Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: «Розробка програмного забезпечення для рекомендацій візуалізації складних даних»

Виконав: студент VI курсу, групи СПм-61
спеціальності
121 – Інженерія програмного забезпечення
(шифр і назва спеціальності)

(підпис) Кривко А.О. (прізвище та ініціали)

Керівник (підпис) Михалик Д.М. (прізвище та ініціали)

Нормоконтроль (підпис) Стоянов Ю.М. (прізвище та ініціали)

Завідувач кафедри (підпис) Петрик М.Р. (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль 2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ проф. Петрик М.Р.
(підпис) (прізвище та ініціали)
« » 20___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)
за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)
студенту Кривку Анатолію Олександровичу

1. Тема роботи Розробка програмного забезпечення для рекомендацій візуалізації складних даних

Керівник роботи Михалик Дмитро Михайлович, к.т.н., доц. каф. ПІ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
Затверджені наказом по університету від « 6 » грудня 2024 року № 4/7-1159
2. Термін подання студентом роботи 13.12.2024
3. Вихідні дані до роботи наукові літературні джерела

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1 Системи рекомендації візуалізацій. 2. Постановка задачі, опис даних
3. Практична частина. 4. Охорона праці та безпека в надзвичайних ситуаціях

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Тема роботи. 2. Мета, задачі, метоли дослідження. 3. Об'єкт, предмет, наукова новизна
4. Порівняння систем рекомендації візуалізацій на основі алгоритмів машинного навчання.
5. Процес вибору дизайну для створення візуалізацій.
6. Приклад опису візуалізації, отриманої за допомогою Plotly, у JSON форматі.
7. Приклади JSON файлів. 8. Діаграма варіантів використання застосунку. .
9. Діаграма послідовності прецедента. 10. Технологічний стек.
11. Визначення типу візуалізації.12. Матриці помилок
13. Скрін-шоти вікна застосунку. 14. Приклади роботи застосунку
15. Висновки. Основні результати проведеного дослідження

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к. т. н., доц., зав. каф. КС		
Безпека в надзвичайних ситуаціях			

7. Дата видачі завдання _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Аналіз предметної галузі дослідження	19.09–27.09.24	Виконано
2.	Обґрунтування актуальності дослідження	30.09–10.10.24	Виконано
3.	Огляд існуючих механізмів та засобів	11.10–28.10.24	Виконано
4.	Проведення дослідження механізмів та засобів опрацювання даних	29.10–15.11.24	Виконано
5.	Оформлення розділу «Системи рекомендації візуалізацій»	16.11–28.11.24	Виконано
6.	Оформлення розділу «Постановка задачі, опис даних»	29.11–05.12.24	Виконано
7.	Оформлення розділу «Експериментальні результати»	06.12–10.12.24	Виконано
8.	Оформлення розділу «Охорона праці та безпека в надзвичайних ситуаціях»	05.12–12.12.24	Виконано
9.	Нормоконтроль	13.12–15.12.24	Виконано
10.	Перевірка на плагіат	13.12–17.12.24	Виконано
11.	Попередній захист роботи	20.12.24	Виконано
12.	Захист кваліфікаційної роботи	27.12.24	

Студент _____ Кривко А.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Михалик Д.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Кваліфікаційна робота магістра містить: 70 сторінок, 30 рисунків, 8 таблиць, 33 джерел

ВІЗУАЛІЗАЦІЯ ДАНИХ, ДЕРЕВО ПРИЙНЯТТЯ РІШЕНЬ, МАШИННЕ НАВЧАННЯ, ОВЕРСЕМПЛІНГ, РЕКОМЕНДАЦІЇ ВІЗУАЛІЗАЦІЙ

У роботі проведено дослідження на класичних алгоритмах машинного навчання для вибору найбільш придатних візуалізацій до набору даних, запропоновано способи покращення точності таких алгоритмів, також написано програмне забезпечення, котре візуалізує отримані результати.

Вивчено алгоритми існуючих систем рекомендацій візуалізацій. Вибрано ознаки, що видобуваються з даних, на яких за допомогою алгоритмів машинного навчання визначається придатний тип візуалізації. Обрано і навчено алгоритми на підготовлених наборах даних. Продемонстровано можливість використовувати ці алгоритми для веб-застосунку, який надає рекомендації типу візуалізацій.

ABSTRACT

The qualifying thesis contains: 70 pages, 30 figures, 8 tables, 33 sources

DATA VISUALIZATION, DECISION TREE, MACHINE LEARNING, OVERSAMPLING, VISUALIZATION RECOMMENDATIONS

In the thesis research has been conducted on classical machine learning algorithms to select the most suitable visualizations for a data set, suggests ways to improve the accuracy of such algorithms, writes software for visualization.

The algorithms of existing visualization recommendation systems have been studied. Features extracted from data have been selected, on which a suitable visualization type is determined using machine learning algorithms. Algorithms have been selected and trained on prepared data sets. The possibility of using these algorithms for a web application that provides recommendations on the type of visualizations has been demonstrated.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

API (Application Programming Interface) – опис способів взаємодії двох або більше програмних забезпечень.

JSON (JavaScript Object Notation) - текстовий формат, призначений для передачі інформації.

RNN (recurrent neural networks) – рекурентні нейронні мережі.

НД – набір даних.

ПЗ – програмне забезпечення.

Візуальна аналітика - напрямок в аналізі даних, який фокусується на винесенні аналітичного рішення за допомогою створення візуальних інтерактивних інтерфейсів користувача в процесі збору інформації, попередньої обробки даних і встановлення їх взаємозв'язків.

ЗМІСТ

Вступ 8
1 Системи рекомендації візуалізацій 10
1.1 Системи на базі правил 10
1.1.1 Ранні дослідження (до 2010 року) 10
1.1.2 Voyager 15
1.1.3 Недоліки інструментів 15
1.2 Інструменти рекомендації візуалізацій на базі алгоритмів машинного навчання 17
1.2.1 DeepEye 17
1.2.2 Data2Vis 18
1.2.3 Table2Chart 20
1.2.4 Порівняння систем рекомендації візуалізацій на базі алгоритмів машинного навчання 21
1.3 Висновок до першого розділу 23
2 Постановка задачі, опис даних 24
2.1 Візуалізація є вибором дизайну 24
2.2 Симуляція рекомендацій для обирання дизайну 25
2.3 Джерело даних 27
2.4 Опис даних 31
2.5 Видобування ознак 33
2.6 Архітектура застосунку 34
2.7 Діаграма варіантів використання застосунку 35
2.8 Діаграма послідовності прецедента «Візуалізувати дані» 36
2.9 Висновок до другого розділу 37
3 Практична частина 38
3.1 Підготовка даних 38
3.2 Встановлення виду візуалізації 39
3.3 Встановлення осі для відтворення даних 47

3.4 Робота із веб-застосунком..... 50
3.5 Висновок до третього розділу 53
4 Охорона праці та безпека життєдіяльності 55
4.1 Охорона праці..... 55
4.2 Комп’ютерне забезпечення процесу оцінки радіаційної та хімічної обстановки..... 58
4.3 Висновок до четвертого розділу..... 60
Висновки 61
Перелік джерел посилання 62
Додаток А. Тези конференції
Додаток Б. Диск з роботою

ВСТУП

Актуальність теми дослідження. Побудова графіків у багатовимірному НД – досить розповсюджене завдання у різних сферах: навчальна, наукова, фінансова і т. д. Досить часто таке представлення застосовують для одержання інформації, спілкування та прийняття рішень. Досягнення ефективної візуалізації потребує значних людських затрат та перебуває у залежності від досвіду у сферах дизайну різного роду та аналізу даних. Застосування різних програмних продуктів для візуалізації обмежене потребою спеціаліста у додаткових скілах. Гнучкі засоби та досвід необхідні для створення візуалізацій в деяких моментах, ручне відображення є непотрібним для багатьох розповсюджених випадків, зокрема підготовче дослідження даних та створення базових візуалізацій.

Для вирішення завдань рекомендації візуалізацій застосовуються інструменти та засоби на базі правил та, дедалі частіше, системи, що базуються на алгоритмах машинного навчання.

Метою роботи є розробка ПЗ для рекомендації візуалізацій запропонованого НД.

Для досягнення мети потрібно вирішити наступні завдання:

– дослідити існуючі системи рекомендації візуалізацій;

– вивчити алгоритми, на яких вони базуються;

– вибрати ознаки, котрі видобуваються з даних, на яких за допомогою алгоритмів машинного навчання визначається придатний тип візуалізації;

– обрати і навчити алгоритми на підготовлених НД;

– продемонструвати можливість використовувати обрані алгоритми для рекомендації типу візуалізацій.

Об’єкт дослідження: візуалізація складних даних.

Предмет дослідження: сюжети рекомендації візуалізацій, взаємозв'язку між НД і типом візуалізації.

Методи дослідження: наукові праці вітчизняних і зарубіжних вчених за

тематикою дослідження, визначальні положення інженерії програмного забезпечення; методи - аналітичний, порівняльний, узагальнення.

Наукова новизна роботи:

– запропоновано розглянути дизайн візуалізації як значення, що максимізує ефективність візуалізації з урахуванням НД та контекстуальних факторів;

– запропоновано способи покращення точності класичних алгоритмів машинного навчання для підбору найбільш відповідних візуалізацій до НД.

Практичне значення одержаних результатів. Розроблене програмне рішення забезпечить допоможе проаналізувати, виявити особливості, зробити рекомендації за наданими даними для візуалізацій складних даних у різних сферах діяльності

Апробація. Окремі результати дослідження доповідалися на XII науково-технічній конференції «Інформаційні моделі, системи та технології» як опубліковані тези.

1 СИСТЕМИ РЕКОМЕНДАЦІЇ ВІЗУАЛІЗАЦІЙ

1.1 Системи на базі правил

В основі таких інструментів є принципи, котрі доволі часто ґрунтуються на експериментальних даних та експертному досвіді. Результатом використання евристики є опосередковане отримання кращих застосувань, котрі одержані із досвіду іншого фахівця зі створення та використання візуалізацій.

Як найпростіший приклад такої системи можна навести дерево рішень, котре дає змогу встановити, чи є кругова діаграма чи альтернативний тип діаграми найкращим вибором (рис. 1.1).

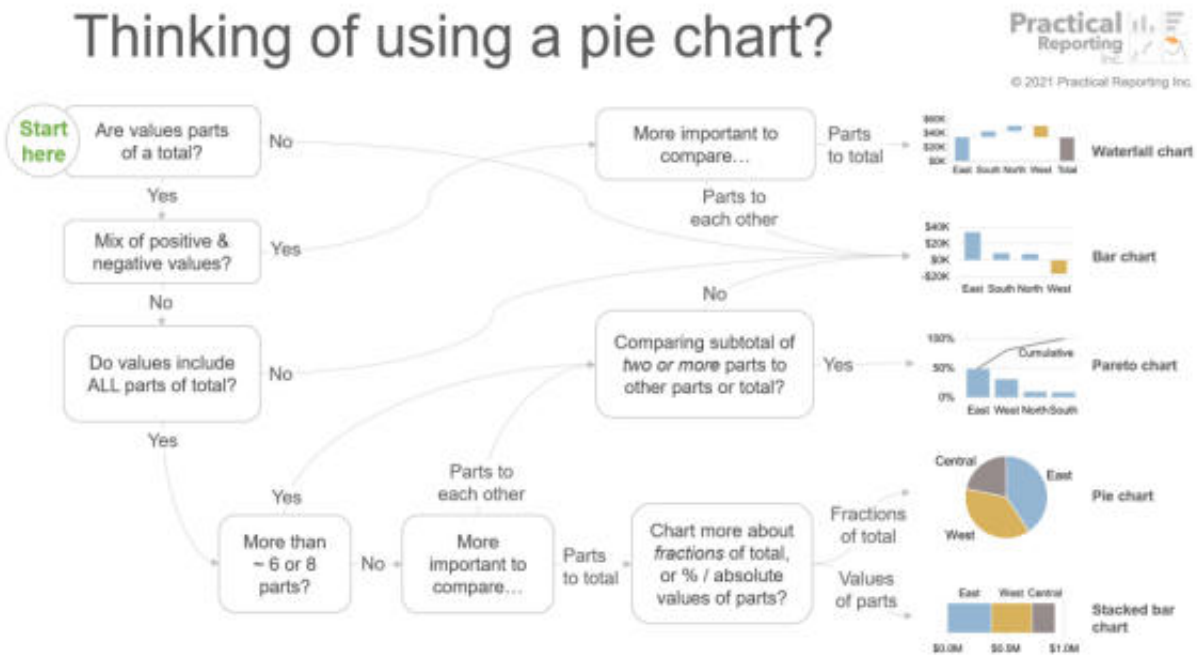

Рисунок 1.1 – Вибір типу діаграми як дерева рішень

1.1.1 Ранні дослідження (до 2010 року)

BHART. Найперший інструмент, який представив умови встановлення того,

котрий варіант візуалізації придатний для визначених параметрів даних. Так як цю роботу було написано в 1981 році, набір можливих візуалізацій не був таким розмаїтим, як ми це маємо зараз. Пропонувався вибір тільки лінійних, кругових та стовпчикових діаграм [4]. За використання безперервної функції, рекомендовано було лінійну діаграму. Якщо ж набори діапазонів можна було просумувати – пропонувалася круговий тип діаграми, та стовпчикова діаграма - для решти випадків. Незважаючи на те, що на зараз BHART має досить примітивний вигляд, вона стала основою для інших наступних систем.

APT. У 1986 Макінлеєм було запропонувано формалізацію та систематизацію варіантів графічного дизайну з метою забезпечити автоматизацію процесу відтворення графіки [15]. Змінну «форма» найкраще використовуватиме відображення відмінностей і подібностей між об'єктами. Макінлей переніс в код семантику Бертена як оператори алгебри, котрі застосовувалися для відшукання ефективних відображень інформації. Автор базував свої висновки на засадах виразності та ефективності. Макінлей прагнув створити перелік мов графіки, котрі можна було б відфільтровувати за атрибутами виразності і впорядковувати за критеріями ефективності кожного вводу. Пізніше Рот та Меттіс [13] запропонували додати інші види візуалізації.

В основі APT лежить алгебра композиції, котра складається з базового набору та операторів композиції. До використання такої алгебри параметри даних потрібно описати кодом при допомозі визначеної мітки візуалізації, котра має підпадати під правила, наведені у таблиці. У композиційній алгебрі основний набір закодовує параметри даних через візуальні змінні (табл. 1.1). Оператори утворюють різні уявлення, складаючи різні базисні набори різних атрибутів даних. Вони становлять візуалізацію через об’єднання частин, котрі закодовують ту саму інформацію.

Таким чином, Макінлей був тим, хто хотів першим сформувати автоматичний механізм проектування графіків, котрий сприймав би всі реляційні дані як вхідну інформацію та вирішував, як розкласти графічні можливості для ефективного передавання даних.

Таблиця 1.1 – Атрибути даних зіставлення візуальних міток (запропоновані Макінлеєм)

Тип даних / Властивість	Номінальний	Порядковий	Кількісний
Розмір		•	•
Насиченість		•	•
Текстура	•	•	•
Колір	•	•	
Орієнтація	•		
Форма	•		

Згодом специфікації Макінлея було використано при розроблення системи. Пізніше Ханрахан удосконалив специфікації Макінглея на декларативну візуальну мову, котра зараз відома нам як VizQL [8].

Tableau та Show Me [15]. Цей модуль візуалізації застосовує специфікації VizQL для надання рекомендацій щодо візуалізації в автоматичному режимі. Як тільки юзер вибирає цікаві для нього атрибути даних, Show Me застосовує положення Visual Mapping Tableau (табл. 1.2), для розпізнавання видів візуалізації.

Tableau встановлює необхідний вид візуалізації для застосування, шляхом перегляду типів параметрів в даних.

Для прикладу, точкова діаграма вимагає наявності від двох до чотирьох кількісних параметрів. До того ж, застосунок здійснює ранжування кожної візуалізацію щодо знайомства та найкращих практик дизайну. Зрештою, вона рекомендує найбільш ту, яка найбільш підходить візуалізацію з найвищим рейтингом. Додаток Show Me від Tableau є набором певних дій, що допомагають у побудові невеликих множинних уявлень та рекомендованих типів діаграм.

Таблиця 1.2 – Правила візуального зіставлення у Tableau

Тип поля 1	Тип поля 2	Тип мітки	Тип відображення
C	C	Текст	Таблиця
Qd	C	Стовпчик	Стовпчикова діаграма
Qd	Cdate	Лінія	Лінійна діаграма
Qd	Qd	Форма	Крапкова діаграма
Qi	C		Діаграма Ганта
Qi	Qd	Лінія	Лінійна діаграма
Qi	Qi	Форма	Крапкова діаграма

Many Eyes. У попередніх роботах інструменти візуалізації створювалися спеціалістами автономно. Цей застосунок вже був першим досить відомим та загальнодоступним веб-сайтом, на котрий бажаючі мали можливість разом вантажити дані та формувати візуалізації у інтерактивному режимі [23].

Тут візуалізація формується як порівняння НД із компонентом візуалізації (або методами візуалізації). Перелік таких компонентів представлений у табл. 1.3.

Будь-який компонент містить візуалізацій, котрі мають спільну схему даних. Якщо юзер здійснює вибір якихось стовпців даних, тоді ці стовпці порівнюються зі тією схемою, котра пов'язана із визначеною візуалізацією. Сама схема даних є масивом іменованих типізованих слотів. Наприклад: T у наведеній табл. 1.3 є текстовими даними в одному стовпцю, толі як T+ вказує на те, що НД має вже більше, ніж один стовпець.

Таблиця 1.3 – Схема візуального відтворення Many Eyes

Тип графіка	Схема даних
Пухирцева діаграма, Гістограма, Кругова діаграма, Карти, Хмара слів	{Підписи/найменування: T, Значення: N1
Стовпчикова діаграма, Реберний граф, Діаграма із накопиченням	{Мітки осі: T, Значення: N+}
Діаграма мережі	{Від: T, До: N+}
Точкова діаграма	{Вісь X : N, Вісь У : N , Підпис: T, [Розмір точок: N]}
Діаграма з накопиченням /категорією	{Ієрархія: T+, Значення: N+}
Деревоподібна діаграма	{Ієрархія: T+, Розмір: N Колір: N}
Хмара слів	{U}

Таким чином, деревоподібну карту ймовірно представити як впорядковану сукупність текстових стовпців, в якій будь-який рядок відображає ієрархічну дорогу від вершини до фінального елемента. У подальшому сукупність даних і побудована візуалізація можуть використовуватися іншими користувачами для внесення коментарів, пропозицій та майбутніх покращень, що здає змогу мати загальне середовище для формування візуалізації [12].

Таким чином, популярність Many Eyes підтвердила зручність застосування та простоту доступу до ПЗ для візуалізації у вигляді веб- застосунку. Поряд із цим середовище панелі інструментів, що надається Tableau, аналогічно стало моделлю для засобів побудови візуалізації.

1.1.2 Voyager

Це відносно новий веб-додаток з рекомендаціями з візуалізації, котрий базується на середовищі типу панелі приладів [11]. В основі Voyager є механізм рекомендацій Compass, котрий рекомендує візуалізації на базі статистичних параметрів даних. Рекомендації надані як специфікації Vega-lite [1], JSON-об’єкта, котрий визначає одне джерело даних, вид мітки, візуальне закодовування змінних даних, ключ-значення та перетворення даних, включаючи фільтри та агрегатні функції. Процес надання пропозицій Compass перш за все рекомендує перелік візуалізацій на основі одномірного зведення кожної змінної НД. Після цього застосовується перетворення даних (як приклад, агрегування чи групування) формування переліку таблиць з похідними даними. Користувач може вибрати або забрати змінні з переліку, з метою зосередження на визначеному комплекті змінних.

1.1.3 Недоліки інструментів

Як правило, виділяють такі мінуси засобів, котрі пропонують рекомендації візуалізації на базі правил:

- оскільки візуалізація є складним процесом, котрий здатен вимагати моделювання нелінійних відносин, які важко вловити за допомогою простих правил;
- створення наборів правил — коштовний процес, що ґрунтується на експертних оцінках;
- по ходу зростання розмірності даних на вході, комбінаторна сукупність правил може призвести до вибухового зростання можливих рекомендацій;
- коли візуалізації оцінюються із застосуванням переліку правил, котрі

визначені вручну, значна кількість візуалізацій одержують ідентичну за точністю оцінку [3]. Таке питання постає через спосіб підрахунку очок із застоуванням правил, котрий нерідко лише надає позитивний (або негативний) бал в залежності від того, яке правило дотримано або порушено, після чого всі ці очки з достатньо малого комплекту правил об'єднують з метою одержання фінальної оцінки [21]. Наслідком цього є те, що значне число візуалізацій будуть мати ту саму оцінку і, ось чому, вони не підпадають під ранжування;

– такі підходи не ґрунтуються на даних та їх важко адаптувати, так як треба досить регулярно додавати свіжі правила ручним способом, що є затратним з погляду часу та зусиль. Тобто їх буде складніше покращувати по ходу зміни переваг юзерів та візуалізацій з часом;

– правила для таких систем повинні бути визначені в ручному режимі окремо для кожного домену, що цікавить. Наприклад, візуалізації для спеціалістів із наукових галузей ймовірно відрізнятимуться від візуалізацій, які віддають перевагу журналістам, або тим, які підходять для людей, які не працюють з даними. Тому, ймовірно, будуть потрібні додаткові переліки прави для надання ефективних рекомендацій щодо візуалізації кожній групі.

1.2 Інструменти рекомендації візуалізацій на базі алгоритмів машинного навчання

1.2.1 DeepEye

Ця розробка [14] дає змогу поєднати формування візуалізації на базі визначених правил з моделями машинного навчання для розв’язання поставлених задач: візуалізація класифікується як «хороша» чи «погана»; можливо ранжувати списки візуалізацій.

З метою навчання моделей інструмент застосовує визначений перелік атрибутів, котрі одержані із даних для проведення візуалізації:

1. Кількість різних значень стовпця X, d(X).
2. Кількість кортежів у стовпці X, | X |.
3. Співвідношення унікальних значень r(X) = d(X)/|X| у стовпці X.
4. Максимальне max(X) та мінімальне min(X) значення у стовпці X.
5. Тип даних T(X) у стовпці X.
6. Кореляція між стовпцями c(X, У).
7. Тип візуалізації (вид діаграми.

Детектування візуалізації. Перш за все необхідно із врахуванням комбінацій стовпців списку даних та зазначеного типу візуалізації, визначити: запропонована візуалізація є хороша або погана. Для розв'язання такого завдання було взято звичайний бінарний класифікатор, зокрема тут застосовували дерева рішень.

Рейтинг візуалізації. Наступна задача – при двох запропонованих візуалізаціях вибрати ту, котра є кращою. Для цього використовувався метод машинного навчання на вирішення завдання ранжування.

Задача ранжування - це задача навчання з учителем, котра бере простір X на вході як список векторів ознак, тоді як простір на виході Y, що складається з оцінок (або рангів). Ідея в тому, щоб навчити якусь функцію $F(\cdot)$ з навчальних прикладів таким чином, щоб за присутності векторів входу x_1 і x_2 можна було визначити, який краще, $F(x_1)$ або $F(x_2)$.

НД. Застосовувалися 42 реальні НД із різних галузей, зокрема ринок нерухомості, дослідження з соціальної інженерії та транспорту. Після розмічення даних у кожному наборі були пораховані знову всі можливі візуалізації-кандидати та студентів попросили відзначити, які з них хороші, а які є поганими. Потім хороші візуалізації попарно порівняли між собою для визначення найкращих. В результаті було одержано 2520/30892 анотованих добрих/поганих графіків та 285236 попарних порівнянь візуалізацій (якщо таблиця має до візуалізацій, тоді для цієї таблиці є до $k \times (k - 1)/2$ ранжування).

1.2.2 Data2Vis

Це ПЗ наголошує на створення специфікацій візуалізацій із застосуванням правил, котрі навчені на прикладах, не вдаючись до попередньо визначеного переліку правил чи евристик [22].

Задача візуалізації даних у даній роботі представлена як задача переформатування послідовності на таку послідовність, котру здатно просто розв'язати при використанні seq2seq моделей. Послідовність на вході – НД json–формату та застосовується в навчальних прикладах, а послідовність на виході є діючою специфікацією візуалізації Vega-Lite [1].

Наявні моделі, котрі використовуються для трансляції послідовності, є сімейством мереж кодер-декодер, коли кодер зчитує і кодує вихідну послідовність як вектор визначеної довжини, а декодер власне вже перекладає (транслює) на базі такого вектора. Потім кодер-декодер навчається разом з метою отримання максимальної можливості виводу достовірного перекладу за заданої вихідної послідовності.

Модель на рис. 1.2 базується на варіанті кодер-декодер із додатковим механізмом уваги, котрий перед тим використовувався для машинного перекладу. Кодер – двонаправлена RNN, котра бере вхідну кількість вихідних токенів $x = (x_1, \ldots, x_m)$ та видає набір станів $h = (h_1, \ldots, h_m)$.

Декодер також є RNN, який обчислює ймовірність цільової послідовності $y = (y_1, \ldots, y_k)$ із застосуванням прихованого стану h. Імовірність кожного маркера в цільовому наборі буде зґенерована на базі повторюваного стану декодера RNN, минулих маркерів в цьому наборі та контекстного вектора c_i. Вектор контексту (також званий вектор уваги) і буде середньозваженою величиною вихідних станів. Його призначення – захоплення контексту вихідного набору, що дає змогу окреслити біжучий цільовий токен.

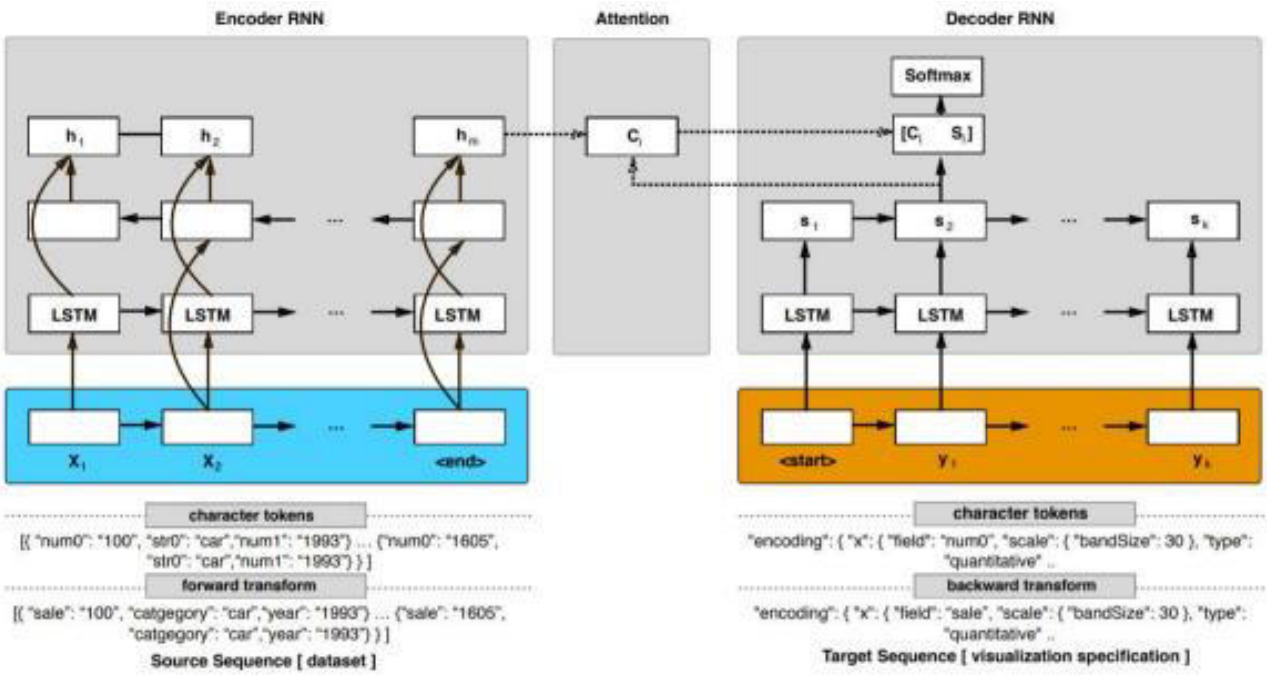

Рисунок 1.2 – Схема моделі, яка використовується в Data2Vis

Застосовувалися дворівневі RNN -кодер (двонаправлений) та -декодер, кожен з них володіє 512 одиницями довготривалої короткочасної пам'яті (LSTM) (комірок). Для обрання типу одиниці RNN для застосування, були проведені експерименти як з GRU, так і з LSTM, обидва з яких є розповсюдженими типами комірок RNN. Було виявлено, що з комірками LSTM отримувалися кращі результати (дійсні) у порівнянні з комірками GRU, що узгоджується з більш ранніми теоретичними результатами.

НД. Навчання модулі відбувається із застосуванням 4300 прикладів Vega-Lite, котрі автоматично згенеровані. Вони містять з 1-3 змінних, які сформовані на базі 11 різноманітних НД. Прогнозування моделі успішно підкріпляються через вивчення візуалізацій, котрі побудовані на базі 24 НД. НД має графіки з шістьма різноманітними видами візуалізації (діаграми різного типу) та трьома перетвореннями (як то агрегація, квантування та timeUnit). Для створення набору навчальних даних, потрібно застосувати циклічну генерацію джерела (одного рядка з НД) та цільової пари з кожного прикладного файлу. Всякий приклад у подальшому проходить відбір 50 разів, в результаті чого отримується 215 000 пар, котрі використовуються для навчання моделі.

1.2.3 Table2Chart

У [25] досліджується інструмент Table2Chart дослідження спільних темплейтів побудови діаграм, включно із запитами даних, та вибором дизайну із значного числа прикладів (як то таблиць та діаграм) та інструкції з вибору діаграм для всякої таблиці. Рекомендації щодо діаграм описані як таблиця для сукупності з оцінюванням токену чергової дії для заливання діаграми-темплейту. Таке формулювання дає змогу пропонувати діаграми із неповним наміром, якщо деякі запити даних і варіанти дизайну вже задано. Як евристика оцінки для пошуку променя була створена глибока мережа Q-значень кодера-декодера (DQN), котра здійснює вибір поля таблиці з метою заливки темплейту(ів) при викоривтанні способу скопійовування. Усі задачі рекомендацій застосовують той самий кодувальник, проте володіють власними декодерами, що дає змогу розв’язати проблему деяких витрат. Навчання DQN проходить із застосуванням змішаного навчання на багатотипній задачі головних видів діаграм. Шляхом надання своєї частини, котра кодує, різноманітним таблицям відповідно різноманітних видів діаграм, він досліджує уявлення спільних таблиць, котрі мають різного роду інформацію (семантичну та статистичну) щодо полів таблиці. Потім попередньо навчені табличні уявлення передаються для специфічних типів декодерів задач одного типу, це дає можливість усунути задачу даних, що є незбалансованими.

Однак різноманітні види діаграм володіють різними візуальними та поведінковими ефектами, проте головні дії щодо їх формування із таблиць поділяються на 2 варіанти: вибирання полів таблиці/показчика на них і втілення визначених операцій чи вказівок створення діаграм щодо організації та побудови визначених полів. Тут діаграму можна представити послідовністю кроків відносно запитів даних і вибирання дизайну.

В загальному, будову Table2Charts показано на рис. 1.3. Щоб апроксимувати q* (s, a) для створення діаграм, було розроблено архітектуру глибокої Q -мережі кодувальника-декодера (DQN) із застосуванням механізму скопійовування. Так як

постійна помилка експозиції є серйозною для генерації сукупності із застосуванням темплейтів, застосовується метод вибірки відшукання з метою навчання DQN. Зрештою, щоб вирішити проблему незбалансованих даних та взаємно покращити продуктивність різноманітних виді графіків, запропоновано застосувати сукупність елементів навчання «змішувати та переносити» з метою вирішування встановлених задач.

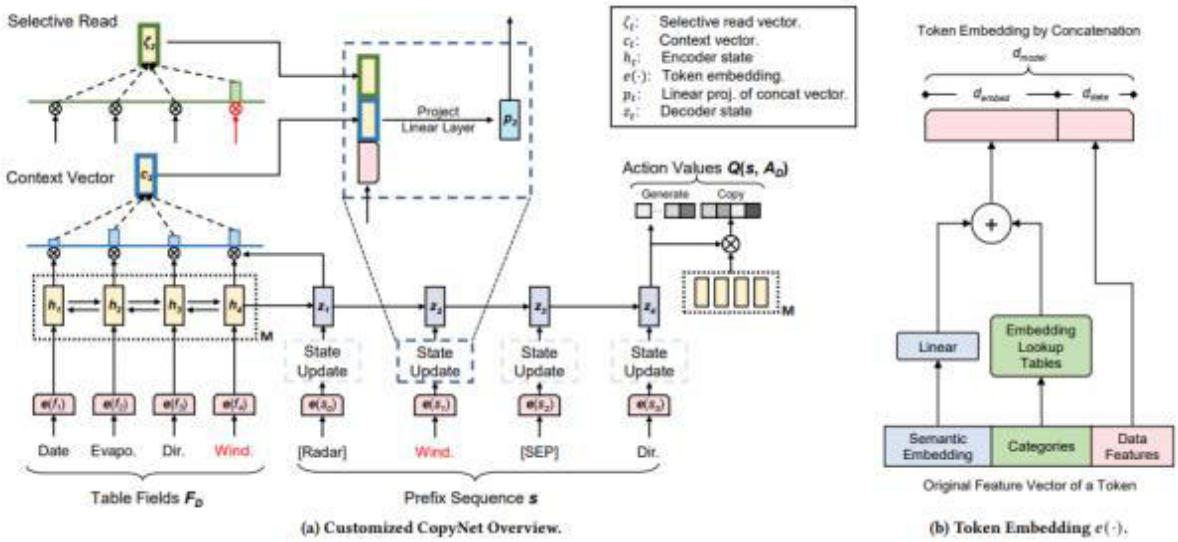

Рисунок 1.3 – Схема моделі, яка використовується в Table2Charts

НД. З мережі було зібрано великий НД із 266252 діаграм, створених із 165214 таблиць, що містяться у Excel-книгах.

Дані з таблиць Excel були витягнуті за допомогою інструмента Open XML SDK. Разом сукупність діаграм містить (у %): 42,7 лінійних, 25,6 стовпчастих, 24,6 точкових, 6,6 кругових та 0,6 радіальних.

1.2.4 Порівняння систем рекомендації візуалізацій на базі алгоритмів машинного навчання

Результати порівняльного аналізу таких систем відображені у табл. 1.4.

Таблиця 1.4 – Порівняльний аналіз інструментів рекомендації візуалізацій на базі механізмів машинного навчання

Система	Вирішуване завдання	Підхід до навчання	Моделі	Набір даних	Дані	Джерело даних	Генератор даних
Table2Charts	Запити даних + вибір дизайну	Наскрізна генерація діаграми у вигляді послідовності токенів дії	CopyNet as deep Q-network	(Повні таблиці, графіки) пари	165k таблиць з 266k графіків	Web Excel files	Людина
VizML	Вибір дизайну	5 завдань класифікації	Fully-connected feed-forward NN	(Неповні таблиці графіки) пари	119k таблиць	Plotly community feed	Людина
DracoLearn	Запити даних + вибір дизайну	М'які обмеження/правила ваг для clingo ASP solver	RankSVM	Попарне порівняння	1100 + 10 пар	Various	Правила → Інструкції
Data2Vis	Запити даних + вибір дизайну	Наскрізна генерація діаграми у вигляді послідовності рядків JSON	Character-level seq2seq NN	(Повні таблиці, графіки) пари	11 таблиць із 4300 графіками	Tool (Voyager)	Правила → Валідації
DeepEye	Запити даних + вибір дизайну	1. Хороша/погана класифікація 2. Рейтинг	1. Decision tree 2. LambdaMART	1.Погані/хороші мітки графіка 2. Попарне порівняння	42 таблиці 1. 33.4 к міток 2. 285k пар	Various	Правила → Інструкції

Результати проведеного порівняльного аналізу свідчать про переваги та недоліки розглянутих систем, які призначені для надання рекомендацій щодо викоання (проведення) візуалізації.

1.3 Висновок до першого розділу

У цьому розділі було розглянуто існуючі системи рекомендації візуалізацій з урахуванням правил і системи, засновані на методах машинного навчання.

Були розглянуті основні ідеї та алгоритми, котрі перебувають в основі даних систем, а також виявлено недоліки системи рекомендацій візуалізації на основі правил. Проведено порівняльний аналіз систем, які використовують методи машинного навчання.

2 ПОСТАНОВКА ЗАДАЧІ, ОПИС ДАНИХ

2.1 Візуалізація є вибором дизайну

Охарактеризувати початкову візуалізацію НД d ймовірно набором різновидів дизайну C = {c}, котрі взаємопов'язані між собою, вибір кожного з них відбувається із ймовірного простору c ~ ℂ. Однак не всі варіанти дизайну призводять до правильних візуалізацій - деякі варіанти несумісні між собою. Отже, множина варіантів, що призводять до правильних візуалізацій, є підмножиною простору всіх ймовірних різновидів $\mathbb{C}1 \times \mathbb{C}2 \times \ldots \times \mathbb{C}_{|C|}$.

Аналіз минулих робіт показав, що ефективність можливо визначити низькорівневими канонами сприйняття і властивостями НД, додатково до контекстуальних факторів, таких як задача, естетичне сприйняття, домен, аудиторія та середовище [16]. Інакше кажучи, аналітик здійснює вибір візуалізаційного дизайну C_{max}, який максимізує ефективність властиво візуалізації Eff, беручи до уваги НД d і контекстуальних факторів T :

$$C_{max} = arg \max_{C} \text{Eff}(C|d, T) \tag{2.1}$$

Проте ймовірно, що вибирання дизайну буде дороговартісним. Мета рекомендацій візуалізації - знизити вартість створення візуалізацій використовуючи автоматичну пропозицію підмножини різновидів дизайну $C_{rec} \subseteq C$ (рис. 2.1).

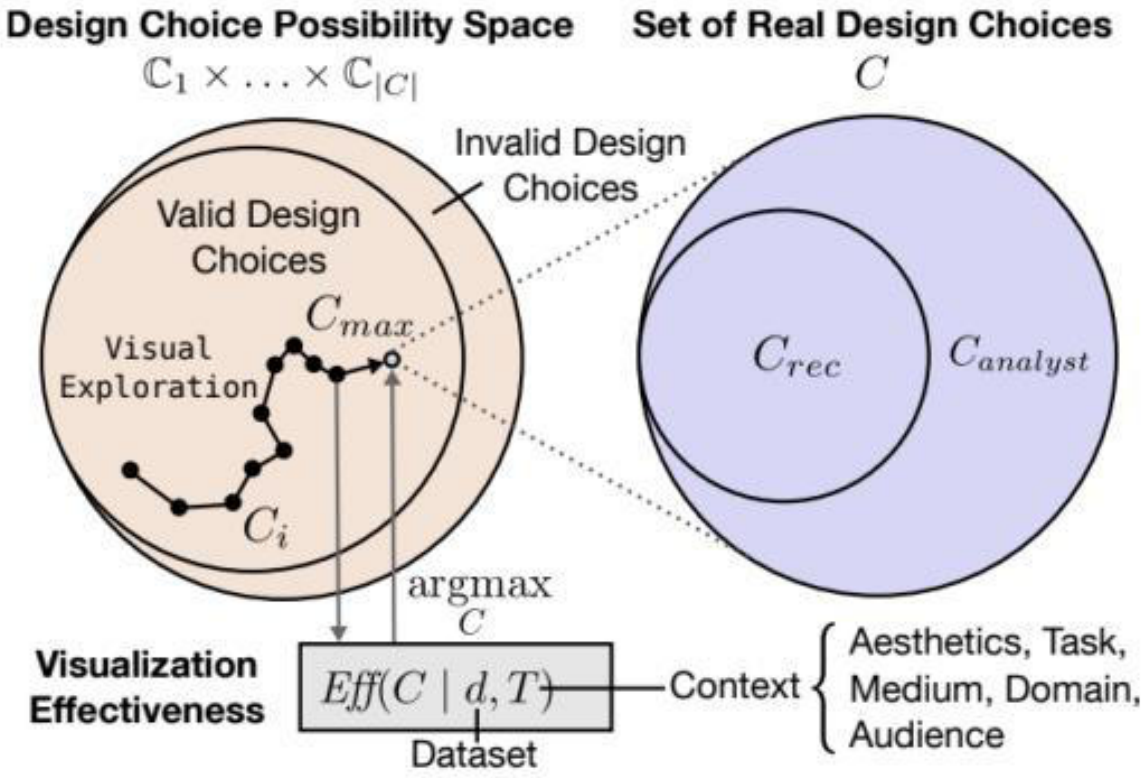

Рисунок 2.1 – Створення візуалізацій

2.2 Симуляція рекомендацій для обирання дизайну

Візьмемо до уваги один різновид обирання дизайну візуалізації c ∈ C. Позначимо C' = C \ {c} множину інших варіантів візуалізацій, крім c. З огляду на C', НД d і контекст T є ідеальна функція рекомендацій щодо вибору дизайну візуалізації F_c, результатом якої є обирання дизайну $cmax \in Cmax$ з рівняння (2.1), що максимізує ефективність візуалізації:

$$F_c(d|C', T) = c_{max} \tag{2.2}$$

Наша мета – апроксимувати F_c функцією $G_c \approx F_c$. Тепер зробимо припущення, що є НД $D = \{d\}$ та відповідних візуалізацій $V = \{V_d\}$, будь-яку

можна описати обиранням дизайну $C_d = \{c_d\}$. Інструменти рекомендацій, в основі яких є методи машинного навчання, розглядають G_c як модель із переліком атрибутів Θ_c, котра здатна навчитися на цих НД при допомозі алгоритму навчання, котрий максимізує цільову функцію Obj:

$$\Theta_{fit} = arg \max_{\Theta_c} \sum_{d \in D} Obj(c_d, G_c(d|\Theta_c, C', T)) \quad (2.3)$$

Без обмеження спільності вважатимуться, що цільова функція максимізує можливість спостереження результату навчання $\{C_d\}$. Навіть за умови, що фахівець обере неоптимальний дизайн, загальна оптимізація ймовірності всіх варіантів дизайну, що спостерігаються, все таки ймовірно буде оптимальною. Властиво це якраз той момент, коли існує обрання дизайну візуалізації $c_d = F_c(d|C', T) + noise + bias$. Тому з огляду на невидимий НД d^*, максимізація цільової функції може призвести до рекомендації, яка максимізує ефективність візуалізації.

$$G_c(d^*|\Theta_{fit}, C', T) \approx F_c(d^*|C', T) = c_{max} \quad (2.4)$$

У цій роботі модель G_c є механізмом машинного навчання, а Θ_c — параметрами, отриманими при навчанні даного алгоритму. Питання надання рекомендацій ймовірно спростити шляхом оптимізації кожного G_c незалежно та без урахування контекстних факторів: $G_c(d|\Theta_c) = G_c(d|\Theta_C, C', T)$ (рис. 2.2). Варто загадати, що незалежні рекомендації ймовірно несумісні, і водночас вони не обов'язково максимізують загальну ефективність. Створення повного висновку візуалізації вимагатиме моделювання залежностей між G_c для кожного c.

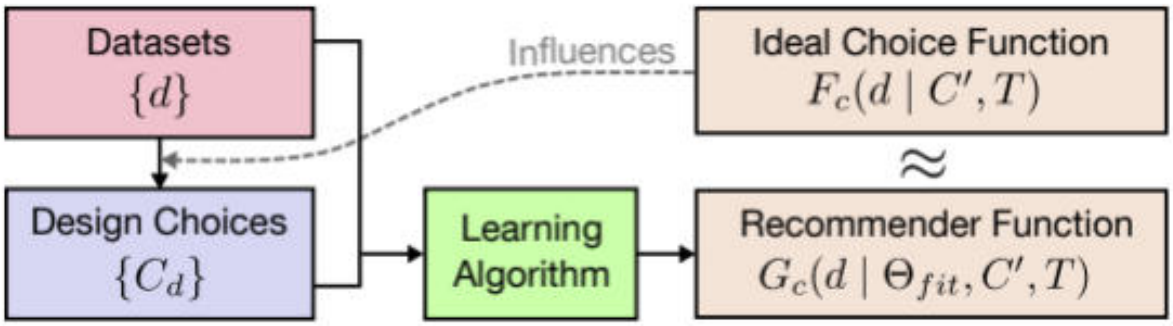

Рисунок 2.2 – Стартова модель навчання для обрання рекомендацій різновидів дизайну з НД і відповідні варіанти дизайну

2.3 Джерело даних

Фірма Plotly створює інструменти та бібліотеки для візуалізації та аналізу даних. Зокрема Plotly Chart Studio [18] є веб-застосунком, який дає змогу юзерам завантажувати НД та вручну створювати інтерактивні візуалізації D3.js та WebGL із використанням ≥ 20 видів візуалізації. Додатково наявна бібліотека Plotly Python для побудови таких самих візуалізацій за допомогою коду.

Візуалізації у Plotly задаються декларативною схемою. Тут будь-яка візуалізація задається при допомозі двох структур даних. Перша з них є списком об'єктів, які показують, як саме проходитиме візуалізація колекції даних. Другою є словник, котрий визначає естетичні моменти візуалізації, не пов'язані з даними, як то мітки та підписи осей, розміри графіка, використовувані шрифти і т.д. Для прикладу, точкова діаграма, показана на рис. 2.3, встановлюється одним об'єктом з типом «scatter» з Hp як параметр осі *x* і MPG як параметр осі *y* (рис. 2.4).

Model	MPG	Cyl	Disp	Hp	Wgt	Acc	Year	Origin
chevrolet chevelle	18	8	307	130	3504	12	70	US
buick skylark 320	15	8	350	165	3693	11.5	70	US
			...					
chevy s-10	31	4	119	82	2720	19.4	82	US

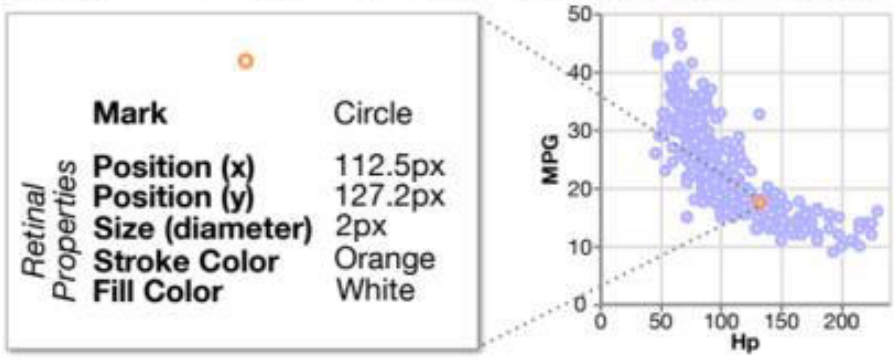

Рисунок 2.3 – Приклад точкової діаграми та даних, за якими вона побудована

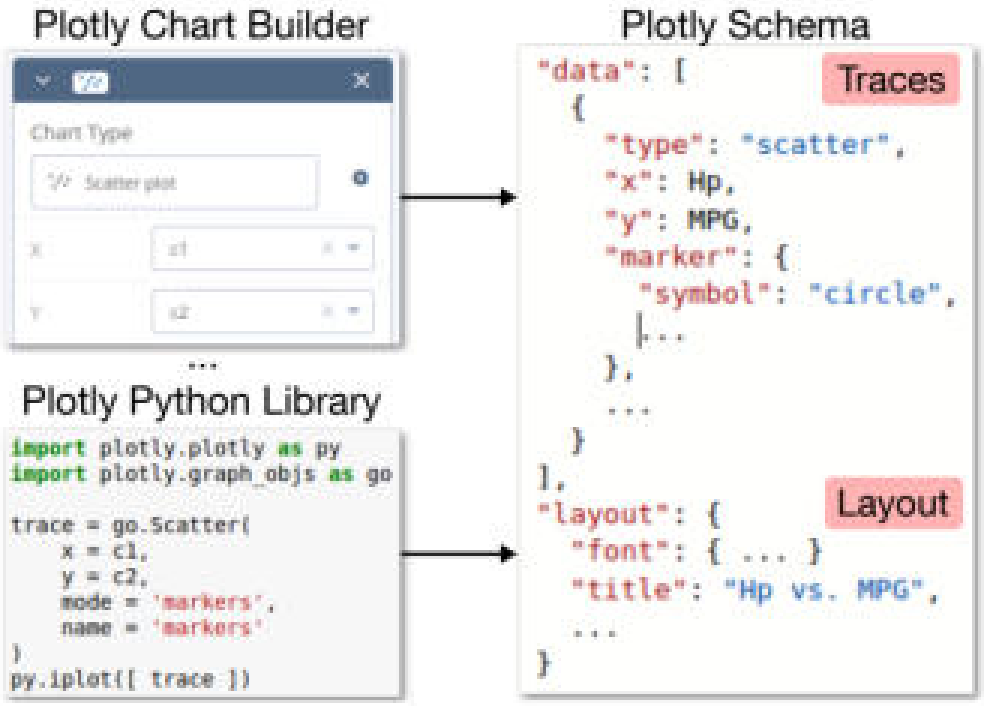

Рисунок 2.4 – Опису Plotly -візуалізації у форматі JSON

Vega-lite [1] та Vega [2], які є досить популярними схемами описування візуалізації, аналогічно дають змогу відтворювати складні графіки у виді JSON, проте залишають менший контроль.

Plotly підтримує спільне використання та роботу. Користувачі здатні

створювати діаграми в Plotly Community Feed [20], котрий володіє інтерфейсом для відшукання, розподіляння та фільтрації мільйонів візуалізацій, створених та завантажених користувачами (рис. 2.5).

Рисунок 2.5 – Скриншот загальнодоступного набору діаграм Plotly Community Feed

Plotly володіє REST API [19], при його допомозі для будь-якої візуалізації можна одержати: вихідну інформацію для візуалізації (table data, рис. 2.6),


```
{
  "JasonTing:2":{
    "reason":"restricted",
    "cols":{
      "Col2":{
        "uid":"946e9a",
        "order":1,
        "data":["203.58", "152.68", "254.47", "165.4", "267.19"]
      },
      "Col1":{
        "uid":"c3967e",
        "order":0,
        "data":["12.8", "17.8", "22.8", "27.8", "32.8"]
      }
    }
  }
}
```


Рисунок 2.6 – Приклад JSON із вихідними даними для візуалізації

Інформацію про спосіб та параметри візуалізації (chart data, рис. 2.7):


```
[
  {
    "mode":"markers",
    "uid":"0c920e",
    "type":"scatter",
    "name":"Col2",
    "ysrc":"JasonTing:2:946e9a",
    "xsrc":"JasonTing:2:c3967e"
  }
]
```


Рисунок 2.7 – Приклад JSON із інформацією щодо способу та параметрів візуалізації

Також за допомогою REST API для кожної візуалізації можна одержати і макет (layout, рис. 2.8) визначення конфігурації відображення.


```
{
  "width":1044,
  "height":600,
  "title":"Temperature vs Volume of Gas",
  "yaxis":{
    "type":"linear",
    "title":"Volume of Gas (cm^3)",
    "dtick":50,
    "autorange":false,
    "tickmode":"linear",
    "range":[ -50, 400 ] },
  "autosize":true,
  "xaxis":{
    "type":"linear",
    "title":"Temperature (\\u00b0C)",
    "dtick":5,
    "autorange":false,
    "tickmode":"linear",
    "range":[
      -20.2785396986,
      50.4249719973
    ]
  }
}
```


Рисунок 2.8 – Приклад JSON із інформацією про структура відображення

2.4 Опис даних

При проведенні цього дослідження застосовувався той самий НД, що у роботі про VizML [9, 24]. Його зібрали при допомозі Plotly API із загальнодоступних візуалізацій, і перш за все містив дані десь за 2,5 роки – від липня 2015 р. до січня 2018 р. Всього 2 359 175 об'єктів, 2 102 121 з них включали всі три параметри, котрі відтворюють візуалізацію, а 1989068 з них вдалося безпомилково проаналізувати. Одержаний НД Plotly має візуалізації, створені 143007 унікальними користувачами, які сильно відрізняються за своїм використанням. Розподіл кількості візуалізацій на користувача показано на рис. 2.9. Якщо усунути 0,1% юзерів-ботів, кожен юзер створив в середньому 6,86 візуалізацій, медіана кількості візуалізацій, що припадають на одного користувача, дорівнює 2.

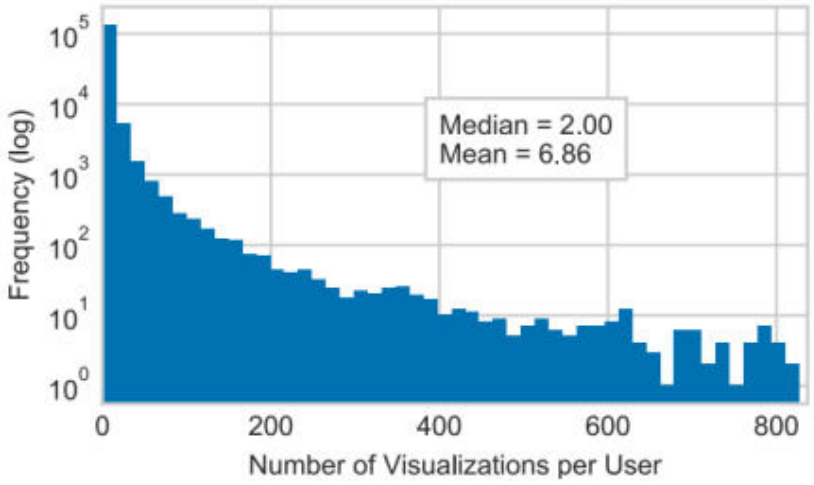

Рисунок 2.9 – Розподіл кількості графіків, що припадають на користувача у логарифмічно-лінійній шкалі

НД для візуалізації досить різняться за кількістю стовпців та рядків. Навіть якщо взяти до уваги, що окремі НД мають понад 100 стовпців, 94,97 % з яких мають ≤ 25 стовпців. Коли усунути НД з > 25 стовпцями, тоді середній НД матиме 4,75 стовпці, і медіанний НД матиме 3 стовпці. Їх розподіл у НД наведено на рис. 2.10,

розподіл рядків — на рис. 2.11.

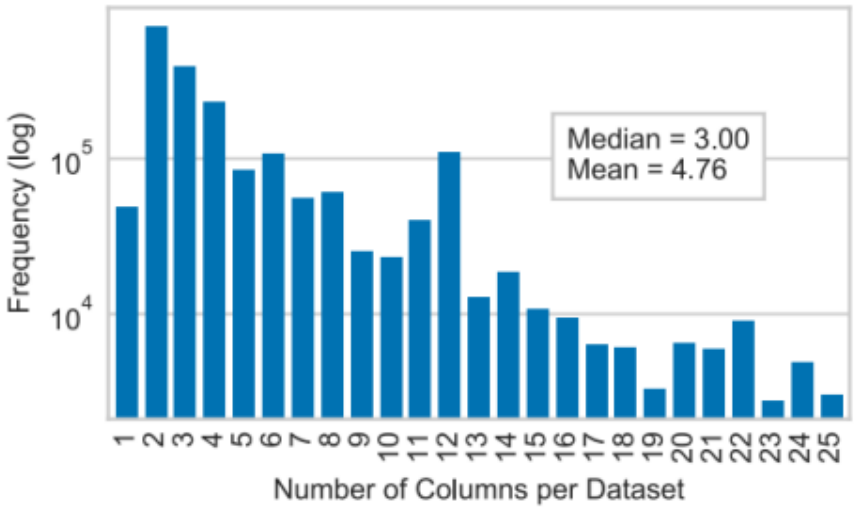

Рисунок 2.10 – Розподіл стовпців у наборі даних після видалення 5,03% НД з більш ніж 25 стовпцями, представлений у логарифмічно-лінійній шкалі

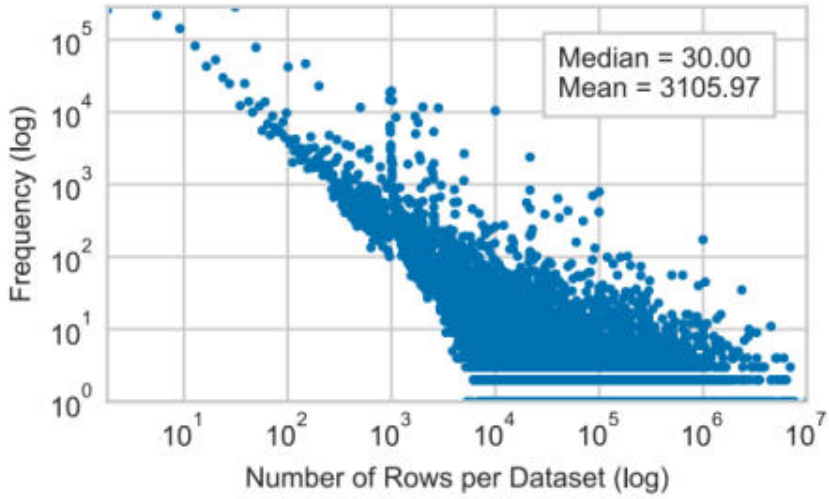

Рисунок 2.11 – Розподіл рядків за НД, візуалізований у логарифмічному масштабі

Такі розподіли із складними хвостами добре координуються з розподілами IBM ManyEyes і Tableau Public. Незважаючи на те, що Plotly дає змогу юзерам будувати візуалізації з використанням кількох НД, у 98,32% візуалізацій використовується лише один вихідний НД, тому нас цікавлять лише візуалізації із застосуванням одного НД. Більш того, для більш ніж 90% візуалізацій використовувалися всі стовпці вихідного НД.

2.5 Видобування ознак

Згідно з VizML, будь-який стовпчик з НД можна бути описаний 81 функцією, котрі поділені на 4 типи (табл. 2.1):

– розмірності (Dimensions) – це кількість рядків у стовпці;

– типів (Types), показують, чи є стовпчик категоріальним, непостійним чи кількісним;

– значень (Values), описують статистичні та структурні властивості значень у стовпці;

– назв (Names), визначають назву стовпчика.

Таблиця 2.1 – Функції, що описують параметри для даних одного стовпця

Розмірність (1)	
Довжина (1)	Кількість значень
Типи(8)	
Загальні (3)	Категоріальний (C), кількісний (Q), часовий (T)
Спеціальні (5)	Рядковий, логічний, цілий, раціональне число, дата і час
Значення (56)	
Статистичні [Q, T] (16)	Середнє значення, медіана, діапазон (необроблений/нормалізований по максимуму), дисперсія, стандартне відхилення, коефіцієнт дисперсії, мінімум, максимум, (25-й/75-й) процентиль, абсолютне медіанне відхилення, середнє абсолютне відхилення, кількісний коефіцієнт дисперсії
Розподіл [Q] (14)	Ентропія, коефіцієнт Джині, коефіцієнт асиметрії, ексцес, моменти (5-10), нормальність (статистика, p-значення) є нормальним при ($p < 0,05$, $p < 0,01$).

Продовження таблиці 2.1

Викиди (8)	(Має/%) викиди при (1,5 x IQR, 3 x IQR, 99% ile, 3σ)
Статистичні [C] (7)	Ентропія, (середнє/медіанне) значення довжини, (мін., станд., макс.) довжина значень, % моди
Послідовність (5)	Відсортована, монотонність, упорядкованість, (лінійний/ логарифмічний) коефіцієнт просторової послідовності
Унікальність (3)	(Є/кількість/%) унікальні(их) значення(ь)
Пропущені значення (3)	(Має/ кількість /%) відсутні(і) значення(і)
Назви (14)	
Властивості (4)	Довжина імені, кількість слів, кількість символів верхнього регістру починається з великої літери
Значення (10)	Містить символи («x», «y», «id», time», цифра, пробіл, «$», «€», «£», «¥») у назві

2.6 Архітектура застосунку

Проектування розробленого застосунку відбувалося згідно з вимогами Чистої архітектури. Це паттерн, котрий повинен максимально поділити код та модулі проекту на кілька шарів, при чому кожен з них має свої функції і не повинен перебувати у залежності від інших. Водночас, ці шари повинні визначити ту кількість модулів, котрі можливо застосувати незалежно від фреймворків, котрі застосовуються для написання ПЗ.

Візуальне відображення таких шарів наведено на рис. 2.12.

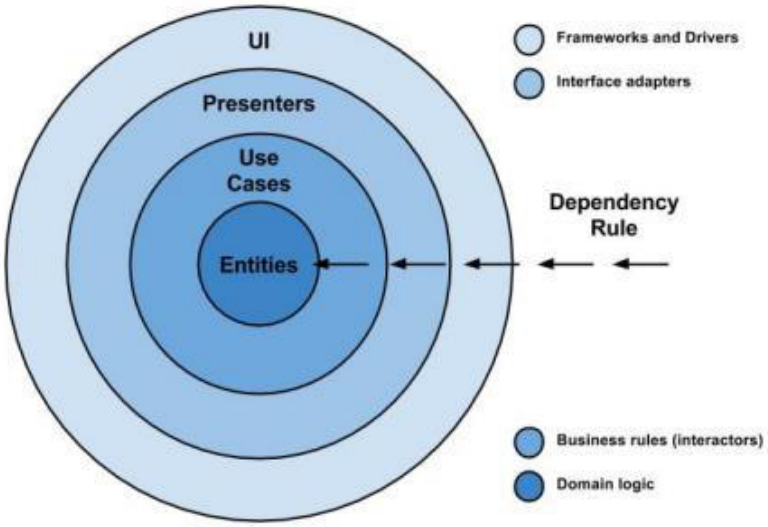

Рисунок 2.12 – Візуальне представлення Clean Architecture

Центральним є шар з Entities, самий незалежний шар програмного коду, де містяться всі класи даних, та ті класи, котрі ці дані власне і передають в одну чи другу сторону.

Наступний - це шар Use Cases. Він відповідальний за всю бізнес логіку ПЗ: будь-яке опрацювання даних і зведення їх до необхідного виду, обчислення і тд. .і т.п.

За шаром Use Cases йде шар Presenters, який є спеціалізованими класами, котрі застосовуються з метою зв’язування між собою шарів даних та шару UI (інтерфейсу користувача). Потрібно це щоб шар UI одержував готові дані негайно і не брав жодної участі у їх опрацюванні. Шар Presenters не має володіти інформацією щодо шарів даних та того, як з ними власне проводити взаємодію.

2.7 Діаграма варіантів використання застосунку

На рис. 2.13 наведено Use Case Diagram в нотаціях UML. На діаграмі є актор «Користувач», котрий показує функціонал користувачів застосунку. Виділено головний варіант використання – «Візуалізація даних». Кожен з інших варіантів є

певним способом використання. Отже, будь-який варіант використання відповідає логічності дій, щоб користувач зумів одержати визначений результат.

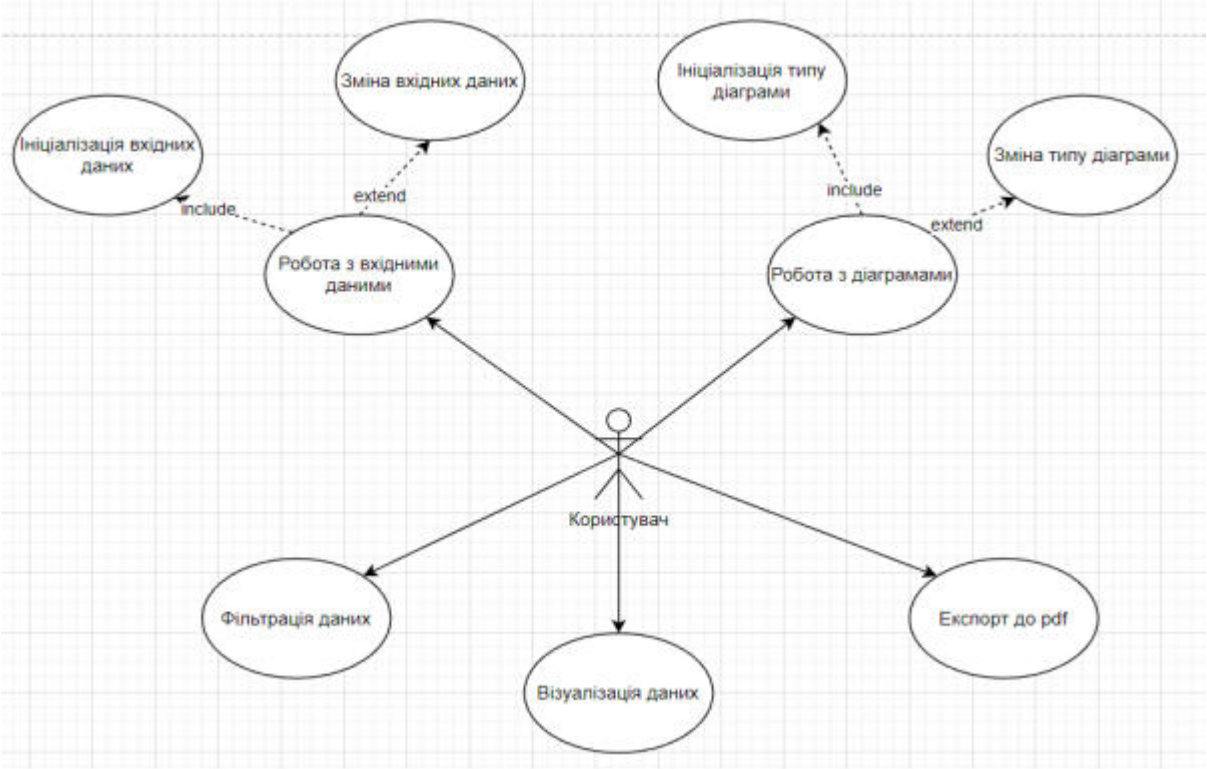

Рисунок 2.13 – Діаграма варіантів використання

2.8 Діаграма послідовності прецедента «Візуалізувати дані»

Наведена на рис. 2.14 діаграма містить такі основні елементи:

– об’єкти;
– повідомлення;
– лінія часу;
– загальні фрейми.

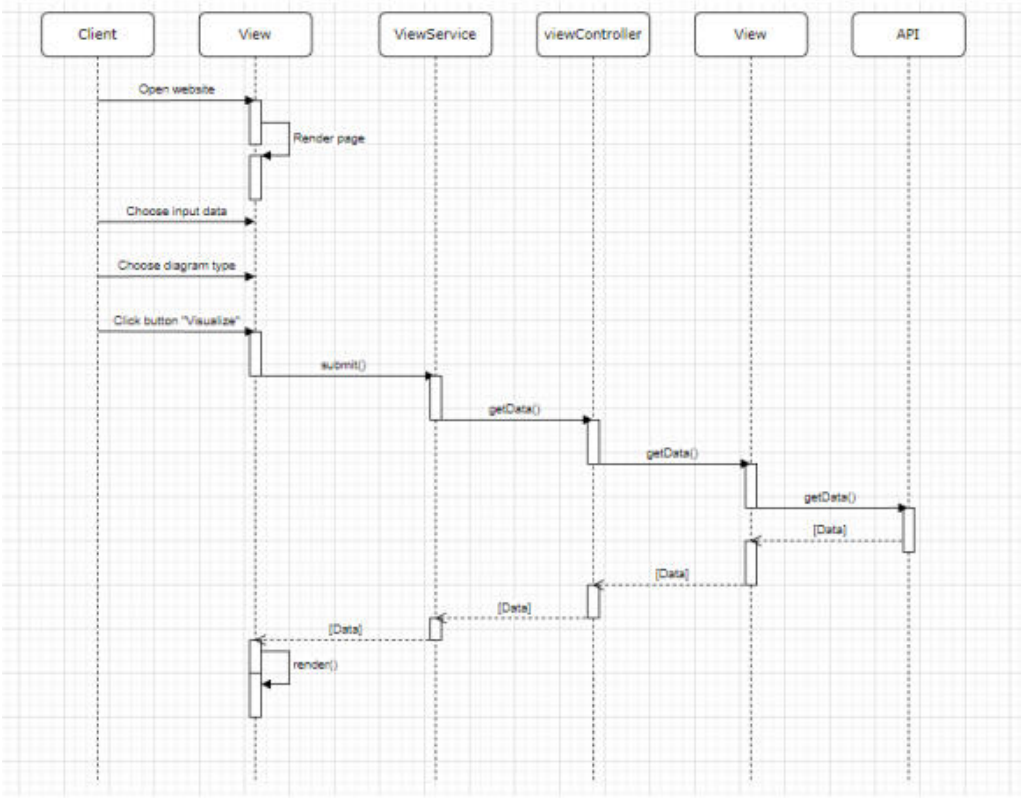

Рисунок 2.14 – Діаграма послідовності

2.9 Висновок до другого розділу

У цьому розділі дизайн візуалізації розглядається як значення, що максимізує ефективність візуалізації з урахуванням НД та контекстуальних факторів. Метою рекомендації візуалізацій є зниження вартості їх створення за рахунок автоматичної пропозиції підмножини варіантів дизайну візуалізацій.

НД, що використовується в роботі, був одержаний при допомозі Plotly REST API. Також досліджені 4 категорії функцій, котрі застосовуються для отримання ознак даних.

Архітектура застосунку базується на рекомендаціях Clean Architecture. Наведень Use Case Diagram та діаграма послідовності одного із прецедентів.

3 ПРАКТИЧНА ЧАСТИНА

Розв'язання задачі побудови візуалізації за НД відбувається у 2 етапи:

– встановлення типу візуалізації;

– встановлення осей для відтворення інформації.

Обидва етапи є завданнями класифікації. Аби вирішити обидва завдання застосовувалися класичні алгоритми машинного навчання. Такі методи дають змогу одержати задовільні візуалізації та виявити найважливіші атрибути, на базі яких і було визначено рішення щодо типу візуалізації.

З НД, описаного в розділі 2.2 було отримано два набори ознак для проведення експериментів. Перший НД (single_column_features) включає ознаки для кожного окремого стовпця, описані в таблиці2.1. Другий НД (pairwise_column_agg_features) включає ознаки кожного НД для візуалізації, описані в табл. 2.2 і 2.3 відповідно

3.1 Підготовка даних

З тих даних, котрі описані у другому розділі, для проведення експерименту було випадково обрано 20.000 об'єктів. Приклад подання об'єкта є у розділі 2.1.

З цього НД були усунені діаграми користувачів, котрі, можливо, побудовані з одного НД. Після цього було запущено алгоритм, який витягує з даних ознаки, описані у розділі 2.3. У процесі отримання ознак траплялися помилки преведення типів даних. Також не для всіх об'єктів набору вдалося отримати інформацію щодо типу візуалізації. Як наслідок отримано НД з 2622 екземплярів.

Набір має дані для візуалізації таких типів: точкова діаграма; стовпчикова діаграма; лінійна діаграма; гістограма; діаграма розмаху; теплова карта.

Не беручи до уваги поширеність кругової діаграми, вона була усунена з

дослідження. Число діаграм для кожного типу наведено на рис. 3.1.

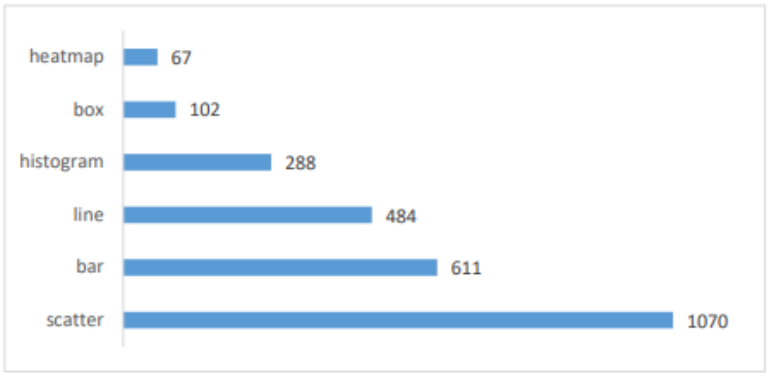

Рисунок 3.1 – Кількість діаграм різних типів у НД pairwise_column_agg_features

За даною вибіркою було обчислено ознаки та одержаний НД pairwise_column_agg_features.

У подальшому з кожного НД, котрий відображає візуалізацію, було вилучено дані всіх стовпчиків. Стовпчик даних є того самого типу візуалізації, як і весь набір. Для кожного вилученого стовпця даних були обчислені ознаки і одержаний НД single_column_features (рис. 3.2).

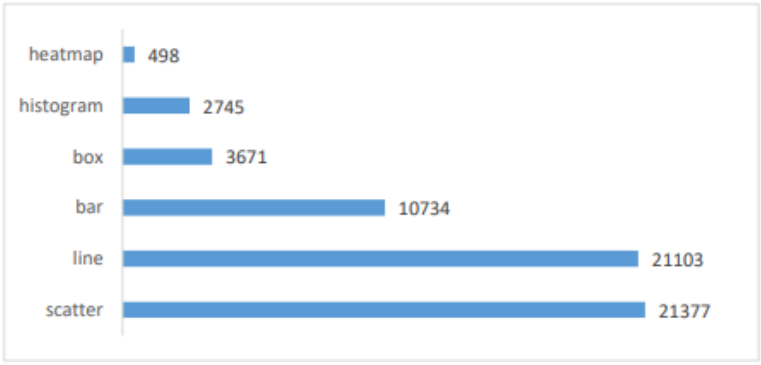

Рисунок 3.2 – Кількість стовпців діаграм різних типів у НД single_column_features

3.2 Встановлення виду візуалізації

Для вирішення цього завдання застосовувалися такі методи машинного

навчання: дерево прийняття рішень; випадковий ліс; модель градієнтного бустингу; бегінг модель.

Для одержання оцінки результатів функціонування алгоритму окрім класичних метрик, таких як accuracy, precision, recall, матриця помилок, застосовувалася метрика top2-accuracy (частка правильних варіантів візуалізації, котрі потрапили у перші два передбачувані варіанти).

У силу того, що розподіл даних є за типами візуалізації є незбалансованим, були застосовані техніки оверсеплінгу та андерсемплінгу з модуля imblearn для підтримування балансу класів [9, 25].

Дерево прийняття рішень. Робить прогноз величини цільової змінної через формування нескладних правил прийняття рішень, виведених із ознак. У вузлах дерева є ознаки об'єктів; ребра дерева мають різні величини ознак, від яких власне і перебуває у залежності цільова функція; листки є значеннями цільової величини - мітки класів.

Даний алгоритм має наступні переваги: дозволяє легко отримати інтерпретовану модель, є ефективним щодо використання до тих даних, котрі мають пропущені значення. Також є низка гіперпараметрів, котрі здатні чинити вплив на узагальнюючу здатність алгоритму. Найчастіше настроюваними гіперпараметрами є глибина дерева, мінімально допустиме число елементів у листі і мінімально допустиме число елементів, котре потрібно для розбивання вузла.

Самі кращі результати алгоритму були одержані за наступних гіперпараметрів: 'max_depth' = 12, 'max_features': 'auto', 'min_samples_leaf = 1, 'min_samples_split' = 7.

Випадковий ліс. Є методом машинного навчання, в основі якого лежить застосування ансамблю дерев рішень. Рішення щодо належності об'єкта до певного класу приймається мажоритарним голосуванням. У своїй більшості застосувань такий підхід демонструє більш точний результат, аніж саме дерево прийняття рішень. Недоліком даного методу є більш тривалий процес навчання і, у результаті цього, дещо великий розмір одержуваної моделі, але алгоритм піддається розпаралелюванню. Ключовими гіперпараметрами даного методу крім тих, котрі

описують дерево рішень є кількість таких дерев.

Самі кращі результати методу були одержані за таких значень гіперпараметрів: 'n_estimators'=140, 'max_depth'= 12, 'max_features': 'auto', 'min_samples_split' = 7.

Градієнтний бустинг. Базою даного методу аналогічно є дерева рішень. Для градієнтного бустингу навчання ансамблю проводиться послідовно, іншими словами будь-яке наступне дерево формується з метою уточнення попередніх: для кожної ітерації розраховується відхилення передбачень ансамблю, котрий був навчений раніше, а наступна модель, котру буде додаватися до ансамблю, буде навчатися. Отже, якщо додавати відповіді нового дерева до уже існуючих відповідей ансамблю, тоді буде зменшуватися властиво середнє відхилення моделі. Власне нові дерева будуть додаватися до ансамблю до тих пір, поки помилка зменшуватиметься, або ж допоки не справдиться одна з умов ранньої зупинки.

Значення гіперпараметрів, за яких були одержані найкращі результати методу, є такими: 'n_estimators'=120, 'max_depth'= 12, 'max_ learning_rate'=1.0.

Бегінг. Є методом формування ансамблю моделей, за якого навчання базових моделей проходить паралельно. За даних обставин кожна модель проходить навчання на своїй вибірці, котра сформована із вихідного НД при допомозі алгоритму бутстрапу.

Беггінг дозволяє покращити точність та стійкість роботи методів машинного навчання, шляхом зменшення дисперсії помилки та зменшення ефекту перенавчання.

Результати методу, які є найкращими, було одержано при таких величинах гіперпараметрів: 'n_estimators'=10, 'max_samples'= 0.7, 'max_features'=0.75.

Результати функціонування методів НД single_column_features (табл. 3.1). Попереду використання алгоритмів класифікації щодо тренувальних даних використовувася алгоритм андерсемплінга AllKNN [5] з метою зменшення числа екземплярів класів стовпчикової, лінійної та точкової діаграм, водночас для збільшення числа примірників – застосовувалися діаграми розмаху, гістограми та теплової карти - методи SMOTE [17], ADASYN SVMSMOTE [6, 7], SMOTETomek

[10] (табл. 3.1).

Таблиця 3.1 – Результати роботи алгоритмів на НД колекції

Алгоритм	Тип оверсемплінгу	top2-accuracy
Дерево прийняття рішень	Без оверсемплінгу	0.757
	ADASYN	0.761
	SMOTE	0.754
	SVMSMOTE	0.768
	SMOTETomek	0.758
Випадковий ліс	Без оверсемплінгу	0.814
	ADASYN	0.828
	SMOTE	0.823
	SVMSMOTE	0.826
	SMOTETomek	0.817
Градієнтний бустинг	Без оверсемплінгу	0.855
	ADASYN	0.858
	SMOTE	0.867
	SVMSMOTE	0.866
	SMOTETomek	0.854
	Без оверсемплінгу	0.763
	ADASYN	0.773
	SMOTE	0.774
	SVMSMOTE	0.778
	SMOTETomek	0.771

Матриці помилок для методів, з використанням яких було отримано найкращий результат – рис. 3.3.

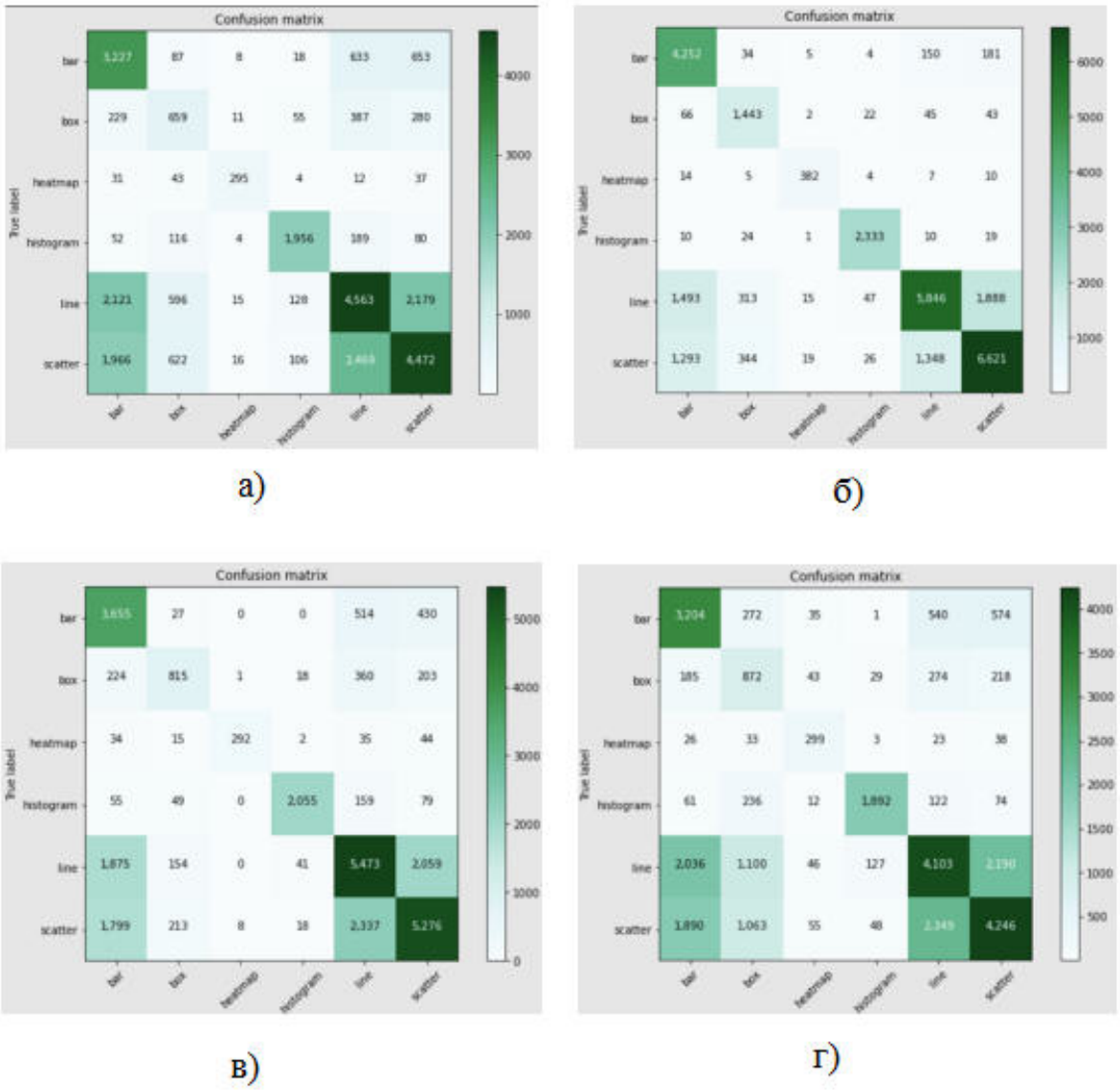

Рисунок 3.3 – Матриці помилок: а) дерева прийняття рішень з оверсемплінгом SVMSMOTE; б) градієнтного бустингу з оверсемплінгом SMOTE; в) випадкового лісу з оверсемплінгом ADASYN; г) бегінга з оверсемплінгом SVMSMOTE

Особливістю обраних алгоритмів є можливість вилучення та ранжирування ознак якраз по їх важливості, та оцінювання відсоткового співвідношення вкладення будь-якого з них у кінцевий результат. На рис. 3.4 видно, що найважливішими ознаками є min_value_length, length, moment_5.

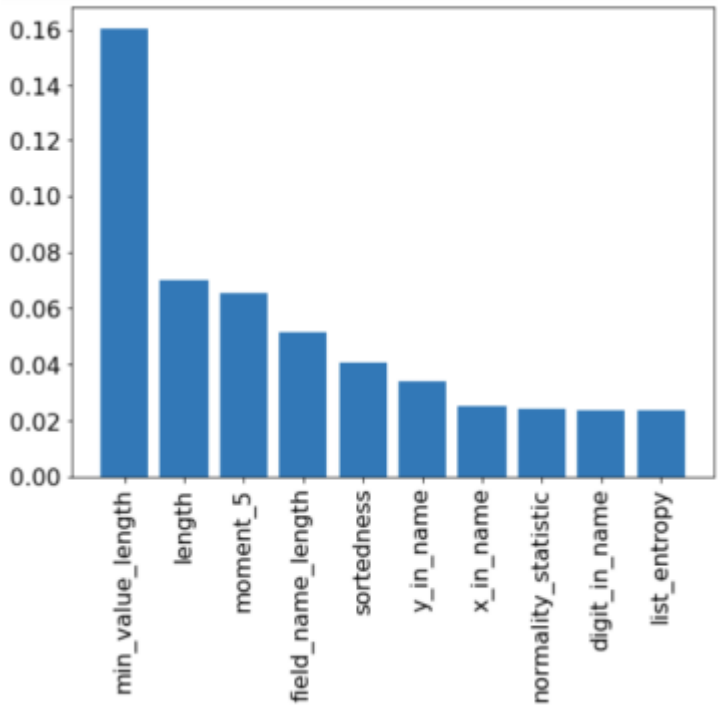

Рисунок 3.4 – Важливість ознак у методі градієнтного бустингу з оверсемплінгом SMOTE

Результати функціонування методів на НД pairwise_column_agg_features. Аналогчно до попереднього НД до використання алгоритмів класифікації щодо тренувальних даних, використовувалися алгоритми оверсеплінгу SMOTE, ADASYN, SVMSMOTE, SMOTETomek (табл. 3.2)

Таблиця 3.2 – Результати функціонування алгоритмів на НД pairwise_column_agg_features

Алгоритм	Тип оверсемплінгу	top2-accuracy
Дерево прийняття рішень	Без оверсемплінгу	0.820
	ADASYN	0.839
	SMOTE	0.842
	SVMSMOTE	0.824
	SMOTETomek	0.818

Продовження таблиці 3.2

Алгоритм	Тип оверсемплінгу	top2-accuracy
Випадковий ліс	Без оверсемплінгу	0.887
	ADASYN	0.898
	SMOTE	0.893
	SVMSMOTE	0.901
	SMOTETomek	0.894
Градієнтний бустинг	Без оверсемплінгу	0.864
	ADASYN	0.879
	SMOTE	0.873
	SVMSMOTE	0.869
	SMOTETomek	0.871
Бегінг	Без оверсемплінгу	0.862
	ADASYN	0.871
	SMOTE	0.873
	SVMSMOTE	0.869
	SMOTETomek	0.870

Матриці помилок для алгоритмів, на яких було отримано найкращий результат (рис. 3.5):

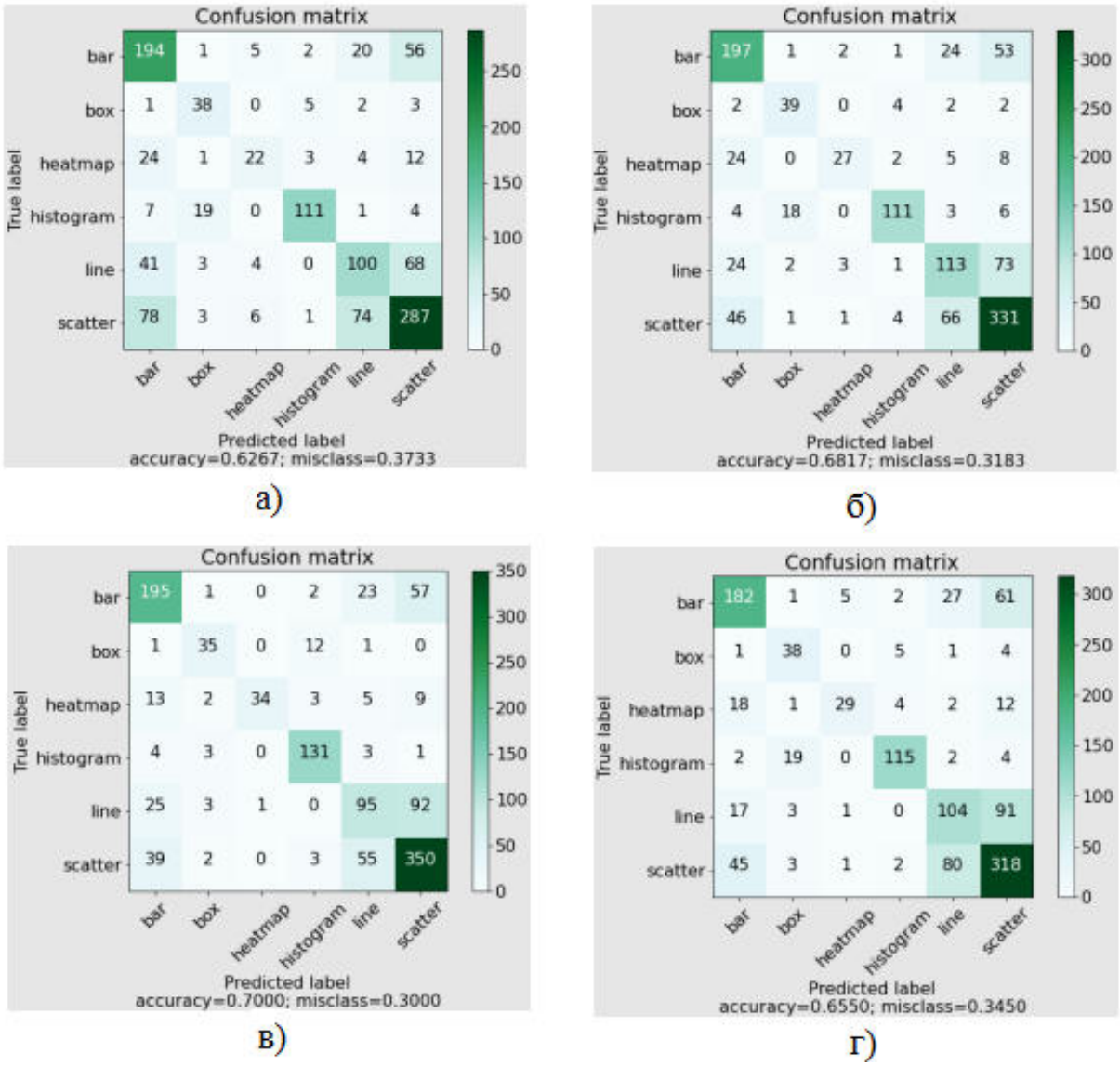

Рисунок 3.5 – Матриці помилок: а) дерева прийняття рішень з оверсемплінгом SMOTE; б) градієнтного бустингу з оверсемплінгом ADASYN; в) випадкового лісу з оверсемплінгом SVMSMOTE; г) бегінга з оверсемплінгом SMOTE

Для цього НД найбільш важливими атрибутами є y_in_name-agg-has, x_in_name-agg-has, mrmaHzed_edit_distance-agg-mean (рис. 3.6):

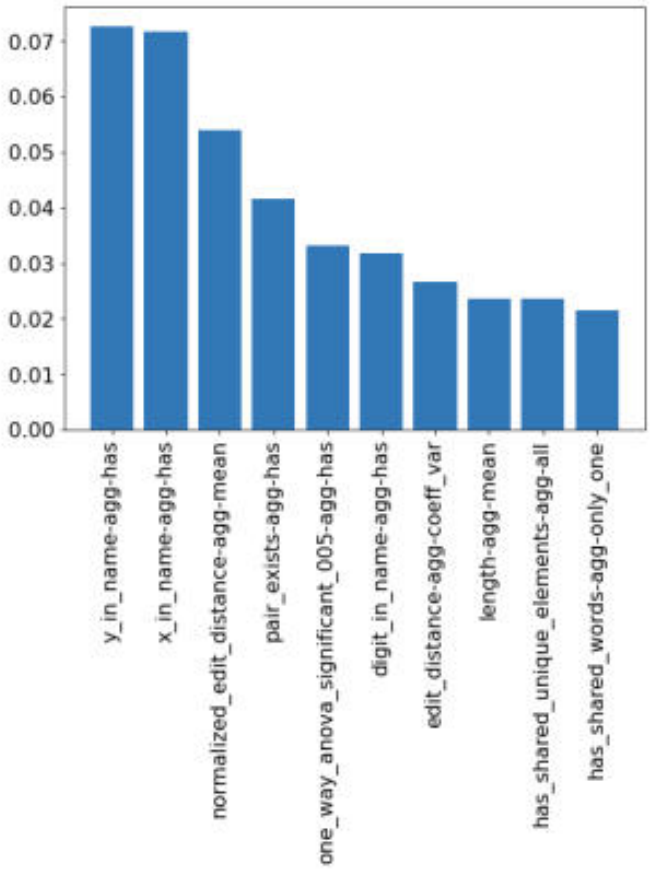

Рисунок 3.6 – Важливість ознак для алгоритму випадкового лісу з оверсемплінгом SVMSMOTE

Проаналізувавши одержані матриці помилок для обох НД, можна стверджувати, що найбільш склдано поділяються класи точкової, лінійної та стовпчикових діаграм.

3.3 Встановлення осі для відтворення даних

Для вирішення задачі застосовувалися аналогічні алгоритми методи машинного навчання, як і для встановлення типу візуалізації. Для навчання моделей використано НД single_column_features.

Також було використано алгоритми оверсеплінга SMOTE, ADASYN,

SVMSMOTE для підтримування балансування класів, так як клас із міткою осі у має перевагу над класом із міткою осі x. Кожен із алгоритмів навчався на вихідному НД, а також потім використано оверсемплінг до цих методів (табл. 3.3).

Таблиця 3.3 – Результати функціонування методів на НД single_column_features

Алгоритм	Тип оверсемплінга	AUC	Precision, recall
Дерево прийняття рішень	Без оверсемплінга	0.954	[0.89842927, 0.88892135] [0.82407808, 0.91332486]
	ADASYN	0.951	[0.91921654, 0.88352668] [0.83530184, 0.94450301]
	SMOTE	0.951	[0.90273319, 0.8935334] [0.85334646, 0.93048924]
	SVMSMOTE	0.957	[0.90057974, 0.89356879] [0.85367454, 0.92875302]
Випадковий ліс	Без оверсемплінга	0.959	[0.90416531, 0.87212676] [0.8172769,0.92485831]
	ADASYN	0.965	[0.92449664, 0.89861724] [0.85867782, 0.94698332]
	SMOTE	0.965	[0.92478542, 0.89769656] [0.85720144, 0.94729336]
	SVMSMOTE	0.964	[0.91158005, 0.89686072] [0.85744751, 0.93712408]
Градієнтний бустінг	Без оверсемплінга	0.957	[0.89943753, 0.90123606] [0.86564961, 0.92683078]
	ADASYN	0.955	[0.89795572, 0.90284332] [0.86827428, 0.9254046]
	SMOTE	0.951	[0.89682742, 0.90049511] [0.8648294,0.92478452]
	SVMSMOTE	0.954	[0.88804968, 0.90184722] [0.86794619, 0.91728158]

Продовження таблиці 3.3

Алгоритм	Тип оверсемплінга	AUC	Precision, recall
Бегінг	Без оверсемплінга	0.962	[0.92380867, 0.88949697] [0.84432415, 0.94735537]
	ADASYN	0.963	[0.91933476, 0.89136164] [0.84785105, 0.94375891]
	SMOTE	0.963	[0.91445868, 0.89545535] [0.85490486, 0.93954238]
	SVMSMOTE	0.962	[0.91087109, 0.89256394] [0.85080381, 0.93706207]

Матриці помилок для алгоритмів, на яких було отримано найкращий результат (рис. 3.7)

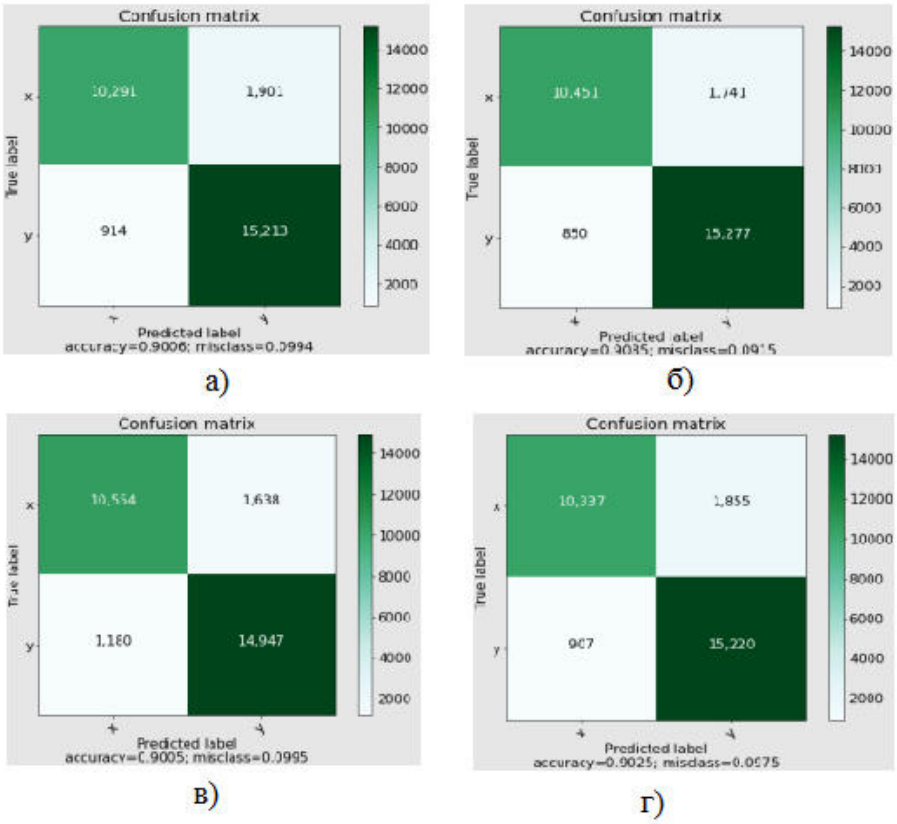

Рисунок 3.7 – Матриці помилок: а) дерева прийняття рішень з оверсемплінгом SVMSMOTE; б) випадкового лісу з оверсемплінгом SMOTE; в) градієнтного бустингу без оверсемплінгу; г) бегінга з оверсемплінгом ADASYN

Для розв’язання задачі встановлення осі для відтворення даних найважливішими параметрами є y_in_name, x_in_name, is_sorted (рис. 3.8).

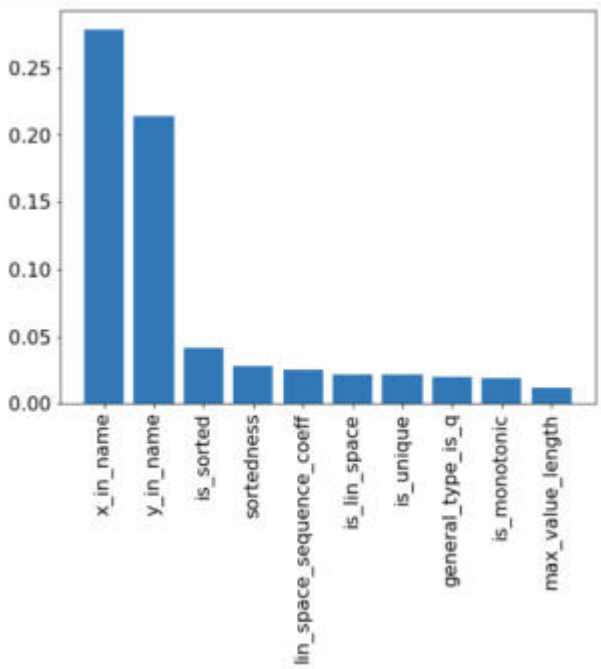

Рисунок 3.8 – Важливість ознак у методі випадковий ліс із використанням оверсемплінгу SMOTE

Отримані результати функціонування алгоритмів незначно різняться, також для випадку градієнтного бустингу використання техніки оверсемплінга не покращило показники метрик , якщо порівнювати із результатами, котрі одержані на вихідному НД. Самий кращий результат при встановленні осі для відтворення даних було одержано для випадкового лісу із використанням оверсемплінга SMOTE.

3.4 Робота із веб-застосунком

З метою візуалізації результатів функціонування алгоритму було розроблено веб-застосунок із архітектурою «клієнт-сервер». Інтерфейс програми реалізований

мовою програмування Dart, як бібліотека для візуалізації даних застосвувалася Highcharts JS. Сервер програми реалізований мовою програмування Python за допомогою фрейморка Flask.

Розроблене ПЗ дає змогу завантажити дані у представленнях JSON або CSV. Після виконання процесу завантаження дані можна редагувати (рис. 3.9).

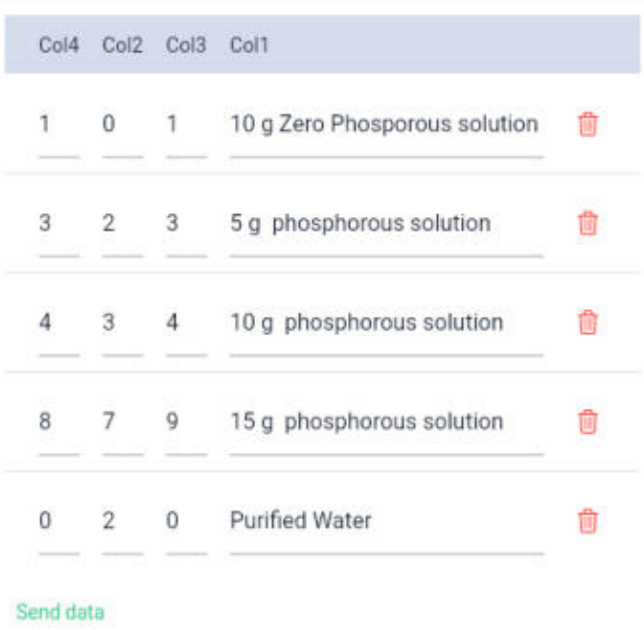

Рисунок 3.9 – Екран застосунку із даними, доступними для редагування

Опісля відправлення дані проходять обробку на сервері:

– розраховуються потріьні ознаки для всього НД та кожного стовпчика даних;

– при допомозі алгоритмів градієнтного бустингу із застосуванням методу оверсемплінгу SMOTE та випадкового лісу з SVMSMOTE встановлюються два найкращі типи візуалізації;

– при допомозі алгоритму випадкового лісу із оверсемплінгом SMOTE встановлюються осі для кожного стовпчика даних;

– створюється відповідь клієнту у JSON-форматі з тією інформацією, котра необхідна для візуалізації.

Прикладом того, що відповість сервер, може бути фрагмент коду на рис. 3.10.

```
{
  "type": "box",
  "legend": [
    {
      "name": "Col 1",
      "axis": "x"
    },
    {
      "name": "Col 2",
      "axis": "y"
    },
    {
      "name": "Col 3",
      "axis": "y"
    }
  ]
}
```


Рисунок 3.10 – Відповідь сервера після опрацювання даних

На підставі відповіді, котра одержана від сервера, створюються та показуються юзеру ті діаграми, котрі встановлені алгоритмом найбільш вдалими для завантажених даних (рис. 3.11 - 3.13).

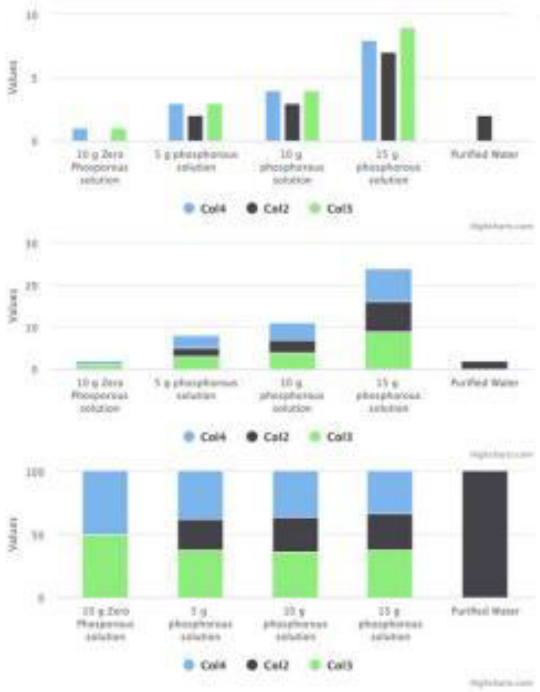

Рисунок 3.11 – Приклад роботи застосунку

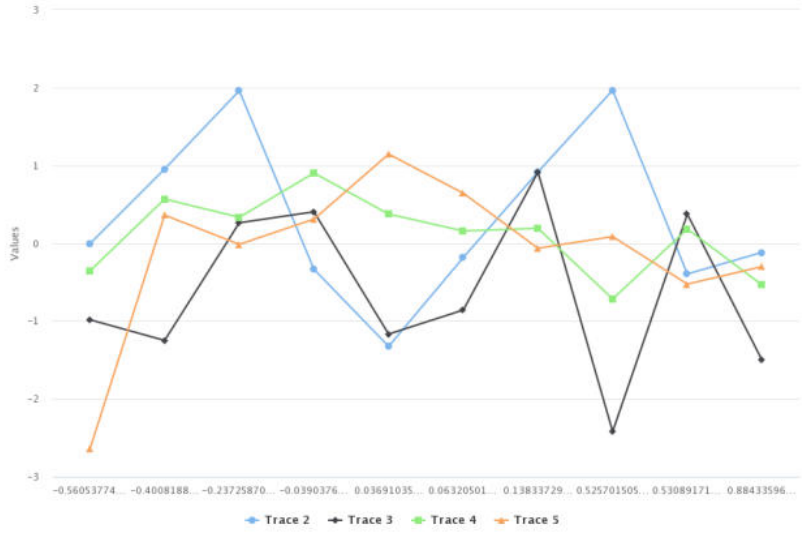

Рисунок 3.12 – Приклад роботи застосунку

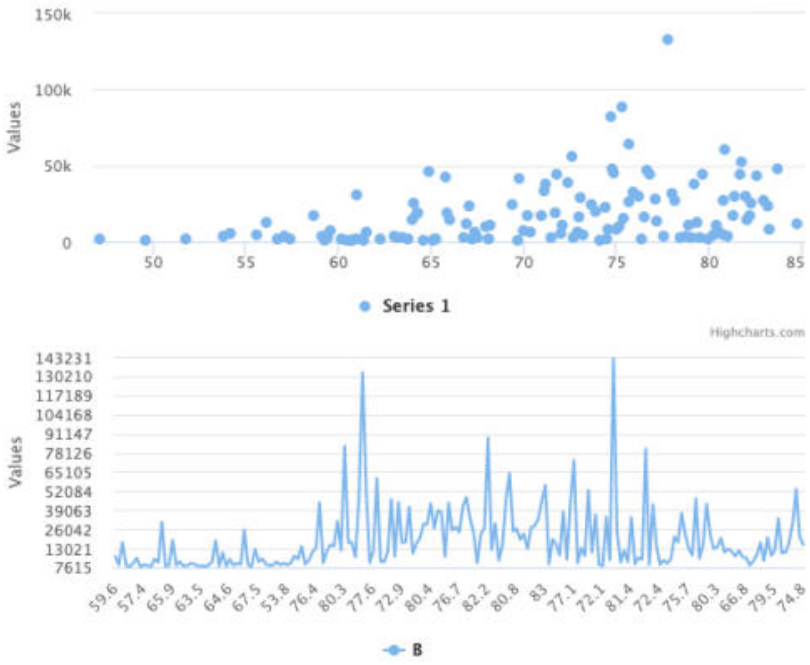

Рисунок 3.13 Приклад роботи застосунку

3.5 Висновок до третього розділу

Результатом проведених експериментів (із використанням алгоритмів машинного навчання та методів оверсемплінгу і андерсемплінгу для вирішення

проблеми незбалансованості даних) є надання рекомендацій для візуалізацій складних даних.

Одержані моделі володіють певними неточностями щодо класів точкової, лінійної і стовпчикових діаграм. Це може бути результатом прогалин у знаннях контексту створення візуалізації, саме тому для рекомендації візуалізацій використовуються два найбільш ймовірні типи візуалізації, котрі запропонував алгоритм.

Для відображення функціонування алгоритму було написано веб-застосунок, що дозволяє за наявним НД встановити ті типи візуалізації, котрі найкраще підходять.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ

4.1 Охорона праці

Метою кваліфікаційної роботи магістра є розробка ПЗ для рекомендації візуалізацій запропонованого набору складних даних. Оскільки, проведення робіт із розробки та використання ПЗ передбачає застосування комп’ютерної техніки, зокрема ПК та периферійних пристроїв, то обов’язковим є дотримання вимог з охорони праці і техніки безпеки.

Для ефективної і безпечної роботи колективу працівників з розробки ПЗ комп’ютерних систем, в тому числі і фахівців з підвищення ефективності контролю доступу в приміщення, необхідно організувати безпечні умови праці. При цьому керівник організації несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці [32]. Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями». Згідно Вимог приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до цих вимог;

– переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 22.08.2005 N 161, зареєстрованого в Міністерстві юстиції України 05.09.2005 за N 990/11270 (НАПБ Б.06.004-2005);

– Державних будівельних норм "Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд", затверджених наказом Держбуду України від 28.10.98 N 247 (далі - ДБН В.2.5-56:2014, з димовими пожежними

сповіщувачами та переносними вуглекислотними вогнегасниками.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця операторів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників». Загальні технічні вимоги і з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог НАПБ А.01.001-2014. Приміщення, в яких розміщуються робочі місця операторів сервера загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.2.5-56:2014, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними.

Лінія електромережі для живлення комп’ютера та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам НПАОП 40.1-1.01-97.

У приміщенні, де одночасно експлуатуються понад п’ять комп’ютерів, на помітному, доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Комп’ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз’єднання при відключенні має бути зворотним. Не допускається підключати комп’ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення комп’ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При підвищенні ефективності контролю доступу в приміщення, де для забезпечення безпеки мешканців, співробітників і збереження майна використовуються ДС, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18. Організація робочого місця фахівця із дослідження методів та програмно-апаратних засобів оптимізаційних процесів на основі ГА повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги». Відстань від екрана до ока фахівців, які працюють за комп’ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрана комп’ютера,

зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля згідно з вимогами ДСанПіН 3.3.2.007-98.

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп'ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи фахівців розробка ПЗ для рекомендації візуалізацій складних даних.

4.2. Комп'ютерне забезпечення процесу оцінки радіаційної та хімічної обстановки

Екологічне співтовариство розробило сімейство інструментів комплексної екологічної оцінки. Програмне забезпечення і послуги (ESS), комерційна група IIASA, включаючи AirWare (для повітряних проблеми якості), WaterWare (для якості води), CityWare (якість повітря і води в контексті великих міст) і EIAxpert (для надання допомоги із загальним впливом на навколишнє середовище). Функціональність в цілому схожа на RAISON, хоча з великим акцентом на моделювання і меншим акцентом на керування даними. Знову ж таки, інструменти ESS розроблені як модульні набори інструментів (доступні спеціальні системи для вирішення конкретних завдань). Компоненти включають стандартні імітаційні моделі, включаючи моделі ISC і PBM Агентства з охорони навколишнього середовища США, управління даними, в тому числі ГІС, аналіз даних (наприклад, аналіз часових рядів даних спостережень), візуалізація, а також оптимізація [33].

Іноді немає готових моделей, придатних для конкретного застосування, але тягар розробки нової програми на Фортрані або C / C ++ є надмірним. Розробка моделі оточення може відносно легко реалізувати власні моделі комп'ютерів і не турбуватися про включення процедур для вирішення рівнянь, візуалізації і т. д. Як правило, за допомогою цих інструментів користувач просто повинен вказати свою модель, використовуючи або математичні рівняння, або спеціальні графічні

символи або значки, які безпосередньо представляють поведінку системи.

На даний момент є розроблені моделі комп’ютерного забезпечення процесу для оцінки радіаційної та хімічної обстановки.

GEMS – це система на основі моделей, яка підтримує оцінки схильності і ризику, надаючи доступ до одиночних і мультимедійних моделям експозиції, фізико-хімічні властивості методи оцінки, статистичний аналіз, графічні та картографічні програми з відповідними даними на навколишнє середовище, джерела, рецептори і популяції. У розробці з 1981 року, GEMS надає аналітикам 84 84 інтерактивний, легко досліджуваний інтерфейс для різних моделей, програм і даних, які необхідні для оцінки хімічного впливу і ризику [33].

HSPF – це комплексний пакет для моделювання кількості і якості стоків з багатоцільових водозборів і процесів радіації, що відбуваються в потоках або повністю змішаних озерах. Це дозволяє інтегроване моделювання землі і грунту, процесів забруднення при гідравлічній і осадово-хімічній взаємодії. Результатом моделювання є тимчасові дані витрати стоку, концентрація поживних речовин і пестицидів, а також дані кількості і якості води в будь-якій точці водозбору. Алгоритми якості води включають динаміку BOD / DO, вуглець, азот і фосфор. Процеси трансформації, які включені в модель це: гідроліз, фотоліз, окислення, випаровування, сорбція і биодеградація. Вторинні або «дочірні» хімічні речовини також моделюються.

Вимоги до даних для моделі можуть бути досить широкими в залежності від конкретного застосування.

Модель MMSOILS – це методологія оцінки впливу на людину і ризику для здоров'я, пов'язаних з викидами забруднень з небезпечних відходів. Мультимедійна модель, що стосується перенесення хімічної речовини в ґрунтові води, поверхневі води, атмосферу і накопичення в їжі. Шляхи впливу на людину, які розглянуті в методології включають: потрапляння в грунт, вдихання летких речовин в повітря і тверді частинки, шкірний контакт, прийом питної води і т.д. Ризик, пов'язаний із загальною дозою опромінення, розраховується на основі хімічної токсичності [33].

4.3 Висновок до четвертого розділу

В цьому розділі проаналізовано важливі питання охорони праці та безпеки в надзвичайних ситуаціях, висвітлено питання комп’ютерного забезпечення процесу оцінки радіаційної та хімічної обстановки.

ВИСНОВОК

У процесі написання роботи було вивчено існуючі системи рекомендації візуалізацій, виявлено недоліки систем з урахуванням правил. Було розглянуто основні ідеї та алгоритми, котрі є основою даних систем, а також проведено порівняльний аналіз систем, які застосовують методи машинного навчання.

Для дослідження опрацьовано НД, отриманий із загальнодоступного джерела візуалізацій Plotly Community Feed. Наведено 4 категорії функцій, котрі застосовуються для отримання ознак даних. Окрім тих функцій, котрі прикладаються до одного стовпчика, для отримання ознак використовувалися функції, які застосовуються до пари стовпчиків даних, та ті функції, котрі агрегують отримані результати.

Завершальним етапом виконаного дослідження стала реалізація веб-застосунку, що пропонує візуалізації для завантаженого НД. Створене ПЗ дає змогу змінювати дані перед візуалізацією, завантажувати отримані діаграми. Варто зазначити, що цей застосунок може бути легко розширений, за необхідності, для функціонування із графіками інших типів.

Попри те, що алгоритми, котрі використовуються в застосунку, не повністю здатні підібрати найбільш гожу візуалізацію, вони надають достатньо ефективні уявлення, котрі допомагають дослідити, виявити аспекти і зробити самостійні висновки за тими даними, котрі були ними надані.

Напрямок подальших досліджень:

– використання можливостей глибокого навчання;

– збільшення НД за допомогою Plotly REST API за рахунок нових примірників та результатів функціонування розробленого ПЗ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-Lite: A Grammar of Interactive Graphics. IEEE Transactions on Visualization and Computer Graphics, 23(1), pp. 341—350, Jan. 2017

2. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. IEEE TVCG (Proc. InfoVis), 2016

3. Каїро Альберто Функціональне мистецтво: вступ до інфографіки та візуалізації. Видавництво Українського Католицького Університету, 2017. 350 с.

4. Gnanamgari, S.: Information presentation through default displays. Ph.D. dissertation, Philadelphia,PA,USA (1981)

5. Stefanyshyn, I. , Pastukh, O., Stefanyshyn, V. , Baran, I. , Boyko, I. Robustness of AI algorithms for neurocomputer interfaces based on software and hardware technologies CEUR Workshop Proceedings, 2024, 3742, pp. 137–149.

6. H. M. Nguyen, E. W. Cooper, K. Kamei, “Borderline over-sampling for imbalanced data classification,” International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1), pp.4-21, 2009.

7. Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322—1328. IEEE, 2008.

8. Hanrahan, P.: Vizql: a language for query, analysis and visualization. In: Proceedings of the 2006 ACM SIGMOD international conference on Management of data. ACM (2006)

9. Hu, Kevin & Bakker, Michiel & Li, Stephen & Kraska, Tim & Hidalgo, Cesar. (2018). VizML: A Machine Learning Approach to Visualization Recommendation.

10. Ivan Tomek. An experiment with the edited nearest-neighbor rule. IEEE Transactions on systems, Man, and Cybernetics, 6(6), pp.448—452, 1976.

11. K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2016.

12. Kaur, Pawandeep & Owonibi, Michael. (2017). A Review on Visualization Recommendation Strategies. 10.5220/0006175002660273.

13. Kubernátová, Petra & Friedjungová, Magda & van Duijn, Max. (2019). Constructing a Data Visualization Recommender System.

14. Luo, Yuyu & Qin, Xuedi & Tang, Nan & Li, Guoliang. (2018). DeepEye: Towards Automatic Data Visualization. 101-112. 10.1109/ICDE.2018.00019.

15. Mackinlay, J., Hanrahan, P., Stolte, C.: Show me: Automatic presentation for visual analysis. In: IEEE Transactions on Visualization and Computer Graphics 13.6 (2007)

16. Кривко А.О. Візуалізація як вибір дизайну // Інформаційні моделі, системи та технології: Праці XII наук.-техн. конф. (Тернопіль, ТНТУ ім. І. Пулюя, 18-19 грудня 2024 р.) С. 36.

17. Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, pp. 321—357, 2002.

18. Plotly Chart Studio. URL: https://chart-studio.plotly.com/create/ (дата звертання: 16.11.2024).

19. Plotly REST API, v2. URL: https://api.plot.ly/v2/ (дата звертання: 16.11.2024).

20. Plotly. Plotly Community Feed. URL: https://plot.ly/feed (дата звертання: 16.11.2024).

21. Qian, Xin & Rossi, Ryan & Du, Fan & Kim, Sungchul & Koh, Eunyee & Malik, Sana & Lee, Tak Yeon & Chan, Joel. (2021). Learning to Recommend Visualizations from Data. pp. 1359-1369. 10.1145/3447548.3467224.

22. Victor, Dibia & Demiralp, Çağatay. (2018). Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks. IEEE Computer Graphics and Applications. PP. 10.1109/MCG.2019.2924636.

23. Viegas, F., Wattenberg, M., van Ham, F., Kriss, J., McKeon, M.: ManyEyes: a site for visualization at internet scale. In: IEEE Transactions on Visualization and Computer Graphics 13.6 (2007)

24. VizML: Training Data, Feature Extraction, and Model Training. URL: https://github.com/mitmedialab/vizml (дата звертання: 26.11.2024).

25. Zhou, Mengyu & Li, Qingtao & He, Xinyi & Li, Yuejiang & Liu, Yibo & Ji, Wei & Han, Shi & Chen, Yining & Jiang, Daxin & Zhang, Dongmei. (2021). Table2Charts: Recommending Charts by Learning Shared Table Representations. pp. 2389-2399. 10.1145/3447548.3467279.

26. . Petryk, M. , Bachynskyi, M. , Brevus, V. , Mudryk, I. , Mykhalyk, D. Analysis technology of neurological movements considering cognitive feedback influences of cerebral cortex signals CEUR Workshop Proceedings, 2022, 3309, pp. 45–54.

27. Mykhalyk, D. , Petryk, M. , Maria Petryk, K. , Petryk, O. , Mudryk, I Mathematical Modeling of Hydrocarbons Adsorption in Nanoporous Catalyst Media using Nonlinear Langmuir's Isotherm using Activation Energy. Proceedings - International Conference on Advanced Computer Information Technologies, ACIT, 2019, pp. 72–75.

28. Boyko, I. , Petryk, M. , Mudryk, I. , Stoianov, Y., Tsupryk, H. Mathematical Model of the Capacitor Based on Zeolite Material. Proceedings - International Conference on Advanced Computer Information Technologies, ACIT, 2021, pp. 45–48.

29. M. R. Petryk, I. V. Boyko, O. M. Khimich, and M. M. Petryk, “High-performance supercomputer technologies of simulation of nanoporous feedback systems for adsorption gas purification,” Cybern. Syst. Analysis, Vol. 56, No. 5, 835–847 (2020). https://doi.org/https://doi.org/10.1007/s10559-020-00304-y

30. Petryk, M. , Doroshenko, A. , Mykhalyk, D. , Ivanenko, P. , Yatsenko, O. Automated Parallelization of Software for Identifying Parameters of Intraparticle Diffusion and Adsorption in Heterogeneous Nanoporous Media Lecture Notes in Networks and Systems, 2023, 667 LNNS

31. . Stefanyshyn, V. , Stefanyshyn, I. , Pastukh, O. , Yatsyshyn, V. , Yakymenko.

Accuracy of software and hardware of computer systems for human-machine interaction, I. CEUR Workshop Proceedings, 2024, 3842, pp. 178–183.

32. Зеркалов Д.В. Охорона праці в галузі: Загальні вимоги. Навчальний посібник. К.: Основа. 2011. 551 с

33. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

ДОДАТКИ

ДОДАТОК А

Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА

М А Т Е Р І А Л И

XII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

18-19 грудня 2024 року

ТЕРНОПІЛЬ
2024

УДК 004.92
А. О. Кривко
Тернопільський національний технічний університет імені Івана Пулюя, Україна

ВІЗУАЛІЗАЦІЯ ЯК ВИБІР ДИЗАЙНУ

A. Kryvko

VISUALIZATION AS A DESIGN CHOICE

Сформулювати базову візуалізацію набору даних d можна як набір взаємопов'язаних варіантів дизайну C = {c}, кожен з яких вибирається з можливого простору c ~ ℂ. Однак не всі варіанти дизайну призводять до правильних візуалізацій - деякі варіанти несумісні один з одним. Наприклад, кодування категоріального стовпця з віссю Y лінійної графіки неприпустимо. Отже, множина варіантів, що призводять до правильних візуалізацій, є підмножиною простору всіх можливих варіантів $\mathbb{C}1 \times \mathbb{C}2 \times \ldots \times \mathbb{C}_{|C|}$.

Ефективність візуалізації може бути визначена інформаційними показниками, такими як ефективність, точність і запам'ятовуваність, або емоційними показниками, такими, як залучення. Дослідження також показують, що ефективність визначається низькорівневими принципами сприйняття та властивостями набору даних, на додаток до контекстуальних факторів, таких як завдання, естетика, домен, аудиторія та середовище. Інакше кажучи, аналітик вибирає дизайн для візуалізації C_{max}, який максимізує ефективність візуалізації Eff з урахуванням набору даних d і контекстуальних факторів T:

$$C_{max} = arg\max_{C} \mathrm{Eff}(C|d,T)$$

Але вибір дизайну може бути коштовним. Мета рекомендацій візуалізації - знизити вартість створення візуалізацій за рахунок автоматичної пропозиції підмножини варіантів дизайну $Crec \subseteq C$ (рис. 1).

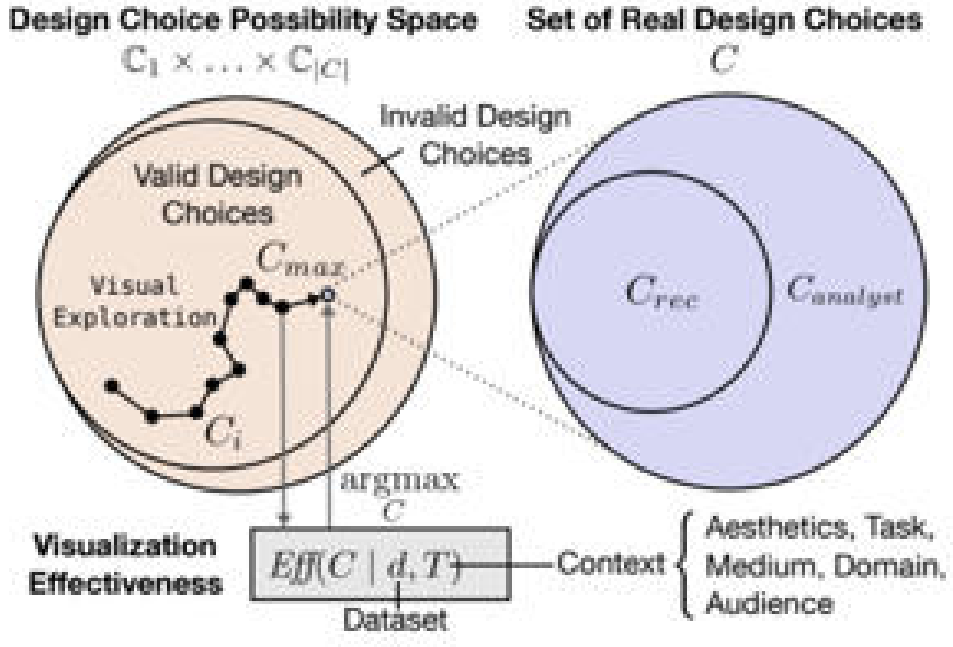

Рисунок 1 – Створення візуалізацій – це процес вибору дизайну, який може бути рекомендований системою або вказаний аналітиком

Додаток Б

Диск з роботою