладнання, на якому працюватиме програмна система. Він також визначає спосіб розгортання програмного забезпечення на базовому обладнанні. Він прив’язує частини програмного забезпечення системи до пристрою, який збирається її виконувати.

Діаграма розгортання відображає програмне забезпечення та архітектуру системи, створену в процесі розробки. У розподілених системах він моделює розподіл програмного забезпечення між фізичними вузлами. На цій діаграмі видно

на які саме модулі розгортується програмна система та як ці модулі будуть взаємодіяти між собою.

3.6 Проєктування бази даних

Для бази даних в проекті використовується СКБД PostgreSQL.

PostgreSQL, також відома як Postgres, – це система управління реляційною базою даних, розробники якої наголошують на розширюваності та відповідності стандартам. Він бере свій початок у 1986 році як частина проекту POSTGRES Університету Берклі і розроблявся з основною платформою більше 30 років [13].

PostgreSQL відповідає вимогам ACID і є транзакційною. PostgreSQL має оновлені представлення, реалізовані уявлення, тригери та зовнішні ключі; підтримує функції та збережені процедури та інші можливості розширення.

PostgreSQL була розроблена глобальною командою розробників PostgreSQL, різноманітною групою компаній та окремих осіб [14]. Це безкоштовний і відкритий вихідний код, випущений під ліцензією безкоштовного програмного забезпечення за умовами PostgreSQL.

ER-діаграму розроблюваної програмної системи приведено на рисунку 3.4.

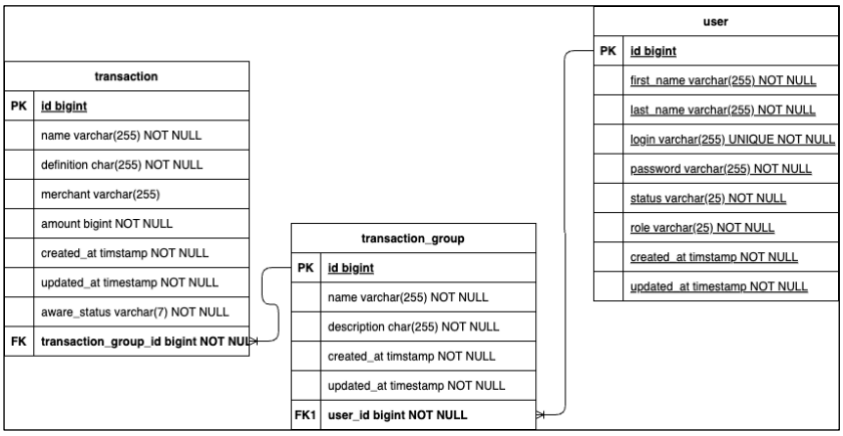


Рисунок 3.4 – ER-діаграма (рисунок створено самостійно)

Таким чином база даних сервісу побудована на 3 таблицях, які містять усі необхідні дані для функціонування сервісу: таблиця для транзакцій, груп та користувачів.

3.7 Створення UI дизайну системи

Під час створення дизайну було приділено особливу увагу інтерфейсу користувача, який повинен бути мінімалістичним та достатньо зрозумілим, щоб користувач міг повноцінно користуватися системою.

На рисунку 3.5 наведено приклад сторінки перегляду транзакцій.

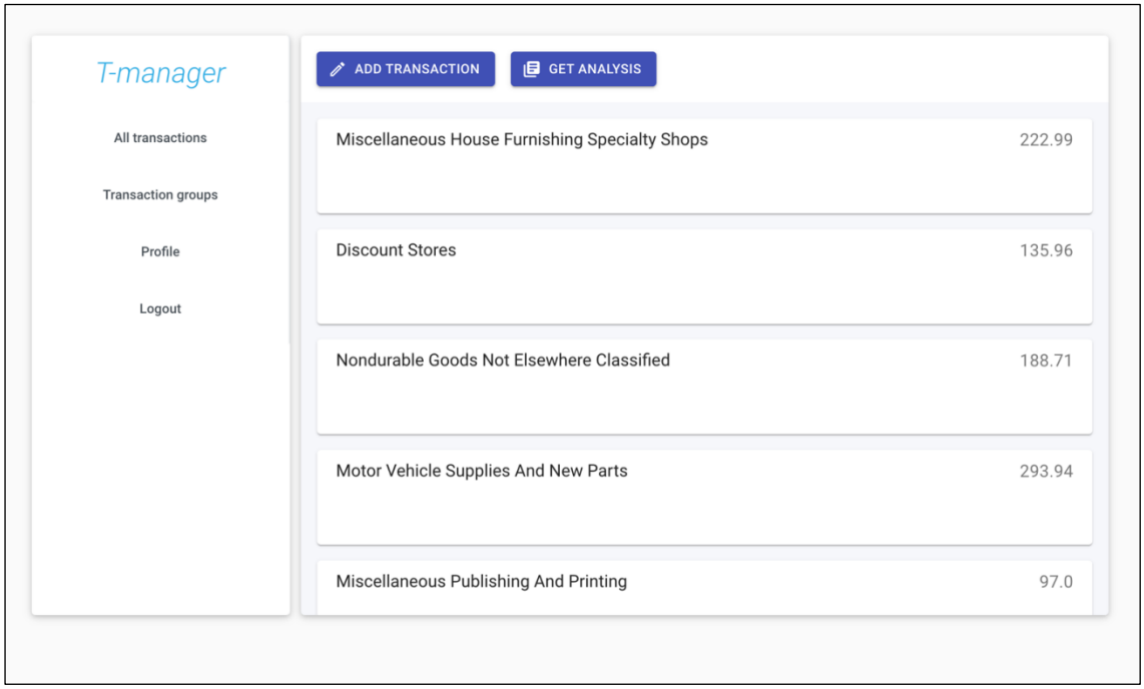


Рисунок 3.5 – Приклад дизайну сторінки групи транзакцій (рисунок створено самостійно)

Користувач переглядає світлу форму, на якій усі необхідні частини логічно виділені окремими формами, а функціональні кнопки мають відділяючий колір.

4 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ДОСЛІДЖЕННЯ

4.1 Обрані технології

Під час розробки програми використовувалися наступні технології:

- середовище розробки Intelij Idea;
- мова програмування Java;
- мова програмування Python;
- бібліотека алгоритмів аналізу Pandas;
- Spring Framework;
- React.js;
- СКБД PostgreSQL Server.

Нижче розібрано технології, що використані для розробки системи, та приведено фрагменти коду програмного забезпечення.

4.2 Серверна частина

Java – мова програмування та обчислювальна платформа, вперше випущена Sun Microsystems у 1995 році. Вона розвинулась із скромних початків до забезпечення великої частки сучасного цифрового світу, забезпечуючи надійну платформу, на якій побудовано багато послуг і програм [15]. Нові інноваційні продукти та цифрові послуги, розроблені для майбутнього, також продовжують використовувати мову програмування Java.

У той час як більшість сучасних програм Java об’єднують середу виконання Java та програму разом, все ще існує багато програм і навіть деякі веб-сайти, які не працюватимуть, якщо на комп’ютері не встановлено Java для настільного комп’ютера. Сайт Java.com призначений для споживачів, яким все ще може знадобитися Java для настільних додатків, зокрема програм, орієнтованих на Java 17 [16].

Spring Framework (Spring) – програма з відкритим вихідним кодом, яка забезпечує підтримку інфраструктури для розробки додатків Java. Один із найпопулярніших фреймворків Java Enterprise Edition (Java EE), Spring допомагає розробникам створювати високопродуктивні програми за допомогою простих старих об’єктів Java [17].

Даний фреймворк – це великий набір попередньо визначеного коду, до якого розробники можуть додати код для вирішення проблеми в певній області [18]. Існує багато популярних фреймворків Java, включаючи Java Server Faces (JSF), Maven, Hibernate, Struts і Spring.

Python та Pandas – відмінно підходять для виконання задач data analysis. Python має простий синтаксис, що дозволяє швидко написати та читати код, навіть для складних операцій з даними [19]. Pandas надає багатофункціональні інструменти для роботи з табличними даними, включаючи зручні методи для фільтрації, агрегації та маніпуляції даними. Python легко інтегрується з іншими бібліотеками для аналізу даних, такими як NumPy, SciPy, scikit-learn та Matplotlib, що робить його потужним інструментом для комплексного аналізу даних [20]. Pandas оптимізований для роботи з великими обсягами даних і надає можливості швидкої обробки та аналізу. Також існує велика спільнота користувачів та розробників Python забезпечує широкий доступ до документації, прикладів коду та готових рішень для різноманітних задач.

PostgreSQL – це потужна система об’єктно-реляційних баз даних з відкритим вихідним кодом, яка використовує та розширює мову SQL у поєднанні з багатьма функціями, які безпечно зберігають і масштабують найскладніші робочі навантаження даних [21]. Витоки PostgreSQL сягають 1986 року в рамках проекту POSTGRES в Каліфорнійському університеті в Берклі і має понад 30 років активної розробки на базовій платформі.

PostgreSQL заслужив міцну репутацію завдяки своїй перевіреній архітектурі, надійності, цілісності даних, надійному набору функцій, розширюваності та

відданості спільноти відкритих вихідних кодів, які стоять за програмним забезпеченням, щоб постійно надавати продуктивні та інноваційні рішення. PostgreSQL працює на всіх основних операційних системах, має ACID-сумісність з 2001 року і має потужні доповнення, такі як популярний розширювач геопросторової бази даних PostGIS. На сьогоднішній день PostgreSQL – реляційна база даних з відкритим кодом, яку вибирають багато людей та організацій [22].

Використовуючи приведені вище технології, було розроблено серверну частину програмного застосунку.

4.3 Клієнтська частина

React (також відомий як React.js або ReactJS) – бібліотека Javascript з відкритим кодом, призначена для створення інтерфейсів користувача з метою полегшення розробки односторінкових програм [23]. Його підтримують Facebook та спільнота вільного програмного забезпечення. У проекті більше тисячі безкоштовних розробників.

React намагається допомогти розробникам створювати програми, які використовують дані, які постійно змінюються. Його мета – бути простим, декларативним і легко поєднуваним. React обробляє лише інтерфейс користувача в додатку; React – це View в контексті, що використовує шаблон MVC (Model-View-Controller) або MVVM (Model-View-View-Model) [24]. Його також можна використовувати з розширеннями на основі React, які піклуються про частини веб-програми, які не є користувацьким інтерфейсом.

Згідно з аналітичною службою JavaScript Libscore, React зараз використовується на перших сторінках Imgur, Bleacher Report, Feedly, Airbnb, SeatGeek, HelloSign тощо [25].

Творцем React є розробник програмного забезпечення Facebook Джордан Вальке. Ідея виникла з фреймворку XHP, який є основою для компонентів HTML для PHP. Вперше він був представлений на Facebook у 2011 році, а пізніше на

Instagram.com у 2012 році. Вихідний код був опублікований на конференції JSConf в США в травні 2013 року [26].

Використовуючи технологію React.js було розроблено клієнтську частину додатку.

Для відображення групи транзакцій слід було розробити логіку їх відображення. За це відповідає наступний фрагмент коду.

4.4 Реалізація програмної системи

Першим кроком до реалізації програмної системи є визначення даних, за якими буде проводитися аналіз. Оскільки в контексті дослідження потрібно досліджувати методи, то це доцільно робити на великих обсягах даних, в такому випадку самостійно створити правильну базу таких даних не є доцільним, тому правильним рішенням буде використовувати вже існуючий датасет, його можна найти на ресурсі kaggle.com [27].

Я обрав датасет із майже 800 000 даними по транзакціях користувачів, який виглядає наступним чином (див. рис. 4.1).

```
Year,Month,Department,Division,Merchant,TranxDescription,TranxDate,TrnxAmount
2019,1,LEGISLATIVE BRANCH,General Assembly House,USPS PO 0917600901,Postal Services-Government Only,06/29/2018,9.85
2019,1,LEGISLATIVE BRANCH,General Assembly House,GAN*NEWSPAPER SUB1052,Direct Marketing-Continuity-Subscription Merchants,07-03-2018,
2019,1,LEGISLATIVE BRANCH,General Assembly House,DS SERVICES STANDARD COFF,Nondurable Goods Not Elsewhere Classified,07-01-2018,170.3
2019,1,LEGISLATIVE BRANCH,General Assembly House,BETHANY BLUES - LEWES,Eating Places Restaurants,06/30/2018,3427.8
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,06/27/2018,113
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,06/27/2018,115
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,06/27/2018,159
2019,1,LEGISLATIVE BRANCH,General Assembly House,QUILL CORPORATION,Stationery-Office Supplies-Printing Writing Pap,07-04-2018,51.7
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,06/27/2018,167
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,06/27/2018,114
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,06/27/2018,111
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,06/27/2018,113
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,07-12-2018,-19
2019,1,LEGISLATIVE BRANCH,General Assembly House,XEROX CORPORATION/RBO,Office Photographic Photocopy Microfilm Equipmt,07-12-2018,-27
2019,1,LEGISLATIVE BRANCH,General Assembly House,MR. NATURAL BOTTLED W,Nondurable Goods Not Elsewhere Classified,07/25/2018,28.25
2019,1,LEGISLATIVE BRANCH,General Assembly House,VZWRLSS*APOCC VISB,Telecom Incl Prepaid-Recurring Phone Svcs,07/14/2018,54.63
2019,1,LEGISLATIVE BRANCH,General Assembly House,FACEBK LW245FWR32,Advertising Services,06/30/2018,5.12
2019,1,LEGISLATIVE BRANCH,General Assembly House,QUILL CORPORATION,Stationery-Office Supplies-Printing Writing Pap,07-07-2018,71.92
2019,1,LEGISLATIVE BRANCH,General Assembly House,QUILL CORPORATION,Stationery-Office Supplies-Printing Writing Pap,07-12-2018,45.87
2019,1,LEGISLATIVE BRANCH,General Assembly House,SNAPFISH US,Photo Developing Photofinishing Laboratories,07/27/2018,281.39
```

Рисунок 4.1 – Приклад обраного датасету (рисунок створено самостійно)

Як ми бачимо для нашої системи датасет має всі необхідні дані

– Year, Month – це рік та місяць зробленого запису;

– Department, Division – це розділ та підрозділ, в якому була зроблена

транзакція;

– Merchant – це виконавець транзакції;
– TranxDescription – це опис транзакції;
– TranxDate – безпосередньо дата транзакції;
– TrnxAmount – сума транзакції;

Далі нам потрібно попередньо перевірити ці данні на наявність некоректних або відсутніх значень, прибрати зайві колонки та створити оновлену версію датасету, перед цим зберігши потрібну кількість даних до нашої бази, щоб потім маніпулювати ними. Для цих цілей ми використаємо java сервіс, який зчитуватиме та оновлюватиме дані, приклад коду наведено нижче:

```
    public List<CsvTransaction> getCleanValues() {
    var values = csvRepository.readValues("credit-card-transactions")
        .filter(it -> it.getAmount() > 0)
        .filter(it -> Objects.nonNull(it.getDate()))
        .filter(it ->
it.getDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate().getYear
() > 2021)
        .sorted(Comparator.comparing(CsvTransaction::getDate))
        .toList();
    logUpdatedValues(values);
    return values;
}
```

Після проведення попереднього оновлення даних, наш датасет виглядає наступним чином (див. рис. 4.2).

З цього прикладу можна побачити новий формат даних. Датасет також було зменшено у розмірі, через те, що він надавав забагато даних для тестування. Вибірка була обмежена із 2018 до 2022 року та вище. Нові колонки мають наступні значення:

– year, month, day – відокремлені числові значення року, місяця та дня:
– merchant – виконавець транзакції;
– description – опис транзакції;
– amount – сума транзакції.

	year	month	day	merchant	description	amount
1	2022	1	1	SIGNUPGENIUS	Organizations Charitable And Social Services	24.99
2	2022	1	1	WATER - COFFEE DELIVERY	Nondurable Goods Not Elsewhere Classified	41.87
3	2022	1	1	MIDTOWN PARKING	Automobile Parking Lots And Garages	2794.0
4	2022	1	1	EIG*CONSTANTCONTACT.COM	Direct Marketing-Continuity-Subscription Merchants	204.0
5	2022	1	1	SECURE STORAGE LLC	Public Warehousing-Farm Refrig Goods Hhg Storage	179.0
6	2022	1	1	ALERTRA INC	Direct Marketing-Other Direct Marketers-Not Elsew	155.6
7	2022	1	1	INSTITUTE FOR INTERNAL CO	Organizations Membership-Not Elsewhere Classified	150.0
8	2022	1	1	INDEED	Direct Marketing-Other Direct Marketers-Not Elsew	504.7
9	2022	1	1	NYTIMES*NYTIMES DISC	Direct Marketing-Continuity-Subscription Merchants	4.0
10	2022	1	1	DROPBOX M554NJ6531D3	Computer Network-Information Services	75.0
11	2022	1	1	NYTIMES*NYTIMES DISC	Direct Marketing-Continuity-Subscription Merchants	4.0
12	2022	1	1	DOJ Internal Transaction	Nondurable Goods Not Elsewhere Classified	7.99
13	2022	1	1	EXTENDEDSTAY #9630	Lodging	947.94
14	2022	1	1	EXTENDEDSTAY #9630	Lodging	957.94
15	2022	1	1	BJS WHOLESALE #0015	Wholesale Clubs	179.76
16	2022	1	1	FAIRFIELD INN And SUITES	Lodging	1538.46
17	2022	1	1	FAIRFIELD INN And SUITES	Lodging	1588.41

Рисунок 4.2 – Оновлений датасет (рисунок створено самостійно)

Наступним кроком у реалізації буде передавання оновленого датасету до python сервісу задля проведення вже безпосередньо самого аналізу даних.

Для успішного проведення аналізу, дані потрібно нормалізувати, python надає багато різних бібліотек для таких цілей, в нашому випадку буде досить sklearn.preprocessing, нам знадобиться LabelEncoder та StandardScaler.

LabelEncoder – перетворює текстові (категоріальні) дані в числові. Алгоритми машинного навчання зазвичай працюють з числовими даними, тому категоріальні значення (наприклад, назви міст або категорії товарів) потрібно перетворити в числа. LabelEncoder кодує кожну категорію в унікальне числове значення, що дозволяє алгоритмам обробляти ці дані.

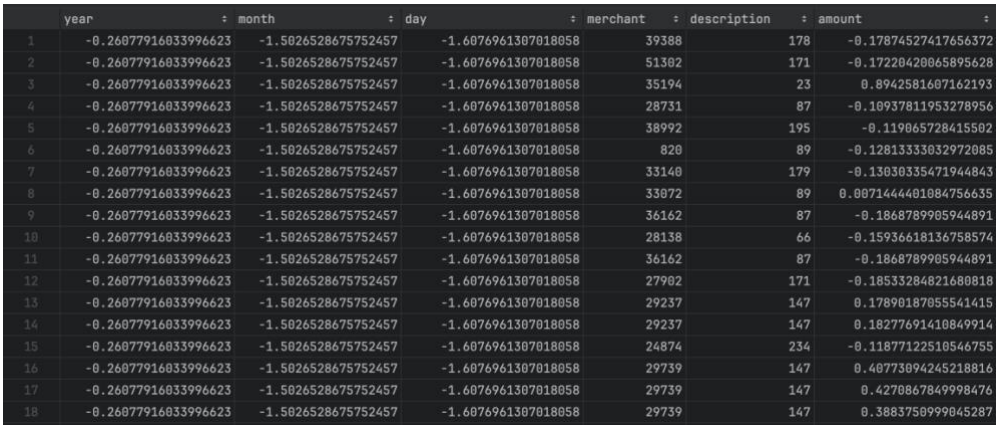
StandardScaler – нормалізує числові дані, щоб вони мали середнє значення 0 і стандартне відхилення 1. Масштабування даних важливе для багатьох алгоритмів машинного навчання, оскільки вони є чутливими до різних масштабів ознак. Якщо одна ознака має значно більші значення, ніж інша, це може вплинути на ефективність алгоритму. StandardScaler стандартизує дані, щоб всі ознаки мали однаковий масштаб, що покращує продуктивність алгоритмів [28].

Нижче наведено приклад коду, який отримує датасет та нормалізує його перед виконанням аналізу даних:

```
def get_normalized_data():
```

```
    df = pd.read_csv('updated-data.csv')
    df['merchant'] = merchant_label_encoder.fit_transform(df['merchant'])
    df['description'] =
description_label_encoder.fit_transform(df['description'])
    df[['year', 'month', 'day', 'amount']] = scaler.fit_transform(df[['year',
'month', 'day', 'amount']])
    return df
```

Приклад датасету із нормалізованими даними наведено нижче (див. рис. 4.3).



	year	month	day	merchant	description	amount
1	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	39388	178	-0.17874527417656372
2	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	51302	171	-0.17220420065895628
3	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	35194	23	0.8942581607162193
4	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	28731	87	-0.10937811953278956
5	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	38992	195	-0.119065728415502
6	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	820	89	-0.12813333032972085
7	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	33140	179	-0.13030335471944843
8	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	33072	89	0.0071444401084756635
9	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	36162	87	-0.18687899059444891
10	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	28138	66	-0.15936618136758574
11	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	36162	87	-0.18687899059444891
12	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	27902	171	-0.18533284821680818
13	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	29237	147	0.17890187055541415
14	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	29237	147	0.18277691410849914
15	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	24874	234	-0.11877122510546755
16	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	29739	147	0.40773094245218816
17	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	29739	147	0.4270867849998476
18	-0.26077916033996623	-1.5026528675752457	-1.6076961307018058	29739	147	0.3883750999045287

Рисунок 4.3 – Приклад нормалізованих даних (рисунок створено самостійно)

Тепер у нашій програмі python створено так званий dataFrame який зберігає в собі нормалізовані дані із csv файлу, далі ці дані можуть бути використані в самому аналізі.

Першим кроком до прогнозування буде використання алгоритму лінійної регресії, він надасть нам певні результати, які потім можуть бути розбиті на кластери та проаналізовані окремо.

Для початку нам треба створити модель та натренувати її перед використанням у прогнозуванні. Після цього в нас створюється натренована модель, яку можна використовувати вже безпосередньо для прогнозування транзакцій. В якості періоду прогнозування було обрано наступний місяць. Нам потрібно надати моделі датасет даних, який би мав значення для усіх незалежних змінних, окрім тієї, що прогнозується. Нижче наведено приклад коду для цього:

```
normalized_data = get_normalized_data()
```

```
x_amount = normalized_data[['year', 'month', 'day', 'description']]
y_amount = normalized_data['amount']
amount_model = LinearRegression()
amount_model.fit(x_amount, y_amount)
predicted_df = amount_model.predict(predictable_df)
```

Тепер ми маємо прогнозовані транзакції на наступний місяць. На рисунку 4.4 можна детальніше побачити розподіл прогнозованих значень сум транзакцій.

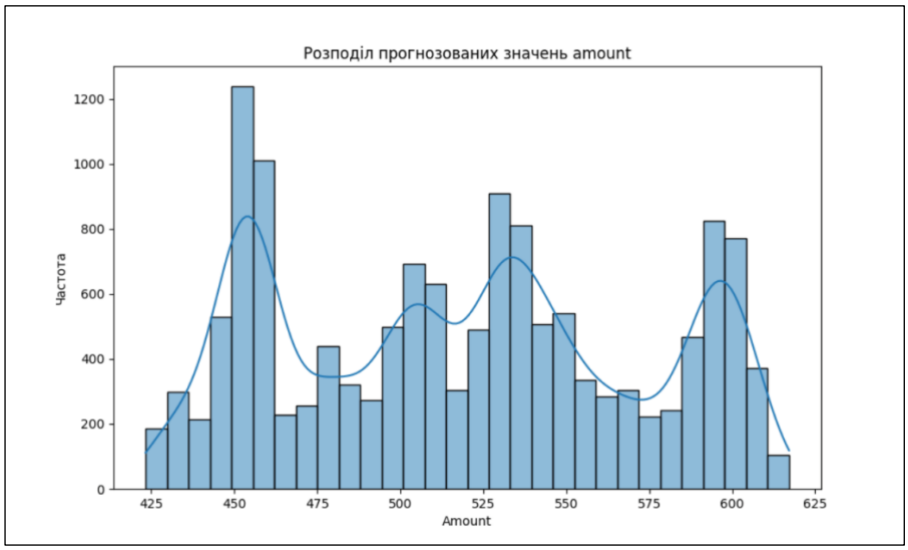


Рисунок 4.4 – Розподіл прогнозованих значень (рисунок створено самостійно)

Далі ми розіб’ємо ці дані на кластери, а після просумуємо значення amount для кожного кластеру, таким чином ми отримаємо прогнозовану суму витрат на кожен із груп транзакцій, створених алгоритмом.

Для того, щоб розбити ці дані на кластери, спочатку треба розуміти яку кількість кластерів ми повинні отримати на виході. Для цього існує декілька різних способів знаходження потрібної кількості кластерів для датасету, в моєму випадку було обрано один із найпоширеніших – це метод ліктя.

Метод ліктя – це спосіб визначення оптимальної кількості кластерів для алгоритму K-means. Його суть полягає у побудові графіку залежності значення інерції (внутрішньої суми квадратів відстаней від кожної точки до її центроїда) від кількості кластерів. Інерція зменшується з збільшенням кількості кластерів, але після певного моменту зменшення стає незначним. Цей момент нагадує форму ліктя на

графіку. Нижче наведено фрагмент коду, який демонструє виконання цього методу:

```
      inertia = []
K = range(1, 20)  # Виберіть діапазон кількості кластерів
data = get_normalized_data().head(14350)
for k in K:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(data)
    inertia.append(kmeans.inertia_)
plt.figure(figsize=(8, 6))
plt.plot(K, inertia, 'bx-')
plt.xlabel('Кількість кластерів')
plt.ylabel('Інерція')
plt.title('Метод ліктя для визначення оптимальної кількості кластерів')
plt.show()
```

Щоб побачити наглядно, яка кількість кластерів підходить найкраще для виконання нашої задачі, ми отримали потрібний графік, який наведено на рисунку 4.5.

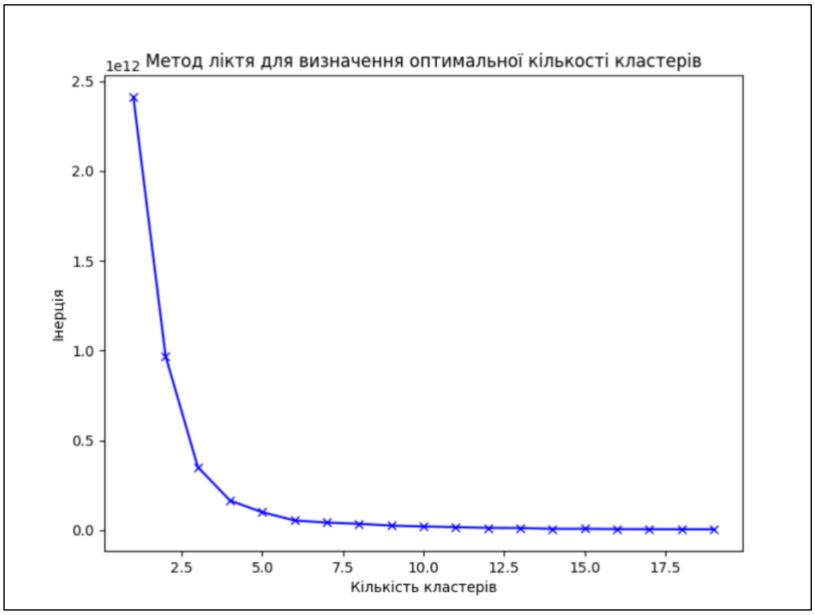


Рисунок 4.5 – Знаходження потрібної кількості кластерів (рисунок створено самостійно)

З графіку можна побачити, що значення інерції перестає помітно зменшуватися саме на значенні кластерів 6, то це і є потрібна кількість для

виконання алгоритму.

Наступним нашим кроком є кластеризація отриманих даних, задля більш детальної візуалізації прогнозових груп витрат. Нижче наведено приклад коду алгоритму:

```
    kmeans = KMeans(n_clusters=6, random_state=0)
predictable_data['cluster'] = kmeans.fit_predict(predictable_data)
predictable_data_decoded = decode_data(predictable_data.copy())
grouped = predictable_data_decoded.groupby('cluster')
pca = PCA(n_components=2)
reduced_data = pca.fit_transform(predictable_data[['year', 'month', 'day',
'merchant', 'description', 'amount']])
predictable_data_decoded['pca1'] = reduced_data[:, 0]
predictable_data_decoded['pca2'] = reduced_data[:, 1]
plt.figure(figsize=(10, 7))
for cluster in range(6):
    cluster_data =
predictable_data_decoded[predictable_data_decoded['cluster'] == cluster]
    plt.scatter(cluster_data['pca1'], cluster_data['pca2'], label=f'Cluster
{cluster}')

plt.xlabel('PCA 1')
plt.ylabel('PCA 2')
plt.title('Visualization of Clusters')
plt.legend()
plt.show()
```

Після виконання алгоритму, ми отримаємо прогнозові дані, розбиті на кластери за певними ознаками. Декодувавши найбільш використовувані значення description можна орієнтовно зрозуміти ознаку групування даних та сформувати витрати у категорії для відображення їх користувачу. На рисунку 4.6 можна детальніше побачити візуалізацію роботи алгоритму.

Графік, створений за допомогою PCA (Principal Component Analysis). Це метод зменшення вимірів, який зберігає якнайбільше варіації в даних. На графіку PCA 1 та PCA 2 представляють два головних компоненти, які містять найбільше інформації (варіації) з оригінальних даних. Тобто, замість шести вимірів даних (year, month, day, merchant, description, amount), ми отримаємо два нових виміри, які зберігають основну структуру даних.

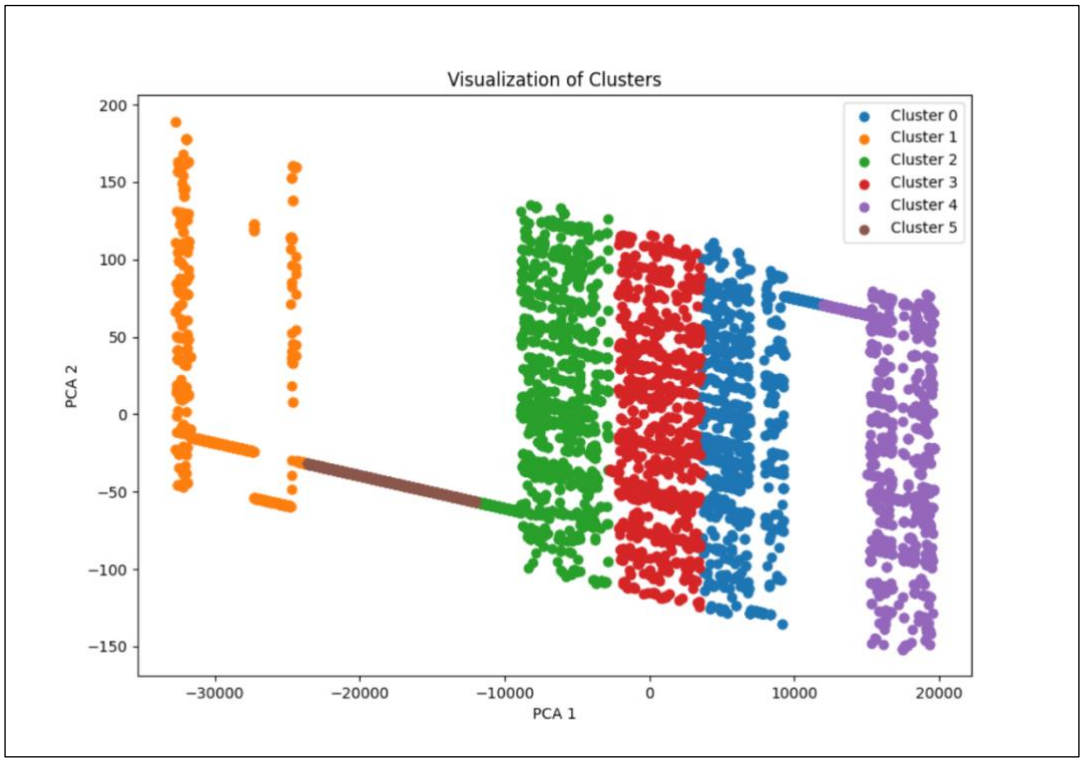


Рисунок 4.6 – Візуалізація кластерного розподілу (рисунок створено самостійно)

Кожна точка на графіку представляє один транзакційний запис з нашого датасету, зменшений до двох вимірів за допомогою PCA. Точки, згруповані разом, представляють кластери, створені алгоритмом. Кожен колір відповідає різному кластеру. Відстань між точками на графіку відображає схожість між транзакціями. Ближчі точки є більш схожими, ніж точки, що знаходяться далі одна від одної.

Далі отриманні категоризовані дані потрібно інтерпретувати та передати до Java сервісу для відображення їх на сторінці користувачу. Для інтерпретації кластера можна використовувати аналіз центроїдів кластерів та частоту категоріальних даних, щоб зрозуміти переважаючі значення в кластері та ідентифікувати сферу витрат [29].

Після отримання кластерів даних сервісом Java, їх можна передати на відображення користувачу. Він тепер може бачить прогнозовані значення витрат, поєднанні у категорії даних зі схожими ознаками. Нижче наведено приклад коду, що відповідає за відображення кластерів:

```
    <>
  <TransactionGroupsControlBar/>
  <ContentCover status={status}>
    {rows.map(row =>
      <CardsRow key={row.id} cards={row.data.map(el =>
        <AppCard
          key={el.id}
          cardName={el.name}
          content={el.description}
          about={getLangMessage("term-group-name")}
          onDelete={() => onGroupDelete(el.id)}
          form={TransactionGroupForm}
          groupName={el.name}
          groupDescription={el.description}
          groupId={el.id}
          onCardOpen={() => onCardOpen(el.id)}
        />)}
      />)}
  </ContentCover>
</>
```

Приклад того, як кінцевий користувач бачить прогнозовані дані, розбиті на кластери можна побачити на рисунку 4.7.

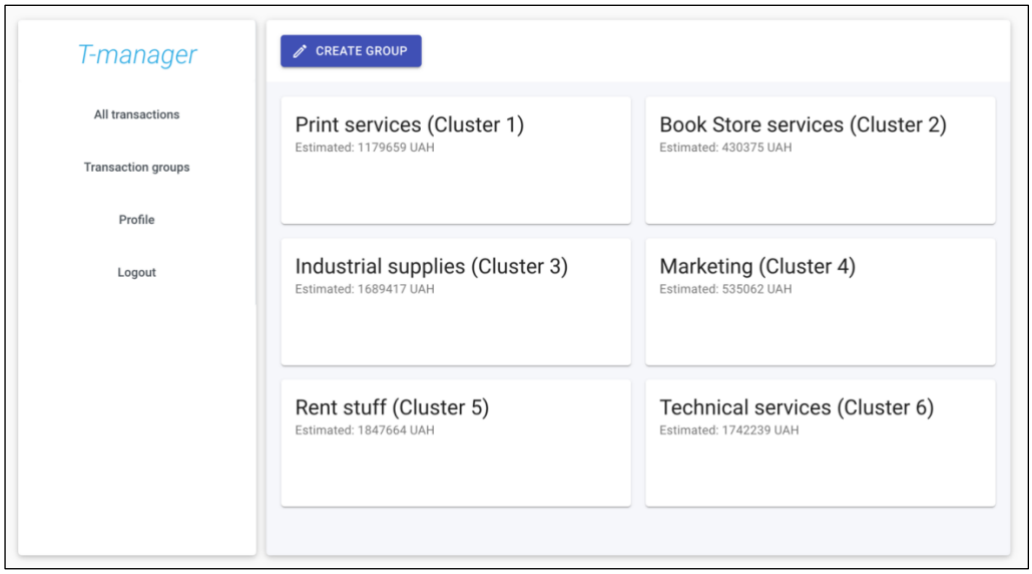


Рисунок 4.7 – Візуалізація кінцевого результату (рисунок створено самостійно)

Можна побачити, що користувач отримує відображення прогнозованих груп витрат із орієнтовною прогнозованою сумою витрат під кожну групу-кластер.

5 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

5.1 Аналіз результату прогнозування

У межах проведеного дослідження алгоритму Лінійна регресія, було отримано результат, який проаналізовано на його точність та відповідність до реальних вимог.

По-перше, був проведений аналіз алгоритму на виявлення коефіцієнта детермінації. Цей коефіцієнт описує точність передбачення моделі R^2. Коли $R^2 = 1$ 1 – це ідеальна модель, яка повністю описує варіацію залежної змінної. $R^2 = 0$ – модель краще за просте середнє передбачення $R^2 < 0$ – одель гірше ніж просте середнє передбачення.

У випадку конкретно мого алгоритму, це значення можна отримати за допомогою методу score, значення дорівнює 0.3459441770429118 – це більше 0, але ще далеко від 1, щоб вважати, що прогнозоване значення розміру транзакції дуже точне. Але це не є критичним в даному дослідженні, бо наша ціль не спрогнозувати конкретні транзакції, а отримати прогнозоване значення груп витрат у певному періоді.

Далі нам потрібно отримати статистичні характеристики прогнозованих значень алгоритму, та порівняти їх результати із фактичним реальними даними із датасету. Для цього можна використати метод describe (див. табл. 5.1).

Таблиця 5.1 – Статистичні характеристики для отриманих прогнозованих значень на період 1 місяць (таблиці виконана самостійно)

Характеристика	Прогнозоване значення	Фактичне
Кількість прогнозованих значень	14320	14383
Середнє значення	519.190149	518.163701
Стандартне відхилення	52.458417	–
Мінімальне значення	423.077525	63.22

Кінець таблиці 5.1

Перший квартіль	471.317582	126.61
Медіана	522.038229	214.53
Третій квартіль	557.943272	358.12
Максимальне значення	616.477966	2021.79
Сума витрат за період	7424418.409	6946303.1191

Із даних таблиці можна побачити, що загалом лінійна регресія дійсно має неточності у прогнозування конкретних значень транзакцій, але в цілому кількість транзакцій та загальна сума витрат залишається достатньо точною та схожою із фактичними даними. Отже метод підходить для використання у прогнозуванні загальних значень, але не підходить для прогнозування точкових даних.

5.2 Аналіз результату кластеризації

Для аналізу роботи алгоритму кластеризації можна використати інструменти візуалізації для надання характеристик отриманих кластерів.

Один із найрозповсюдженіших способів оцінки є графіки частот значень, в нашому випадку значень description в кожному кластері. Це значення фактично описує вид транзакції, і якщо ідентифікувати найбільш частіші значення в кожному кластері, можна побачити наскільки ці дані схожі один із одним для виділення їх у групу. Візуалізація наведена на рисунку 5.1.

З рисунку ми бачимо комбінацію із 5 найбільш популярних закодованих значень, які схожі між собою. Якщо декілька із найчастіших значень схожі один із одним, то групування можна назвати коректним. Наприклад значення 35 (Book store) пов’язане із значеннями 136 (Supplies-Printing Writing Pap).

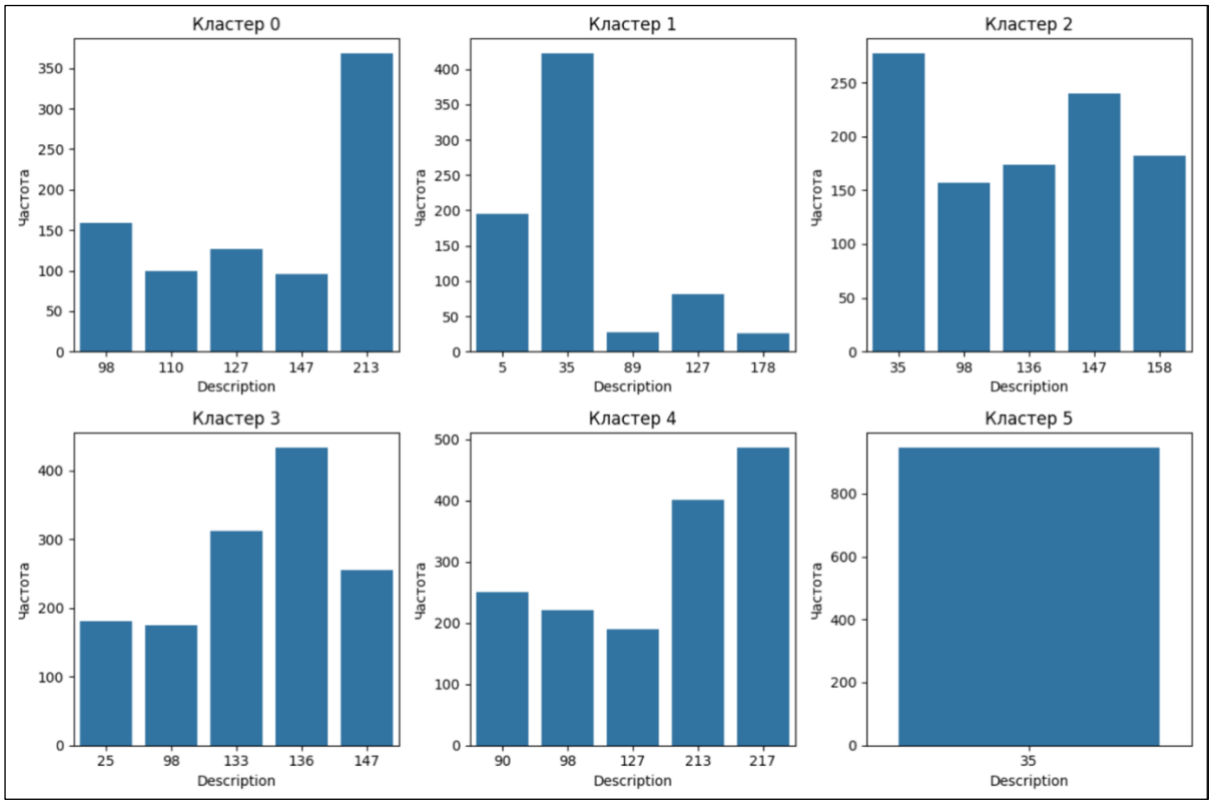


Рисунок 5.1 – Частотна візуалізація значень description

У кластері 1 значення відрізняються між собою за описом, але схожі за типом транзакції, усі 5 найчастіших значень стосуються витрат на покупку товару в магазині, то ж кластер можна ідентифікувати та надати йому загальну характеристику External services.

Таким чином кожен із отриманих кластерів дійсно має власну характеристику, за якою можна ідентифікувати та відокремити групу витрат для користувача.

Останнім кроком аналізу потрібно зробити вимірювання швидкості виконання алгоритмів, щоб зрозуміти наскільки швидко дані будуть оброблюватися для користувача. Для цього зробимо декілька замірів, запишемо отриманні значення у таблицю та вирахуємо середній час виконання (див. табл. 5.2).

Таблиця 5.2 – Вимірювання швидкості виконання алгоритмів (таблиця виконана самостійно)

Вимірювання	Значення в секундах
1	1.33
2	1.12
3	1.09
4	1.11
5	1.22

Отже після вимірювання часу виконання алгоритму для нашого датасету, ми бачимо доволі добрий результат, середнє виконання алгоритмів дорівнює 1.17 секунд, що є швидким виконанням і прийнятним для кінцевого користувача.

ВИСНОВКИ

У ході кваліфікаційної роботи, було проведено дослідження методів аналізу даних та їх застосування у системі банківського фінансового менеджменту. Було описано та проаналізовано проблемну галузь, також була сформована постановка задачі із описом веб-орієнтованої системи.

Під час дослідження було розглянуто декілька основних методів аналізу для прогнозування даних у системах банківського фінансового менеджменту:

– лінійна регресія;
– експоненційне згладжування;
– кластеризація даних.

Методи було детально розглянуто із наведенням графіків та математичних формул для обґрунтування принципу роботи їх алгоритмів. Також було створено порівняльну характеристику за принципом Парето, яка показала переваги алгоритмів у певних умовах та допомогла обрати із них найбільш доцільні для проведення дослідження.

У якості демонстрації проведеного дослідження були сформовані вимоги та створено веб-орієнтовану програмну систему менеджменту транзакцій по банківській карті, яка аналізує історичні дані та створює прогнозовані витрати на певний період у майбутньому. Було детально описано архітектуру та принцип роботи системи із використанням поєднання різних мов програмування та бібліотек, таких як Java, Python, Spring, Pandas. Для візуалізації принципу роботи системи було створено UML діаграми Use Case та діаграма діяльності для відображення взаємодії із системою. Також було створено діаграму розгортання для опису фізичної архітектури системи та ER-діаграму для опису структури бази даних.

У ході виконання практичної частини дослідження було створено систему із двох сервісів на Python та Java із покроковим описом принципу її роботи та виконання цільового завдання – прогнозування витрат користувача.

Після відтворення та тестування програмної системи із реальними даними, було отримано та проаналізовано результат її виконання, який надає певні відповіді на поставлені питання у ході дослідження.

Алгоритм лінійної регресії може бути використаний у якості методу аналізу даних для прогнозування результатів транзакцій користувача на певний період часу в майбутньому. Але точність одного прогнозованого значення не є дуже високою, тому алгоритм доцільніше використовувати у поєднанні із кластерним аналізом для об’єднання даних у групи та прогнозування загальної суми витрат для цих груп. У такому виді поєднання цих двох алгоритмів дійсно може надавати прогнозовану згруповану статистику витрат, яка є корисною та може бути використана у подальших цілях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gorokhovatskyi, V.A., Vechirska, I.D., Chetverikov, G.G. Method for building of logical data transform in the problem of establishing links between the objects in intellectual telecommunication systems Gorokhovatskyi, V.A., Vechirska, I.D., Chetverikov, G.G. Telecommunications and Radio Engineering (English translation of Elektrosvyaz and Radiotekhnika), 2016, 75(18), c. 1645-1655.

2. Мартін, Дж. А., Срівастава, Х. С. Великі дані в банківській справі: Стратегії використання аналітики / Дж. А. Мартін, Х. С. Срівастава. – Springer, 2018. – 320 с. – ISBN 978-3319720612.

3. Провост, Ф., Фосетт, Т. Наука про дані для бізнесу: Що вам потрібно знати про Data Mining і Data-Analytic Thinking / Ф. Провост, Т. Фосетт. – O'Reilly Media, Inc., 2013. – 414 с. – ISBN 978-1449361327.

4. Qian, E. E., Hua, R. H., Sorensen, E. H. Quantitative Equity Portfolio Management: Modern Techniques and Applications. - Chapman & Hall/CRC, 2007. - 464 p.

5. Гороховатський В. О. Методи інтелектуального аналізу та оброблення даних : навч. посіб. / В. О. Гороховатський, І. С. Творошенко; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2021. – С. 16 – 21.

6. Shliakhov, V., Chetverykov, G., Bozhko, I., Shliakhova, N. Predicate Data Model in the Form of a Linear Space. ECONTECHMOD: An International Quarterly Journal on Economics of Technology and Modelling Processes, 8(2), pp.41-54.

7. Хайндман, Р., Атанасопулос, Г. Прогнозування: принципи та практика [Електронний ресурс] / Р. Хайндман, Г. Атанасопулос. https://otexts.com/fpp2/.

8. Шумвей, Р. Г., Стоффер, Д. С. Аналіз часових рядів та його застосування: З прикладами на R / Р. Г. Шумвей, Д. С. Стоффер. – Springer, 2011. – 598 с. – ISBN 978-1441978646.

9. Гороховатський В. О. Методи інтелектуального аналізу та оброблення даних

: навч. посіб. / В. О. Гороховатський, І. С. Творошенко; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2021. С. 42-50.

10. Bittner K., Spence I. Use case modeling. – Addison-Wesley Professional, 2003.

11. Linzhang W. et al. Generating test cases from UML activity diagram based on gray-box method //11th Asia-Pacific software engineering conference. – IEEE, 2004. – С. 284-291.

12. Bell D. Uml basics: The component diagram //IBM Global Services. – 2004.

13. PostgreSQL B. PostgreSQL //Web resource: http://www. PostgreSQL. org/about. – 1996.

14. Momjian B. PostgreSQL: introduction and concepts. – New York : Addison-Wesley, 2001. – Т. 192.

15. Arnold K., Gosling J., Holmes D. The Java programming language. – Addison Wesley Professional, 2005.

16. Gosling J., Holmes D. C., Arnold K. The Java programming language. – 2005.

17. Johnson R. et al. The spring framework–reference documentation //interface. – 2004. – Т. 21. – С. 27.

18. Framework S. Spring framework //Available on:< https://spring. io/>. Access in. – 2018. – Т. 3.

19. Маккінні, В. Python для аналізу даних: Робота з даними з використанням бібліотек Pandas, NumPy та IPython / В. Маккінні. – O'Reilly Media, Inc., 2017. – 550 с. – ISBN 978-1491957660.

20. Грус, Дж. Наука про дані з нуля: Основні принципи з використанням Python / Дж. Грус. – O'Reilly Media, Inc., 2015. – 330 с. – ISBN 978-1491901427.

21. Drake J. D., Worsley J. C. Practical PostgreSQL. – " O'Reilly Media, Inc.", 2002.

22. Douglas K., Douglas S. PostgreSQL: a comprehensive guide to building, programming, and administering PostgresSQL databases. – SAMS publishing, 2003.

23. Gackenheimer C., Paul A. Introduction to React. – Apress, 2015. – Т. 52.

24. Fedosejev A. React. js essentials. – Packt Publishing Ltd, 2015.

25. Okamoto T., Pointcheval D. REACT: Rapid enhanced-security asymmetric cryptosystem transform //Cryptographers' Track at the RSA Conference. – Springer, Berlin, Heidelberg, 2001. – С. 159-174.

26. Mitropoulos D. et al. Time present and time past: analyzing the evolution of JavaScript code in the wild //2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR). – IEEE, 2019. – С. 126-137.

27. Smelyakov K., Chupryna A., Bohomolov O., Ruban I. The Neural Network Technologies Effectiveness for Face Detection // IEEE Third International Conference on Data Stream Mining & Processing (DSMP), August 21-25. – 2020. – P. 201-205.

28. Sharonova, N., Kyrychenko, I., Tereshchenko, G. Application of Big Data methods in E-learning systems // 5th International Conference on Computational Linguistics and In-telligent Systems (COLINS-2021), Kharkiv, Ukraine, April 22-23, 2021. – CEUR Workshop Proceedings, 2021, 2870, Volume I, pp. 1302-1311

29. I. Afanasieva, N. Golian, O. Hnatenko, Y. Daniiel, K. Onyshchenko. Data exchange model in the internet of things concept // Telecommunications and Radio Engineering (English translation of Elektrosvyaz and Radiotekhnika), 2019, 78(10), стр. 869-878.