НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.932.4

До захисту допущено
В. о. завідувача кафедри ММСА
О.Л.Тимощук
«___» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз
на тему: «Моделі і методи прогнозування молекулярних механізмів дії фармацевтичних препаратів»

Виконав:
студент ІІ курсу, групи КА-92мп
Сенюк Костянтин Костянтинович _____

Керівник:
доцент кафедри ММСА,
к. т. н., доцент Жиров О.Л. _____

Рецензент:
доцент кафедри системного проектування
КПІ ім. Ігоря Сікорського,
к. т. н., доцент Кисельов Г.Д. _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань

Студент _____

Київ
2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)
Спеціальність — 124 «Системний аналіз»

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри ММСА
О. Л. Тимощук
«___» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студента Сенюка Костянтина Костянтиновича

1. **Тема дисертації**: «Моделі і методи прогнозування молекулярних механізмів дії фармацевтичних препаратів», науковий керівник дисертації Жиров Олександр Леонідович, к. т. н., доцент, затверджені наказом по університету від «02» листопада № 3182-с
2. **Термін подання студентом дисертації**: 15 грудня 2020 р.
3. **Об'єкт дослідження**: статистичні дані по експресії генів і життєздатності клітин.
4. **Предмет дослідження**: математичні методи та моделі класифікації, оцінювання та аналізу якості класифікацій.
5. **Перелік завдань, які потрібно розробити**:
   1) розглянути методи класифікації та кластеризації;
   2) розробити математичну модель для прогнозування механізмів дії лікарських речовин;
   3) виконати обчислювальні експерименти стосовно моделювання та прогнозування фінансового ринку з використанням GANs;
   4) розробити стартап-проект виведення на ринок результатів дослідження.
6. **Орієнтовний перелік графічного (ілюстративного) матеріалу**:
   1) Графічний аналіз даних(рис.);
   2) Схеми методів аналізу даних та нейронної мережі (рис.);
   3) Таблиці у розділі стартап-проекту
7. **Дата видачі завдання**: 05 вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1.	Концептуальний вступ дисертації. Формулювання об’єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	05.09.2020—12.09.2020
2.	Перший розділ. Огляд літературно-інформаційних джерел. Понятійно-категоріальний апарат. Характеристика об’єкта	16.09.2020—27.09.2020
3.	Другий розділ. Нейронна мережа ж критерієм оптимізації Adam.	30.09.2020—19.10.2020
4.	Третій розділ. Пошук даних та реалізація алгоритму. Імплементація отриманих результатів у програмний продукт. Тестування програми	22.10.2020—15.11.2020
5.	Четвертий розділ. Стартап-проект	18.11.2020—20.11.2020
6.	Концептуальні висновки. Перспективи розвитку отриманих рішень	21.11.2020—25.11.2020

Студент Сенюк К. К.

Науковий керівник дисертації Жиров О.Л.

РЕФЕРАТ

Магістерська дисертація містить 111 сторінку, 41 рисунок, 25 таблиць. А також було використано 17 джерел.

ПРОГНОЗУВАННЯ МЕХАНІЗМІВ ДІЇ ЛІКАРСЬКИХ РЕЧОВИН, КЛАСИФІКАЦІЯ, ДЕРЕВА РІШЕНЬ, АНСАМБЛЬ МОДЕЛЕЙ, НЕЙРОННА МЕРЕЖА, МЕТОД ЗВОРОТНОГО ПОШИРЕННЯ ПОМИЛКИ, АЛГОРИМ ОПТИМІЗАЦІЇ АДАМА.

Метою дослідження є аналіз способів очистки даних і порівняльний аналіз моделей для прогнозування механізмів дії лікарських речовин, і визначення найкращої серед них.

Для цього завдання було використано методи кластеризації даних (DBSCAN, HDBSCAN, One-Class SVM) для пошуку викидів в даних, метод головних компонент для зменшення корельованості вхідних процесів, нейронна мережа, ансамбль моделей (бустінг).

Об’єктом дослідження були статистичні дані по експресії генів і життєздатності клітин.

Було розроблено програму на мові Python для прогнозування вхідних статистичних даних і обчислення оцінок для порівняння використаних моделей і методів.

В результаті було вибрано найкращий метод який підходить для даної роботи, і з допомогою його було створено модель для класифікації різних механізмів дії лікарських речовин.

ABSTRACT

Master`s thesis contains 111 pages, 41 drawings, 25 tables. And also 17 sources were used.

MECHANISM OF ACTION, CLASSIFICATION, DECISION TERMS, ENSEMBLE LEARNING, NEURAL NETWORKS, BACKPROPAGATION, ADAM ALGORYM OPTIMIZATION.

The aim of the study is to analyze the methods of data purification and comparative analysis of models to predict the mechanisms of action of drugs and determine the best one among them.

For this task, data clustering methods (DBSCAN, HDBSCAN, One-Class SVM) were used to search for emissions in the data, the main component method to reduce the correlation of input processes, the neural network, an ensemble of models (boosting).

The object of the study was statistical data on gene expression and cell viability.

The Python software was developed to predict the input statistics and calculate estimates to compare the used models and methods.

As a result, the best method suitable for this work was selected and a model was created to classify the various mechanisms of drug action.

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

MoA (mechanism of action) – механізм дії лікарських речовин

SVM (support vector machine) – метод опорних векторів

CART (classification and regression trees) – дерева рішень

NN (neural network) – нейронна мережа

PCA (principal component analysis) – метод головних компонент

ВСТУП

У минулому вчені отримували ліки з натуральних продуктів або надихалися традиційними засобами. Дуже поширені лікарські засоби, такі як парацетамол, відомий в США як ацетамінофен, було введено в клінічне застосування за десятиліття до того, як були зрозумілі біологічні механізми, що лежать в основі їх фармакологічної діяльності. Сьогодні, з появою більш потужних технологій, відкриття лікарських засобів перетворилося з інтуїтивного підходу минулого в більш цілеспрямовану модель, засновану на розумінні біологічного механізму, що лежить в основі захворювання.

У цих нових рамках вчені прагнуть визначити білкову мішень, пов'язану з хворобою, і розробити молекулу, здатну модулювати цю білкову мішень. В якості короткого опису біологічної активності даної молекули вчені привласнюють ярлик, званий механізмом дії або MoA.

Один з підходів для визначення MoA нового препарату полягає в тому, щоб обробити зразки клітин людини ліками, а потім проаналізувати клітинні реакції за допомогою алгоритмів, які шукають схожість з відомими шаблонами в великих генних базах даних, таких як бібліотеки зразків експресії генів або зразків клітинної життєздатності ліків з відомими MoA.

В даній роботі ми маємо унікальний набір даних, що об'єднує дані по експресії генів і життєздатності клітин. Ця інформація базується на новій технології, яка вимірює одночасно (в межах одних і тих же зразків) реакції клітин людини на ліки в пулі з 100 різних типів клітин (таким чином, вирішується проблема ідентифікації до події, які типи клітин краще підходять для даного препарату).

Для аналізу набору даних ми використовуємо такі підходи, як DBSCAN, HDBSCAN, One-Class SVM (для пошуку викидів), PCA(для зменшення розмірності вхідних даних та корельованості між процесами). Для навчання моделі використано логістична класифікація, CART, XGBOOST та NN з

використанням критерія оптимізації Adam. Для перевірки моделі використовується середнє значення логарифмічної функції втрат, застосовуваної до кожної пари приміток МоА-препарат.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ МЕТОДІВ АНАЛІЗУ ДАНИХ

1.1 Вступ

В даному розділі буде розглянуто основна теорія по методам пошуку викидів в даних, а також способами зменшити корельованість між вхідними процесами для уникнення проблем в подальшому навчанні моделі. Після чого буде описано деякі недоліки та поради для розібраних методів.

1.2 Пошук аномалій в тренувальному наборі даних

У статистиці відхилення - це точка даних, яка значно відрізняється від інших точок даних у вибірці. Часто відхилення в наборі даних можуть попереджати статистиків про експериментальні відхиленнях або помилки в проведених вимірах, що може призводити до того, що вони будуть опускати відхилення з набору даних (Рис. 1.1). Якщо вони будуть опускати виключення зі свого набору даних, то це може привести до значних змін у висновках, зроблених за результатами дослідження. [1]

Рисунок 1.1 – приклад відхилення в даних

Що може викликати відхилення?

- Помилки введення даних: Людські помилки, такі як помилки, викликані під час збору, запису або введення даних, можуть привести до викидів в даних. Наприклад: Річний дохід клієнта складає 50000$. Випадково оператор введення даних ставить додатковий нуль на малюнку. Тепер дохід стає 500000$, що в 10 разів більше. Очевидно, що це буде величина викиду в порівнянні з рештою населення.
- Помилка вимірювання: Це найбільш поширене джерело відхилень. Це відбувається, коли використовується вимірювальний прилад виявляється несправним. Наприклад: Є 10 ваговимірювальних машин. 9 з них правильні, 1 несправна. Вага, виміряна людьми на несправній машині, буде більше / менше, ніж у інших людей в групі. Вага, виміряна на несправній машині, може привести до викидів.
- Експериментальна помилка
- Навмисне відхилення: Зазвичай це трапляється в самозвітах, які включають в себе конфіденційні дані. Наприклад, підлітки повідомляють про кількість алкоголю, яку вони вживають. Тільки частина з них дає достовірну інформацію. Тут фактичні значення можуть виглядати як відхилення, тому що інші підлітки не повідомляють про споживання.
- Помилка при обробці даних: Всякий раз, коли ми здійснюємо видобуток даних, ми здобуваємо дані з декількох джерел. Можливо, що деякі помилки маніпуляції або вилучення можуть привести до викидів в наборі даних.
- Помилка вибірки: Наприклад, ми повинні виміряти зріст спортсменів. Помилково ми включаємо в вибірку кілька баскетболістів. Це включення, швидше за все, призведе до відхилень в наборі даних.
- Природне відхилення: Коли викид не є штучним (через помилки), це природний викид.

1.2.1 Класичні способи пошуку відхилень

Існують різні емпіричні правила для виявлення відхилень.

- Будь-яке значення, що виходить за межі діапазону від -1,5 × IQR до 1,5 × IQR (IQR: Interquartile Range).
- Будь-яке значення, що виходить за межі діапазону 5-го і 95-го процентилів, може розглядатися як відхилення.
- Точки даних, що лежать на відстані трьох або більше стандартних відхилень від середнього вважаються викидом. [1]

Найпростіший спосіб швидкого виявлення відхилення – візуалізація.

Histograms

Цей вид візуалізації показує частоту груп даних. Він може нам показати де знаходяться найбільш часто повторювані значення набору даних. Також ми можем побачити форму нашого розподілу і перевірити, чи не «перекошен» він в будь-який бік (Рис. 1.2).

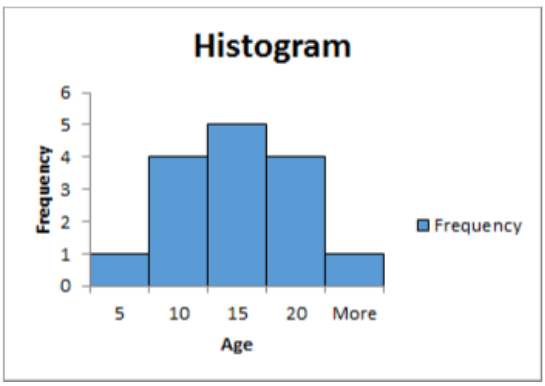


Рисунок 1.2 – приклад гістограми даних

Boxplots

В описовій статистиці, *boxplot* - це метод графічного відображення груп числових даних через їх квантилі. Графіки боксів можуть також мати лінії, що тягнуться вертикально від боксів (вусів), які вказують на мінливість поза верхнього і нижнього квантилів, звідси і терміни "графік боксів і вусів". Надлишки можуть бути нанесені у вигляді окремих точок.

Вищенаведене визначення передбачає, що якщо є відхилення, то воно буде записано як точка в *boxplot*, а інша популяція буде згрупована і відображена як осередки. Давайте спробуємо подивитися самі (Рис. 1.3):

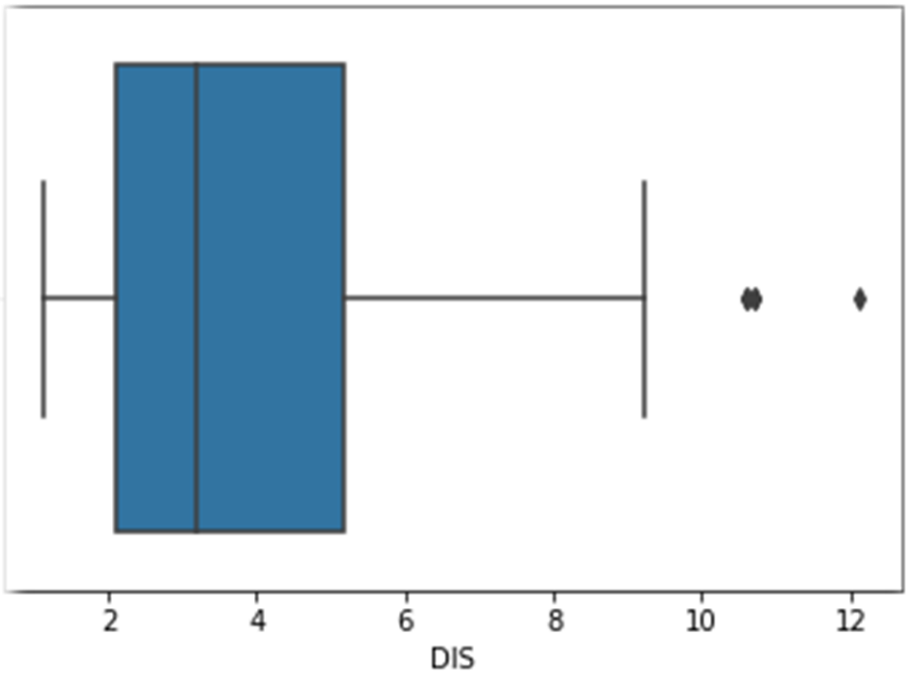


Рисунок 1.3 – приклад *boxplot* даних

На малюнку вище показані три точки в діапазоні від 10 до 12, вони є викидами, так як не включені в вікно іншого спостереження, тобто не включені в вікно поруч з квантилями.

Scatter Plot

Це вид графіка або математичної діаграми, який використовує декартові координати для відображення значень набору даних. Дані відображаються як набір точок, кожна з яких має значення однієї змінної, що визначає положення на горизонтальній осі, і значення іншої змінної, що визначає положення на вертикальній осі. [2]

Як випливає з визначення, *scatter plot* являє собою набір точок, що показують значення для двох змінних Розглянемо можливу ситуацію (Рис. 1.4):

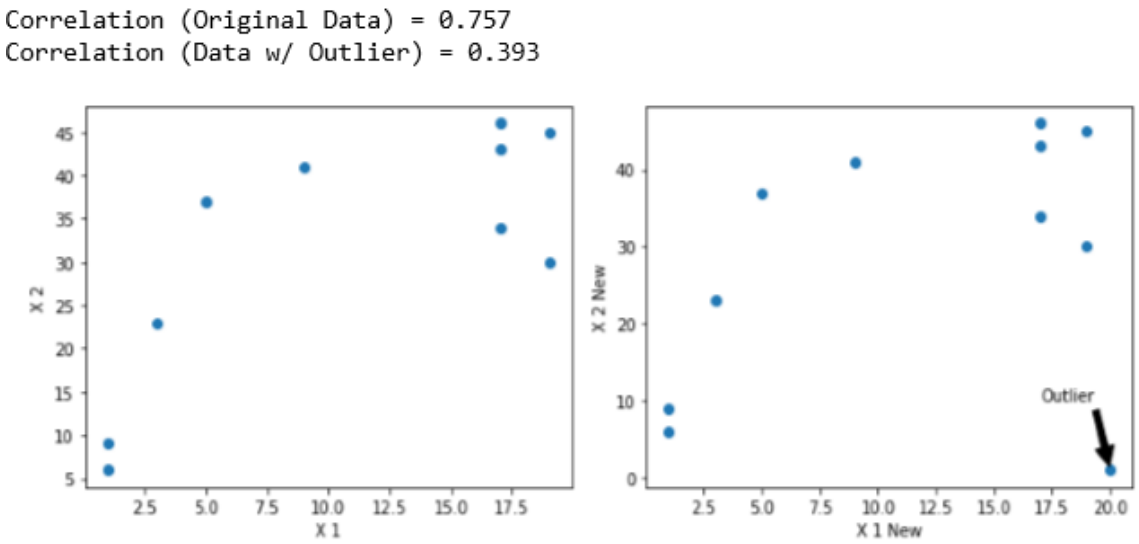


Рисунок 1.4 – приклад викиду на *scatter plot*

1.2.2 Типи викидів

- Одновимірна викид: одновимірний викид - це точка даних, яка складається з екстремального значення однієї змінної.

- Багатомірний викид: Багатомірний викид - це комбінація незвичайних значень як мінімум по двох змінним. [2]

Наприклад, припустимо, ми розуміємо зв'язок між зростом і вагою. Нижче ми маємо одновимірний і двовимірний розподіл для наших даних. Дивлячись на графік *boxplot* (Рис. 1.5), у нас немає ніяких відхилень (вище і нижче 1.5 * IQR, найбільш поширений метод). Тепер подивимось на *scatter plot*. Тут у нас є два значення нижче і одне вище середнього на певному відрізку ваги і висоти.

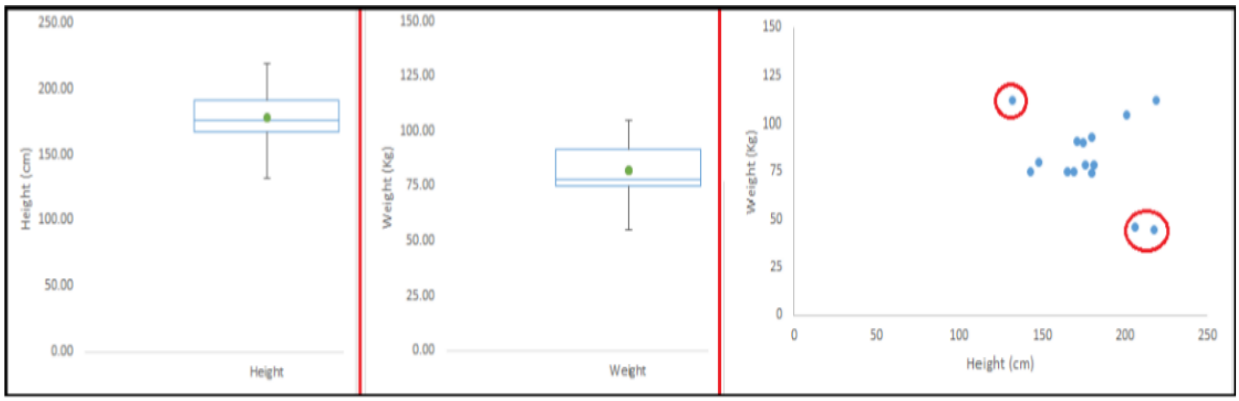


Рисунок 1.5 – приклад багатовимірного викиду

1.2.3 Класичні способі зменшення викидів в даних

- Видалення обсервацій: Ми видаляємо значення викиду, якщо це пов'язано з помилкою при введенні даних, помилкою при обробці даних або спостереженнями за викидами, які дуже малі за кількістю. Ми також можемо використовувати обрізку на обох кінцях для видалення відхилень.
- Перетворення значень: Перетворення змінних також може усунути відхилення. Натуральний логарифм значень зменшує відхилення, викликані екстремальними значеннями (Рис. 1.6).

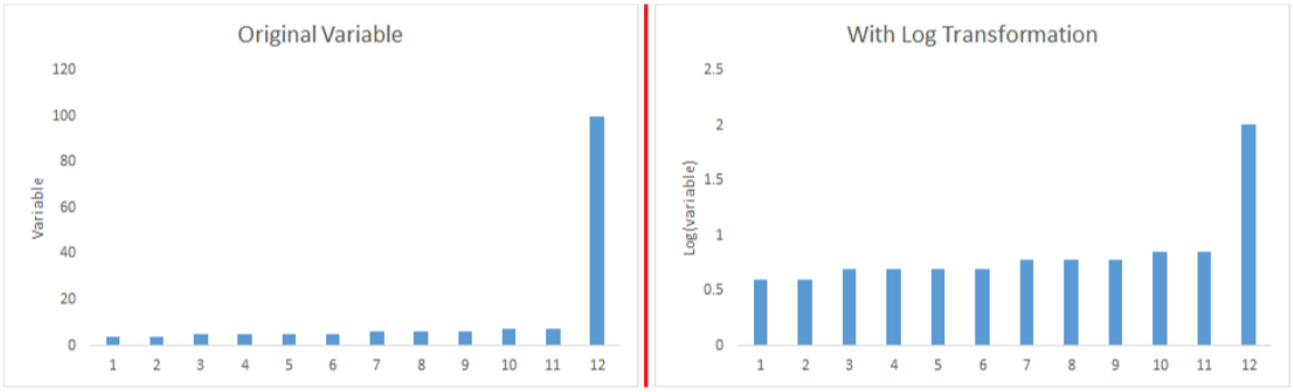


Рисунок 1.6 – приклад перетворення значень вибірки

- Розділення даних: Якщо є значне число відхилень, ми повинні розглядати їх окремо в статистичної моделі. Один з підходів полягає в тому, щоб розглядати обидві групи як дві різні групи і будувати індивідуальну модель для обох груп, а потім об'єднувати результати.

1.3 Кореляційний аналіз в тренувальному наборі даних

Кореляція даних і ознак вважається одним з важливих етапів на етапі вибору ознак в процесі попередньої обробки даних, особливо якщо тип даних для ознак є безперервним. так що ж таке кореляція даних.

Кореляція даних - це спосіб розуміння взаємозв'язку між множинними змінними і атрибутами в вашому наборі даних. [3]

Використовуючи кореляцію, ви можете отримати деяке уявлення щодо даних, наприклад:

- Один або кілька атрибутів залежать від іншого атрибута або причини для іншого атрибута.
- Один або кілька атрибутів асоціюються з іншими атрибутами.
- Кореляція може допомогти в передбаченні одного атрибута від іншого.

- Кореляція може (іноді) вказувати на наявність причинно-наслідкового зв'язку.

Кореляція використовується в якості основної величини для багатьох методів моделювання. І існує три типи кореляцій:

- Позитивна кореляція: означає, що якщо об'єкт А збільшується, то і об'єкт В збільшується, або якщо об'єкт А зменшується, то і об'єкт В зменшується. Обидва об'єкти рухаються в тандемі і мають лінійний зв'язок (Рис. 1.7).

- Негативна кореляція: означає, що якщо функція А збільшується, то функція В зменшується, і навпаки.

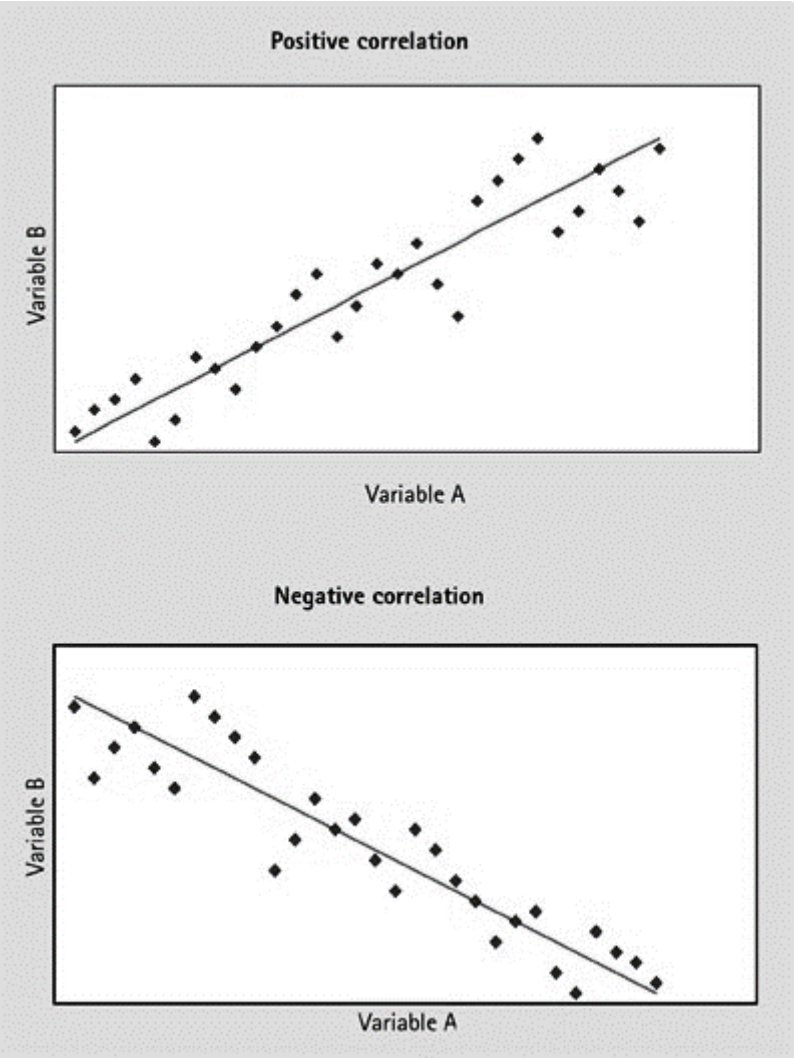


Рисунок 1.7 – приклад корельованості значень вибірки

- Кореляція відсутня: немає зв'язку між цими двома атрибутами або він незначний.

Кожен з цих типів кореляції може існувати в спектрі, представленому значеннями від 0 до 1, де злегка або сильно позитивні кореляційні ознаки можуть бути чимось на зразок 0.5 або 0.7. Існування сильної і ідеальна позитивна кореляція, це може бути результат представлений значенням коефіцієнта кореляції 0.9 або 1.

Якщо набір даних має абсолютно позитивні або негативні атрибути, то існує висока ймовірність того, що на продуктивність моделі впливатиме проблема мультиколінеарності. Мультиколінеарність виникає тоді, коли одна змінна в багаторазової регресійній моделі може бути лінійно передбачена з інших з високим ступенем точності. Це може привести до вводу в оману результатами. На щастя, дерева рішень за своєю природою несприйнятливі до мультиколінеарності. Дерево вибере тільки одну з ідеально корельованих характеристик. Однак, інші алгоритми, такі як логістична регресія або лінійна регресія, не застраховані від цієї проблеми, і ми повинні вирішити дану проблему перед тренуванням моделі.

Є кілька способів вирішення цієї проблеми. Найпростіший спосіб - це видалити або усунути одну з сильно корельованих функцій. Інший спосіб - використовувати алгоритм зменшення розмірів, такий як метод головних компонент (PCA).

1.3.1 Метод головних компонент (PCA)

PCA - це статистична процедура, що використовує ортогональне перетворення для перетворення набору спостережень з корельованими змінними (об'єкти, кожен з яких приймає різні числові значення) в набір

значень лінійно некорельованих змінних, які називаються основними компонентами. [4]

Це перетворення визначається таким чином, що перша основна компонента має найбільшу можливу дисперсію (тобто становить максимально можливу варіабельність даних), а кожна наступна компонента, в свою чергу, має найбільшу можливу дисперсію при обмеженні, що вона ортогональна по відношенню до попередніх компонентів . Отримані вектори (кожен з яких представляє собою лінійну комбінацію змінних і містить n спостережень) є некореляційним ортогональним базисним набором. PCA чутливий до відносного масштабування вихідних змінних. [5]

PCA в основному використовується як інструмент для аналізу даних і для створення моделей прогнозування. Він часто використовується для візуалізації генетичної дистанції і спорідненості між популяціями. PCA може бути здійснено шляхом декомпозиції власних значень матриці коваріації (або кореляції) даних або декомпозиції сингулярних значень матриці даних, як правило, після етапу нормалізації вихідних даних (Рис. 1.8).

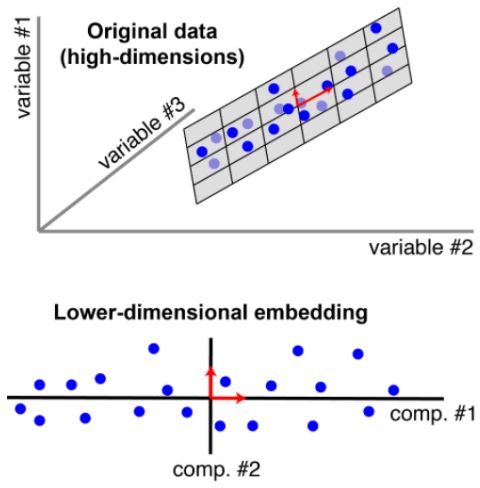


Рисунок 1.8 – зменшення розмірності даних з допомогою PCA

PCA базується на системі коефіцієнтів кореляції Пірсона і успадковує аналогічні припущення.

Розглянемо основні поради для використання даного метода:

- Розмір вибірки: мінімум 150 спостережень, а в ідеалі співвідношення спостережень до характеристик 5: 1. [6]
- Кореляція: набір функцій корельований, тому скорочений набір функцій ефективно представляє собою початковий простір даних.
- Лінійність: всі змінні мають постійну багатовимірну нормальну залежність, а основні компоненти являють собою лінійну комбінацію вихідних ознак.
- Відхилення: Ніяких значних відхилень в даних, так як вони можуть непропорційно сильно впливати на результати.
- Велика дисперсія має на увазі велику структуру: осі високої дисперсії розглядаються як основні компоненти, в той час як осі низькою дисперсії розглядаються як шумові і відкидаються. [6]

Ми опустимо всі математичні похідні для PCA і просто покажемо процедуру, як їх знайти.

Спочатку необхідно обчислити центр даних за середнім значенням і нормалізувати їх:

$$x_i = \frac{x_i - M(x_i)}{std(x_i)} \tag{1.1}$$

де, $i = 1 \ldots m$ переглядає кожну функцію.

По-друге, нам потрібна коваріаційна матриця Σ для набору даних (що є симетричною):

$$\sum = cov(X) = \frac{1}{n-1} X^T X \tag{1.2}$$

По-третє, слід обчислити значення і власні вектори для Σ (знайти правильну ротацію) або просто виконати сингулярне розкладання значень (SVD) (воно більш чисельно стабільно, ніж обчислення власних значень):

$$SVD(\sum) = U * S * V^T$$
$$U^T = U^{-1} \quad (1.3)$$
$$U = V^T$$

де, S - діагональ, що містить власні значення. U - містить власні вектори нового базису.

Тепер можна виконати перетворення PCA за допомогою матриць U, вибравши k-кількість компонентів:

$$Z = X * U[:, :k] = X * V[:k, :]^T \quad (1.4)$$

Зворотне перетворення виконується аналогічним чином за допомогою U-матриці (Рис. 1.9):

$$x_{reconstruction} = Z * U[:, :k]^T = Z * V[:k, :] \quad (1.5)$$

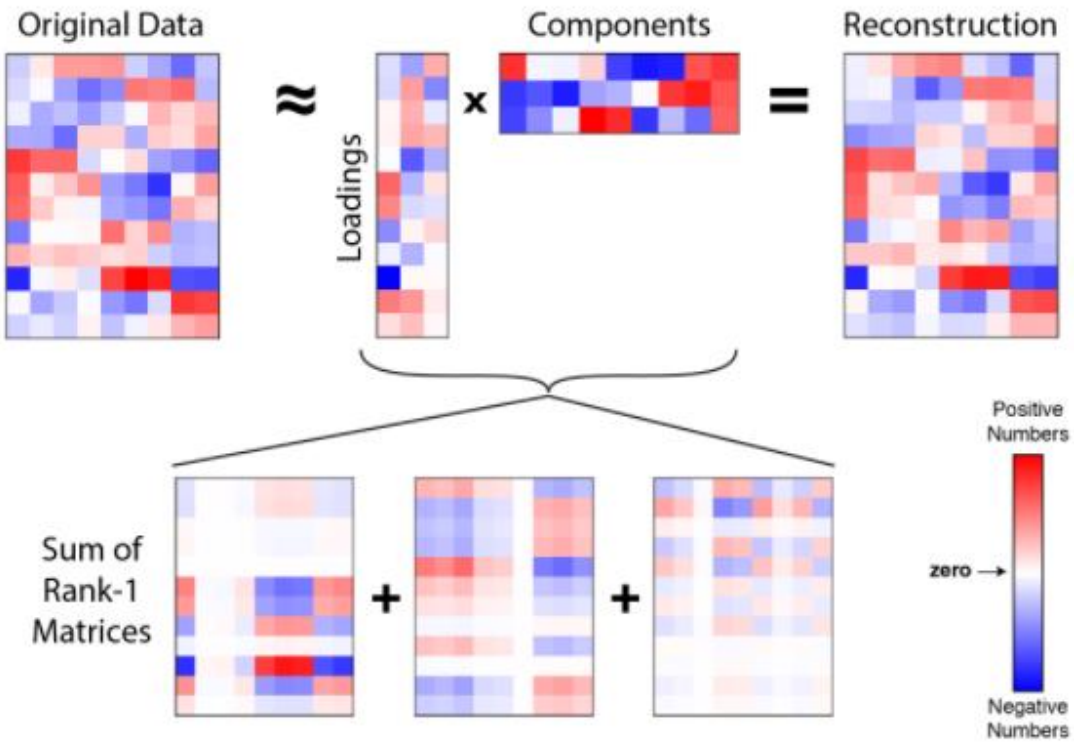


Рисунок 1.9 – візуалізація процедури зменшення розмірності даних

Максимізація дисперсії в просторі основного компонента еквівалентна мінімізації похибки реконструкції методом найменших квадратів (Рис. 1.10).

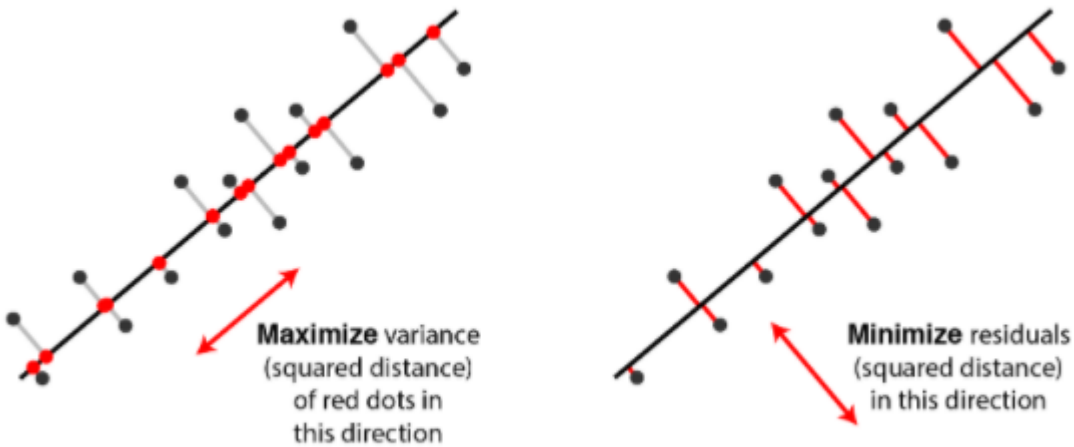


Рисунок 1.10 – мінімізація похибки

Розглянемо точку даних, рядок матриці даних (Рис. 1.11). Припускаючи, що дані є середньоцентрованими, проекція на основні компоненти пов'язує дисперсію, що залишилася, з квадратом залишку за теоремою Піфагора. Вибір компонентів, щоб максимально збільшити дисперсію, такий самий, як вибір їх для мінімізації квадратних залишків.

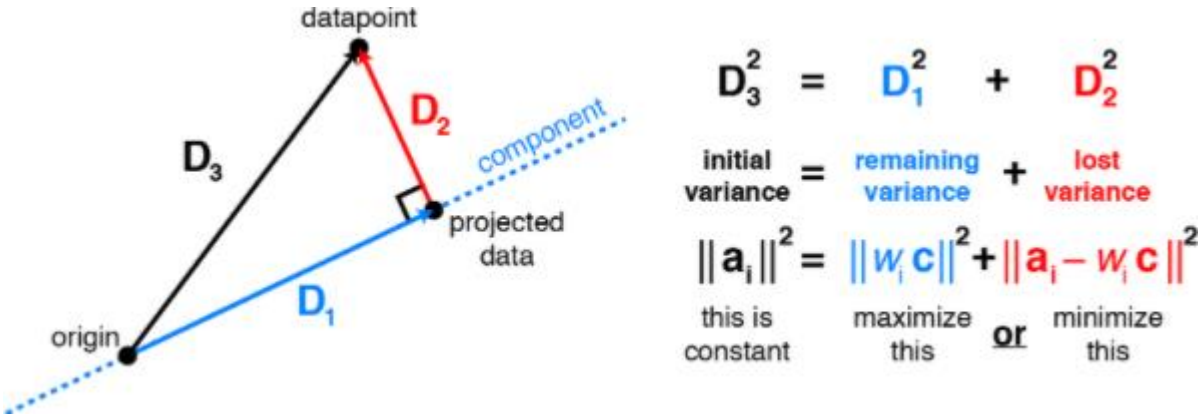


Рисунок 1.11 – мінімізація похибки

Думати про PCA як про мінімізацію похибки реконструкції корисно, тому що він будує зв'язок зі статистичної регресії. Проста лінійна регресія з методом найменших квадратів була розширена і адаптована до широкого спектру статистичних проблем, і ми можемо використовувати цю дослідницьку базу і перспективу для створення більш спеціалізованих версій PCA.

1.4 Нормалізація та обробка даних для подальшого тренування моделі

Попередня обробка даних - це метод видобутку даних, який передбачає перетворення вихідних даних в зрозумілий формат. [7]

Реальні дані часто є неповними, непослідовними і / або не мають певної поведінки або тенденцій, і, ймовірно, містять багато помилок. Попередня обробка даних є перевіреним методом вирішення таких проблем.

Попередня обробка даних (Рис. 1.12) включає: нормалізацію, стандартизацію, кодування категоріальних ознак і біннінг.

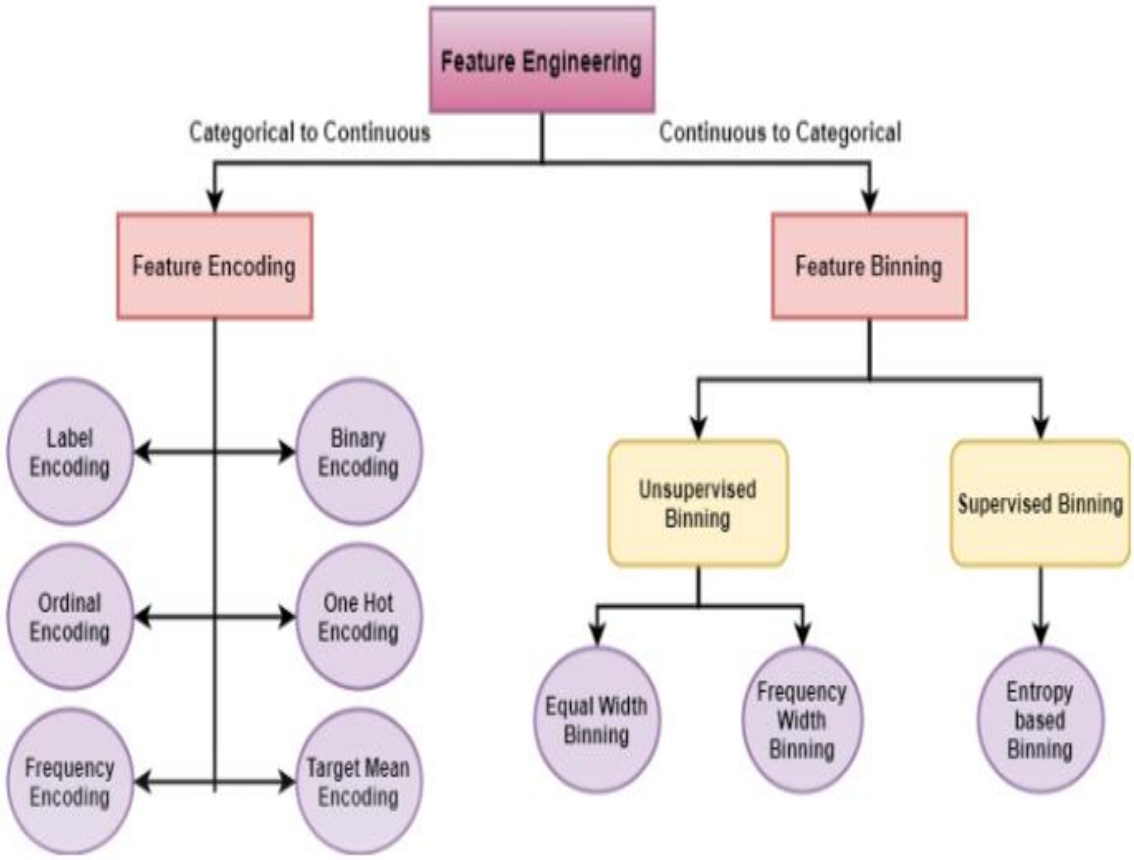


Рисунок 1.12 – попередня обробка даних

1.4.1 Нормалізація даних

Нормалізація - це зміна значень числових стовпців в наборі даних на загальний масштаб, без спотворення відмінностей в діапазонах значень. [8]

Вона потрібно тільки в тому випадку, якщо характеристики мають різні діапазони значень. Не всі алгоритми вимагають нормалізації. Це процес масштабування окремих вибірок до одиничної норми. Існує 3 основні методи нормалізації: максимальна норма, l1-норма і l2-норма.

Максимальна норма:

$$X_i norm = \frac{X_i}{max(X)} \tag{1.6}$$

Отже, що ми робимо тут, так це ділимо кожне значення з розподілу X на максимальне значення цього розподілу.

L1-норма:

$$X_i norm = \frac{X_i}{\sum_{i=1}^{m} X_i} \tag{1.7}$$

І тут ми ділимо на суму X.

L2-норма:

$$X_i norm = \frac{X_i}{\sqrt{\sum_{i=1}^{m} X_i^2}} \tag{1.8}$$

Цей тип також називають евклідовою нормою.

1.4.2 Стандартизація даних

Стандартизація наборів даних є загальною вимогою для багатьох машинобудівних навчальних моделей, для того щоб розмістити всі значення функцій в одному діапазоні.

Мотивація до використання такого масштабного введення включає в себе стійкість до дуже малого стандартного складання характеристик та збереження нульових записів у розрізнених даних.

Змінна з розрізненими даними - це та змінна, у якій відносно високий відсоток значень змінної не містить реальних даних. [9]

Альтернативна стандартизація - це масштабовані змінні, які лежать між заданим мінімальним і максимальним значенням, часто між нулем та одиницею або максимального абсолютного значення кожної функції, масштабованої до розміру одиниць. Розглянемо ж, як можна масштабувати дані для їх стандартизації.

Масштабування з використанням мінімальних і максимальних значень:

$$X_i scaled = \frac{X_i - min(X)}{max(X) - min(X)} \tag{1.9}$$

Це масштабує кожну змінну окремо таким чином, що вона знаходиться в заданому діапазоні навчального набору, наприклад від нуля до одиниці.

Масштабування змінних за максимальною абсолютною величиною:

$$X_i scaled = \frac{X_i}{max(|X|)} \tag{1.10}$$

Це масштабує кожну змінну окремо таким чином, що максимальне абсолютне значення кожної ознаки у навчальному наборі буде 1. Це не зміщує / центрує дані, а отже, не знищує розрідженість.

Стандартизація змінної, вилучивши середнє значення та масштабуючи до середньоквадратичного відхилення, обчислюється як:

$$X_i scaled = \frac{X_i - mean(X)}{std(X)} \tag{1.11}$$

Порівняймо вплив різних «скалерів» для масштабування на даних (Рис. 1.13-1.15) з викидами:

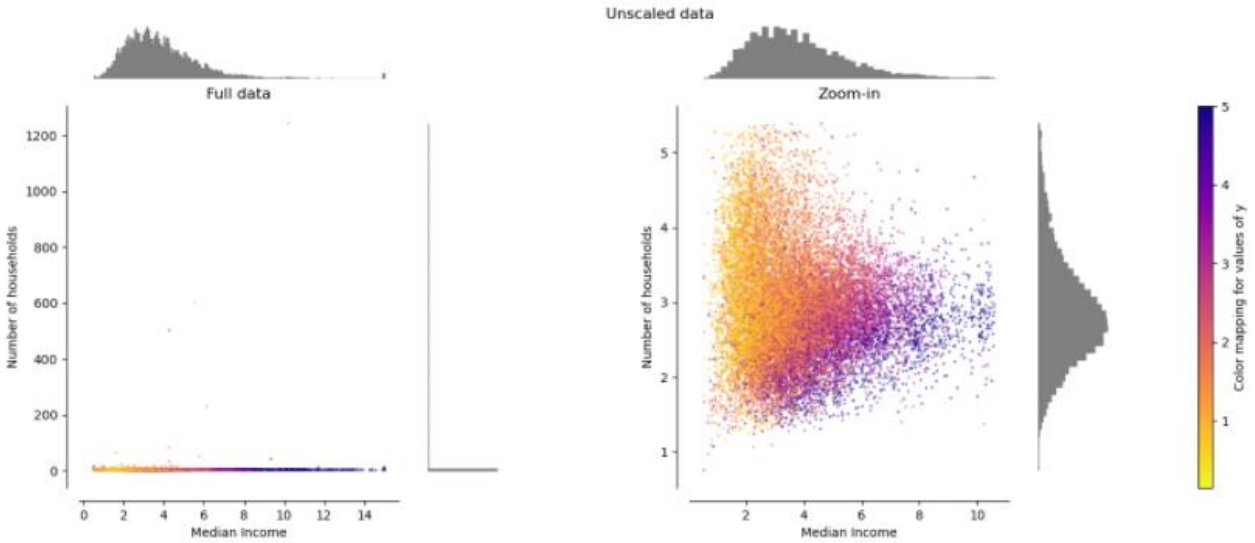


Рисунок 1.13 – дані без обробки

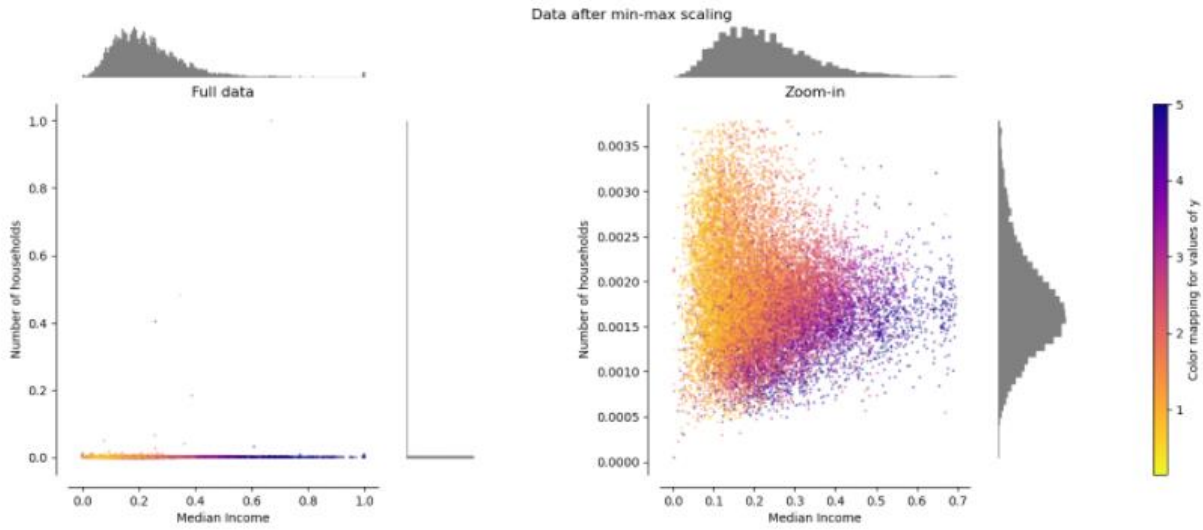


Рисунок 1.14 – дані з масштабуванням «мін-макс»

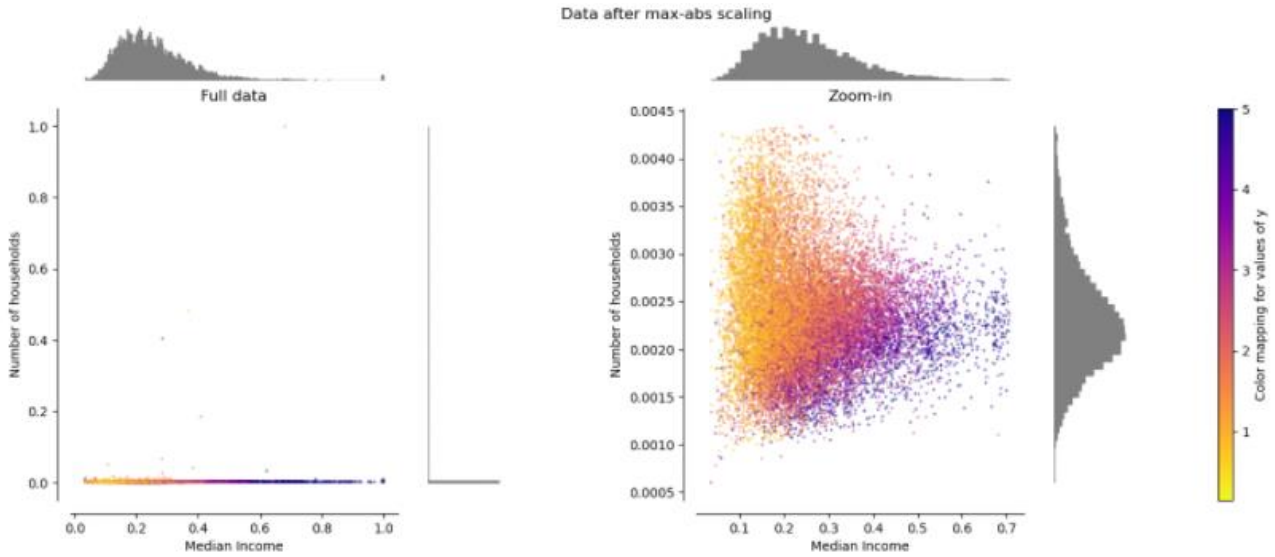


Рисунок 1.15 – дані з масштабуванням за максимальною величиною

1.4.3 Кодування категоріальних змінних

Категоріальні дані представляють дискретні значення, які належать до певного кінцевого набору категорій або класів.

Ці дискретні значення можуть бути текстовими, числовими або навіть зображеннями в природі. Існує два основних класи категоріальних даних, *номінальні* та *порядкові*. [10]

У будь-якому номінальному категоріальному атрибуті даних немає поняття впорядкування серед значень цього атрибута. Приклади номінальної змінної:

- червоний, зелений, синій;
- кішка, собака, змія;
- Київ, Лондон, Варшава.

Порядкові категоріальні атрибути мають певний сенс або поняття порядку серед своїх значень. Приклади порядкової змінної:

- високий, середній, низький;
- відмінно, добре, погано;
- XS, S, M, L, XL.

Розглянемо ж методи для кодування цих типів даних. А саме, *One-hot* кодування і кодування за мітками.

Розглянемо це на прикладі даних зі змінними кольорів та розмірів одягу (Рис. 1.16). Перетворимо категоріальні ознаки таким чином, щоб тепер у нас була окрема колонка для кожного класу цих ознак.

size	color	target	size_le	size_hc	size_l	size_m	size_s	size_xl	size_xs	color_blue	color_green	color_red
xs	blue	1	4	0	0	0	0	0	1	1	0	0
xs	blue	1	4	0	0	0	0	0	1	1	0	0
s	red	1	2	1	0	0	1	0	0	0	0	1
l	red	0	0	3	1	0	0	0	0	0	0	1
m	blue	0	1	2	0	1	0	0	0	1	0	0

Рисунок 1.16 – дані з використанням *One-hot* кодуванням

Тепер ми можемо позбутися оригінальних категоріальних змінних, оскільки наші моделі не зможуть працювати з цими значеннями.

Також, ми втрачаємо структурну інформацію порядкових ознак за допомогою *One-hot* кодування, і тому у нас є інші способи перетворення

такого типу даних. А саме, кодування за мітками. Це навіть простіше, ніж кодування *One-hot*. Застосовуючи кодування міток, ми просто присвоюємо кожному класу категоріальних ознак унікальний номер. Наприклад (Рис. 1.17):

size	color	target	size_le	size_hc
xs	blue	1	4	0
xs	blue	1	4	0
s	red	1	2	1
l	red	0	0	3
m	blue	0	1	2

Рисунок 1.17 – дані з використанням кодування за мітками

1.5 Висновки до першого розділу

В цьому розділі ми розібрали методи обробки даних для подальшого тренування моделей. А саме, як шукати аномалії чи викиди в даних, як вирішувати проблему з високою корельованостю змінних і їх вплив на моделі. А також, як можна нормалізувати чи стандартизувати тренувальні данні. Даний процес аналізу і обробки даних, як правило, найбільш громіздкий по часу для рішення типових задач.

РОЗДІЛ 2 ТЕОРЕТИЧНІ ОСНОВИ МОДЕЛЕЙ ДЛЯ ТРЕНУВАННЯ ДАННИХ

2.1 Вступ

В даному розділі буде розглянуто основна теорія по моделям CART та нейронної мережі з використанням критерію оптимізації Адам. А також буде розібрано алгоритми для класифікації з багатьма вихідними етикетками. Після чого буде описано деякі недоліки та поради щодо розібраних методів.

2.2 Постановка задачі і опис алгоритму для нейронної мережі з методом зворотного поширення помилки

2.2.1 Опис поняття нейронної мережі

Нейронні мережі являють собою набір алгоритмів, змодельованих в загальних рисах з людського мозку, які призначені для розпізнавання закономірностей. Вони інтерпретують сенсорні дані з допомогою машинного сприйняття, маркування або кластеризації вихідних даних. Вхідні данні , які вони розпізнають, є числовими, що містяться в векторах, тому потрібно щоб всі реальні дані, будь то текст зображення, звук або тимчасові ряди, були переведені в числа. [11]

Нейронні мережі допомагають нам групувати і класифікувати. Наприклад, вони допомагають групувати немарковані дані відповідно до подібностей між прикладами входів і класифікувати дані, коли у них є маркований набір даних для навчання. Нейронні мережі можуть також отримувати функції, які передаються в інші алгоритми для кластеризації та класифікації; так що про глибокі нейронні мережі можна сказати як про компоненти більших додатків машинного навчання, які використовують

алгоритми для посилення навчання, класифікації і регресії. Тобто, в загальному вигляді, нейронна мережа представляє собою якийсь «чорний ящик», з допомогою якого ми розробляємо функціональну залежність між вхідними і вихідними даними. А ось і візуальний приклад (рис. 2.1).

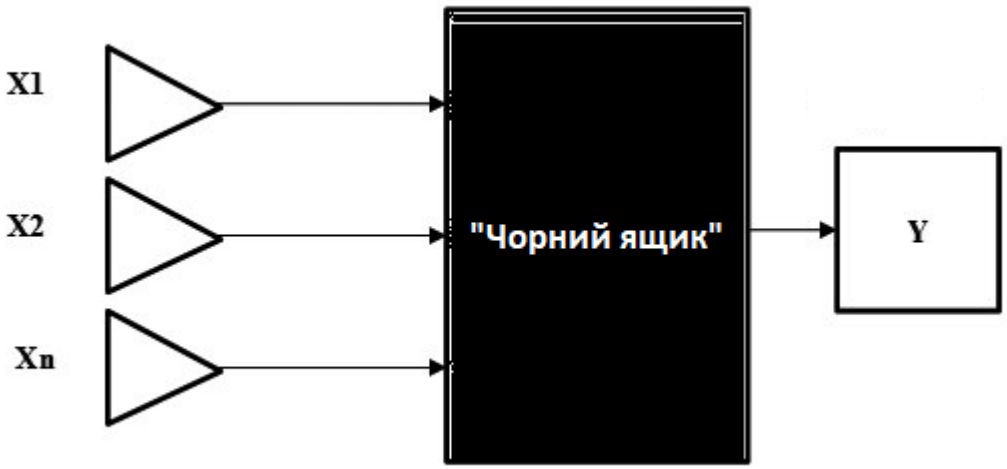


Рисунок 2.1 – Функціональна залежність

Знаючи звідки пішло словосполучення «нейронна мережа» і маючи базове уявлення для чого вона потрібна, тепер можна й поговорити про його математичну модель, та алгоритм пошуку функціональної залежності між вхідними і вихідними даними, а саме ,простіше кажучи, «відкрити чорний ящик» і подивитись, що ж там усередині.

2.2.2 Модель і принцип роботи багатошарової нейронної мережі

Нейронна мережа в першу чергу складається з шарів, а шари – з вузлів. Вузол – це просто місце, де відбувається обчислення, а його структура чимось схожа на нейрон в мозку людини, який спрацьовує, коли стикається з

достатніми стимулами (входи). Вузол поєднує введення даних з набором коефіцієнтів або ,як кажуть, вагові коефіцієнти, які або посилюють, або пом'якшують вхідні данні, надаючи тим самим певні змінені вхідні значення по відношенню до задачі, яку намагається вирішити алгоритм. [12]

Після чого ці продукти зважування (змінені данні) підсумовуються між собою, а потім сума проходить через так звану функцію активації вузла, щоб визначити, чи повинен і в якій мірі має цей сигнал протікати далі по мережі, щоб вплинути на кінцевий результат, наприклад, на акт класифікації. Наприклад, в нашій роботі ми використали сигмоїдну функцію активації:

$$f(x) = \frac{1}{1+\exp(-\alpha x)} \tag{2.1}$$

Зауваження. Якщо сигнали проходять через вузол, то кажуть що нейрон був "активований".

Ось візуалізація того, як може виглядати один вузол (рис. 2.2):

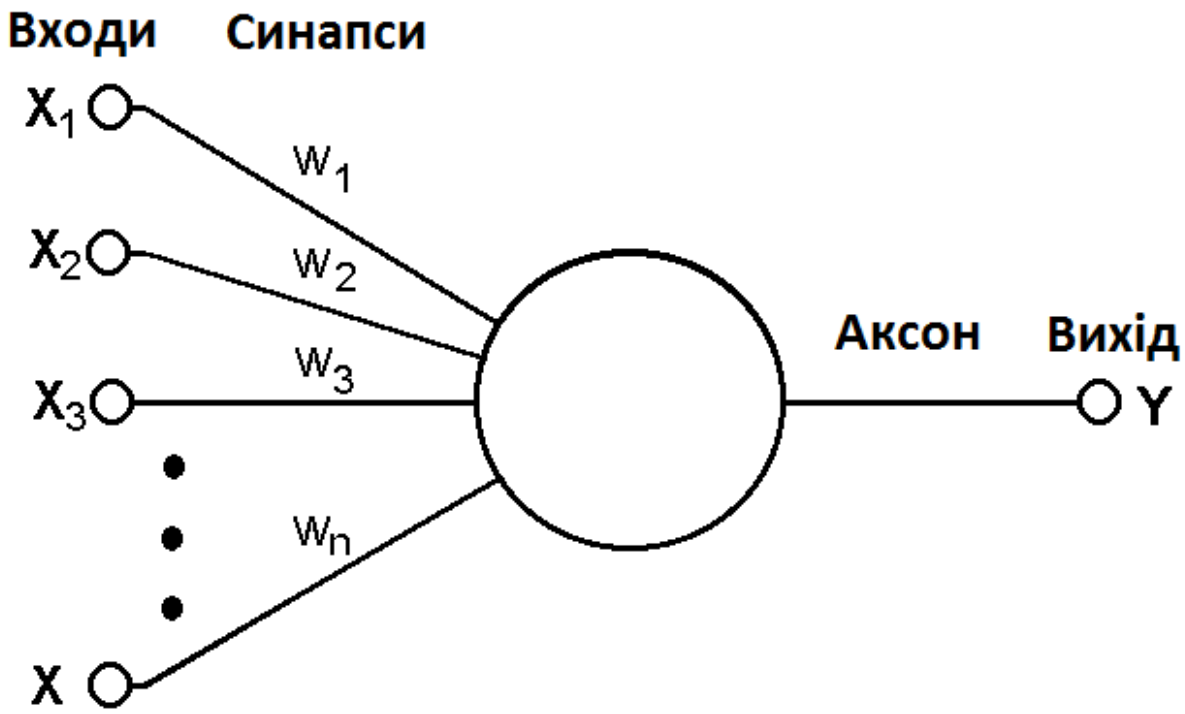


Рисунок 2.2 – Схема нейрона

де, вектор X_1 ... X_n — вхідні данні. Вектор w_1 ... w_n — вагові коефіцієнти. Y — вихідне значення.

Тепер можна поговорити про шари нейронної мережі, і на що впливає їх кількість. І так в першу чергу опишемо визначення цього ж шару.

Вузловий шар – це ряд нейроподібних перемикачів (вузлів), які включаються або вимикаються(активований і не активований) при проходженні вхідного сигналу через мережу. [13]

Вихідний сигнал кожного шару – це одночасно вхідний сигнал наступного шару, починаючи з початкового вхідного шару, що приймає ваші дані. На схемі цього процесу (рис. 2.3) це буде виглядати ось так, на прикладі трьох-шарової мережі:

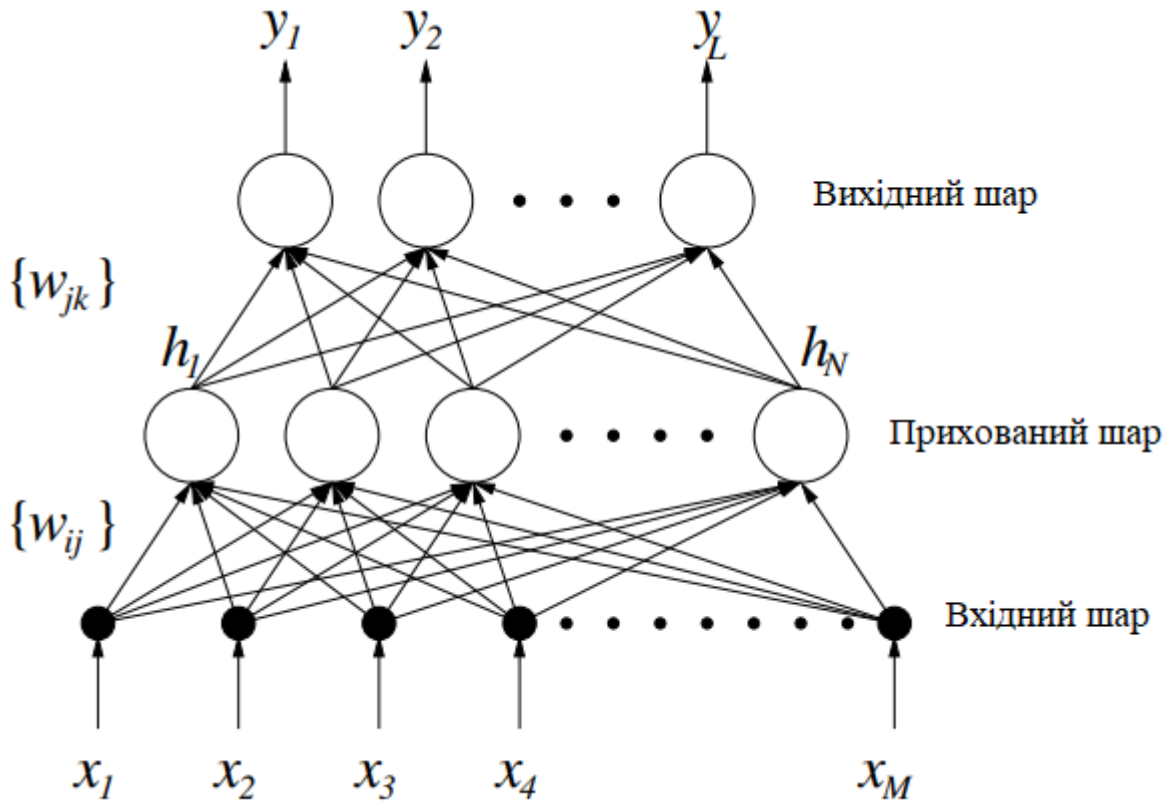


Рисунок 2.3 – Загальна модель трьох-шарової мережі

де, h_1 ... h_n — кількість прихованих шарів.

Багатошарові нейроні мережі відрізняються від більш поширених одношарових нейронних мереж своєю глибиною, тобто кількістю вузлових шарів, через які мають проходити дані в багатоступеневому процесі розпізнавання образів.

Більш ранні версії нейронних мереж, такі як перші перцептрони, були неглибокими, що складаються з одного вхідного і одного вихідного шару, і не більше одного прихованого шару між ними. Якщо в мережі понад трьох рівнів (включаючи входи та результати), то такий процес кваліфікуються як "глибоке" навчання. Це означає що в мережі більше одного прихованого шару. [14]

У мережах з глибоким вивченням кожен шар вузлів навчається за певним набором функцій на основі результатів попереднього рівня. Чим далі ви переходите в нейронну мережу, тим складніше розпізнаються вузлами функції, оскільки вони об'єднують і переструктуровують функції з попереднього шару.

Нашою метою при використанні нейронної мережі є досягнення точки мінімальної похибки якомога швидше. Ми проходимо через нашу мережу стільки разів скільки нам буде необхідно для досяганні нашої мети. А мета наша є стан певних параметрів (вагових коефіцієнтів), в яких вони здатні виробляти досить точні класифікації та прогнози. А початковим станом нашого процесу є стан, в якому започатковано наші ваги.

Сам процес обходу мережі включає в себе безліч етапів, і кожен з них схожий на ті, що робилися до і після. Кожен крок для нейронної мережі включає в себе вгадування, вимір похибки і невелике оновлення ваг, інкрементного коригування коефіцієнтів.

Збір вагових коефіцієнтів, незалежно від того, чи знаходяться вони в початковому або кінцевому стані, також називається моделлю, оскільки він являє собою спробу змоделювати зв'язок даних з мітками наземної істини (наприклад, [0, 0, 1, 0, 1, …]), щоб зрозуміти структуру даних. Моделі зазвичай

починаються погано і закінчуються менш погано, змінюючись з часом по мірі відновлення нейронною мережею своїх параметрів.

Це тому, що нейронна мережа створюється в невіданні. Вона не знає, які ваги і відхилення будуть найкращим чином відображати вхідні дані, щоб зробити правильні здогади. Вона повинна починати з припущень, а потім намагатися робити більш точні припущення послідовно, у міру того як вчиться на своїх помилках.

Ось просте пояснення того, що відбувається під час навчання з нейронною мережею.

2.2.3 Опис алгоритму для нейронної мережі з методом зворотного поширення помилки

І так, тепер розберемо алгоритм для трьохшарової нейронної мережі з методом зворотного поширення помилки. Таким чином, розіб’ємо наш алгоритм на такі етапи:

- Ініціалізація – початкові ваги застосовуються до всіх нейронів.
- Пряме поширення – вхідні дані з навчальної вибірки проходять через нейронну мережу, а вихідні дані обчислюються.
- Функція помилки – оскільки ми працюємо з навчальною вибіркою, то відомий правильний висновок. Визначається функція помилки, яка фіксує різницю між правильним виходом і фактичним виходом моделі з урахуванням ваги поточної моделі (іншими словами, "наскільки погано" модель розраховує результати).
- Зворотне поширення – метою розмноження є зміна ваг нейронів, щоб звести функцію помилки до мінімуму.
- Оновлення ваги – ваги змінюються на оптимальні значення за результатами виконання алгоритму розмноження.

- Повтори до конвергенції – оскільки ваги оновлюються з невеликим кроком дельти за раз, для того, щоб мережа могла вчитися, потрібно кілька ітерацій. Після кожної ітерації сила градієнта спуску оновлює ваги в бік все меншою і меншою функції глобальних втрат. Кількість ітерацій, необхідних для конвергенції, залежить від швидкості навчання, мета-параметрів мережі і використовуваного методу оптимізації. [15]

В кінці цього процесу модель готова робити прогнози для невідомих вхідних даних. Нові дані подаються в модель, виконується прямий прохід, і модель генерує прогноз. Розберемо кожен етап на прикладі такої моделі (рис. 2.4):

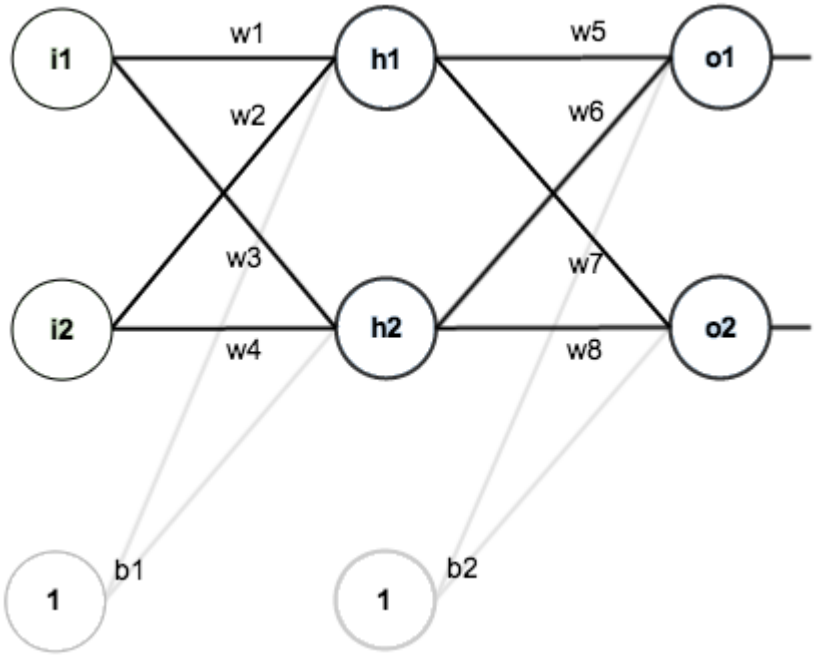


Рисунок 2.4 –Модель трьох-шарової мережі

Як можна побачити знизу зображення в цій моделі використовуються два спеціальні нейрони. Їх називають нейронами зміщення – це спеціальний нейрон, що додається до кожного шару нейронної мережі, який просто зберігає значення 1, що дозволяє переміщати або "транслювати" функцію активації вліво або вправо на графіку.

Без зміщення нейрона, кожен нейрон приймає вхідний сигнал і примножує його на вагу, не додаючи нічого іншого до рівняння. Так, наприклад, неможливо ввести значення 0 і вихід 2. в багатьох випадках для генерації необхідних вихідних значень необхідно перемістити всю функцію активації вліво або вправо, що стало можливим завдяки зміщенню.

Хоча нейронні мережі можуть працювати без зміщення нейронів, в дійсності вони майже завжди додаються, і їх вага оцінюється як частина загальної моделі.

Опис етапу ініціалізації

Установка ваг на початку, перед тренуванням моделі. Типова стратегія в нейронних мережах полягає в випадкової ініціалізації ваг, а потім можна приступати й до оптимізації. Тому після цього кроку ми маємо початковий масив вагових коефіцієнтів: $w_1, \dots, w_i, \dots$ для всієї мережі. [16]

Опис етапу прямого поширення

Далі ми проходимо через усі нейрони зліва на право, отримуючи виходи останнього шару (вихідний шар). Математично вони обчислюються такою рекурсивною функцією :

$$x_j^{(l)} = f(node_j^{(l)}) = f(\sum_j w_{jk}^{(l)} * x_j^{(l-1)} + b_j^{(l)}) \quad (2.2)$$

Ця формула розраховує вихід j – го нейрону в l – шарі, де в верхньому індикаторі в дужках вказується номер шару і $b_j^{(l)}$ – вектор зміщення для l шару.

Сама ж функція f – це наша активаційна функція, приклад якої був уже вказаний вище.

Перевірка функції помилки

Маючи експериментально отримані і фактичні дані за один момент часу, ми можемо розрахувати функцію помилки для нашої мережі:

$$E_{total} = \frac{1}{2}\sum_i^N ({y_{target}}^{(i)} - {y_{out}}^{(i)})^2 \quad (2.3)$$

де, i – вибраний вихідний нейрон. ${y_{target}}^{(i)}$ –вибраний елемент тестового ряду для перевірки схожості між елементом виходу.

Опис етапу зворотного поширення і оновлення ваг

В першу чергу ми розраховуємо похідну для нашої помилки по ваговому коефіцієнту, який хочемо змінити:

$$\frac{\delta(E_{total})}{\delta\left(w_{jk}^{(l)}\right)} = \frac{\delta(E_{total})}{\delta\left(\sum_j w_{jk}^{(l)} * x_j^{(l-1)} + b_j^{(l)}\right)} * \frac{\delta\left(\sum_j w_{jk}^{(l)} * x_j^{(l-1)} + b_j^{(l)}\right)}{\delta\left(w_{jk}^{(l)}\right)} \quad (2.4)$$

де, j – номер нейрона.

Далі для l – шару (output layer) буде:

$$\frac{\delta(E_{total})}{\delta\left(node_j^{(l)}\right)} = \frac{\delta(E_{total})}{\delta({y_{out}}^{(i)})} * \frac{\delta\left(f\left(\sum_j w_{jk}^{(l)} * x_j^{(l-1)} + b_j^{(l)}\right)\right)}{\delta\left(\sum_j w_{jk}^{(l)} * x_j^{(l-1)} + b_j^{(l)}\right)} =$$

$$= \left(\frac{1}{2} * \frac{\delta}{\delta({y_{out}}^{(i)})}\left({y_{target}}^{(i)} - {y_{out}}^{(i)}\right)^2\right) * \left({y_{out}}^{(i)} * \left(1 - {y_{out}}^{(i)}\right)\right) =$$

$$= -{y_{out}}^{(i)} * \left(1 - {y_{out}}^{(i)}\right) * \left({y_{target}}^{(i)} - {y_{out}}^{(i)}\right)$$

А, для (l-1), (l-2), … – шару $\frac{\delta(E_{total})}{\delta\left(node_j^{(l)}\right)}$ буде дорівнювати:

$$\frac{\delta(E_{total})}{\delta\left(node_j^{(l)}\right)} = \sum_{k \in Outputs(j)} \frac{\delta(E_{total})}{\delta\left(node_k^{(l-1)}\right)} * \frac{\delta\left(node_k^{(l-1)}\right)}{\delta\left(node_j^{(l)}\right)} =$$

$$= \frac{\delta\left(node_k^{(l-1)}\right)}{\delta\left(node_j^{(l)}\right)} = \frac{\delta\left(node_k^{(l-1)}\right)}{\delta\left(f\left(node_j^{(l)}\right)\right)} * \frac{\delta\left(f\left(node_j^{(l)}\right)\right)}{\delta\left(node_j^{(l)}\right)} =$$

$$= w_{ij} * f\left(node_j^{(l)}\right) * \left(1 - f\left(node_j^{(l)}\right)\right)$$

де, $Outputs(j)$ – виходи з j - го вузла, які є входами для k – х вузлів.

І тепер пробігши по всій мережі познаходивши похідні справа – наліво, можна змінити наші вагові коефіцієнти таким чином:

$$w_{ij}(n+1) = w_{ij}(n) - \eta * \frac{\delta(E_{total})}{\delta\left(w_{ij}^{(l)}\right)} \tag{2.5}$$

Таким чином завершується одна ітерація циклу навчання. [17]

2.2.4 Умова закінчення навчання

Тепер розібравши роботу навчання мережі, треба зрозуміти коли воно має зупинитись. Насправді, варіантів умов зупинок буває декілька. Коли у нас є певний набір тестових даних, який набагато більший за кількість вхідних і вихідних вузлів, ми можемо зупинити цикл, коли повністю переберемо всю нашу тестову вибірку. І коли ми надаємо певний ліміт значенню помилки нашої мережі, і зупиняємо цикл коли зуміли досягнути наше надане значення ($E_{total} < const$).

В цій же роботі, ми використовуємо обидва способи. А саме, ми не дивимось на помилку, поки повністю не переберемо наш масив тесових даних, а після одного перебору ми задаємо певну константу для помилки нашої нейронної мережі, і вже кожного разу порівнюємо між собою ці значення. Плюсом такого способу, є те що якщо ми одразу б вибрали 2-й варіант, то ми знайшли функціональну залежність тільки для частини даних, а це не означає що вона буде ж такою ж самою для інших. Наприклад, якщо в ряду нашої тестової вибірки є викиди.

2.3 Висновки до другого розділу

В цьому розділі ми розібрали модель, нейронної мережі з методом зворотного поширення помилки. В наступному розділі, ми подивимось вже на практичну реалізацію цієї моделі, і порівняємо отримані результати.

РОЗДІЛ 3 АНАЛІЗ РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ І ПОРІВНЯННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

3.1 Вступ

В цьому розділі буде розібрано реалізацію і розробку моделі для класифікації механізмів дії лікарських речовин, використовуючи моделі логістичну та нейронну мережу з використанням критерія оптимізації Adam. Програма була написана на мові Python. Ця мова була вибрана, тому що на ній реалізовано велика кількість бібліотек, які змогли спростити нам реалізацію багатьох процедур, і можливість натренувати якісну модель.

3.2 Аналіз даних до тренування моделей

Для початку ми отримаємо короткий огляд наборів даних і їх форм. Наведені нижче таблиця (Рис. 3.1) охоплює дані для перших 10 рядків.

	sig_id	cp_type	cp_time	cp_dose	g-0	g-1	g-2	g-3	g-4	g-5	g-6
1	id_000644bb2	trt_cp	24	D1	1.062	0.5577	-0.2479	-0.6208	-0.1944	-1.012	-1.022
2	id_000779bfc	trt_cp	72	D1	0.0743	0.4087	0.2991	0.0604	1.019	0.5207	0.2341
3	id_000a6266a	trt_cp	48	D1	0.628	0.5817	1.554	-0.0764	-0.0323	1.239	0.1715
4	id_0015fd391	trt_cp	48	D1	-0.5138	-0.2491	-0.2656	0.5288	4.062	-0.8095	-1.959
5	id_001626bd3	trt_cp	72	D2	-0.3254	-0.4009	0.97	0.6919	1.418	-0.8244	-0.28
6	id_001762a82	trt_cp	24	D1	-0.6111	0.2941	-0.9901	0.2277	1.281	0.5203	0.0543
7	id_001bd861f	trt_cp	24	D2	2.044	1.7	-1.539	5.944	-2.167	-4.036	3.695
8	id_0020d0484	trt_cp	48	D1	0.2711	0.5133	-0.1327	2.595	0.698	0.5846	-0.2633
9	id_00224bf20	trt_cp	48	D1	-0.3014	0.5545	-0.2576	-0.139	-0.6487	-0.6057	-0.7549
10	id_0023f063e	trt_cp	48	D2	-0.063	0.2564	-0.5279	-0.2541	-0.0182	-1.537	-0.218

Рисунок 3.1 – вхідні дані

Всього в тренувальному наборі 23814 обсервацій. А також, 876 змінних з яких:

- “sig_id” – це унікальний первинний ключ вибірки;
- “g-” змінні – означають дані про експресію гена. Всього 772;
- “c-” змінні – означають дані про життєздатність клітин. Всього 100;
- “cp-type” – позначає зразки, оброблені сполукою (cp_vehicle) або з контрольним збуренням (ctrl_vehicle);
- “cp-time” – тривалість лікування (24, 48, 72 години);
- “cp-dose” – дозування лікування (висока, низька).

Почнемо з побудови розподілів різних предикторів і цільових ознак окремо, а потім перейдемо до багатофункціональних візуальних зображень і кореляцій. Тут ми маємо справу з угрупованням ознак за групами.

3.2.1 Візуалізація індивідуальних особливостей

Особливості лікування (Рис. 3.2). Це ті предикторські особливості, які в більш загальному плані описують, як обробляли зразок, з точки зору дози, тривалості; і чи було це «справжнім» лікуванням чи контролем.

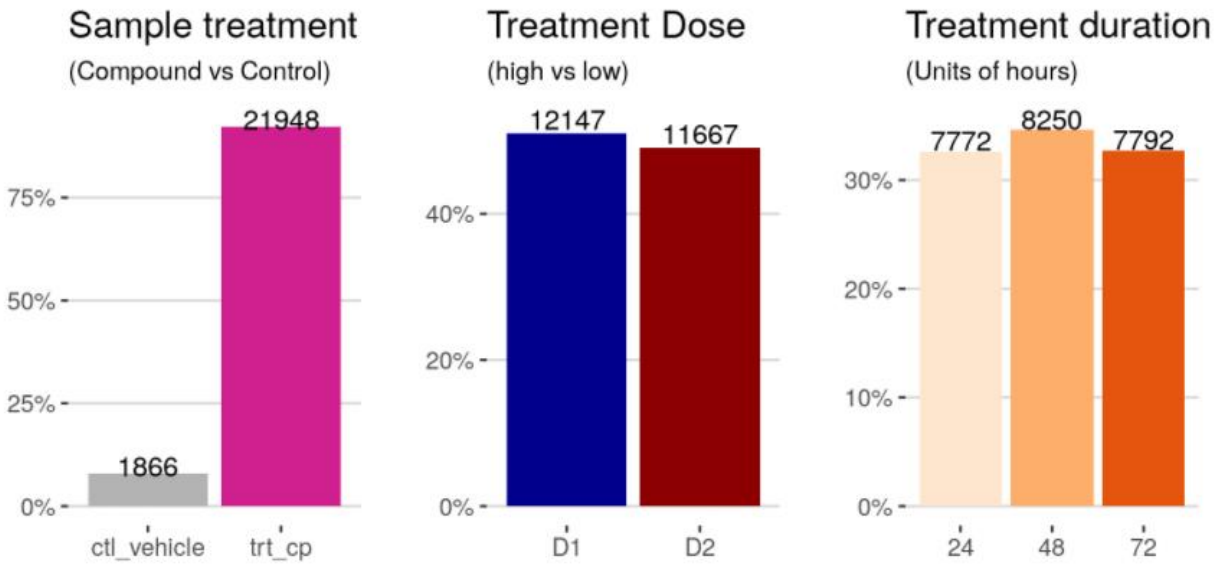


Рисунок 3.2 – особливості лікування

Дивлячись на гістограми що вище можна сказати що:

- Переважна більшість обробок - це обчислювальні обробки ("trt_cp"), в порівнянні з приблизно 8% обробок контрольного обурення ("ctl_vehicle"). Контролі не мають MOA.
- Доза лікування має дві категорії, D1 проти D2, які кодують високі проти низьких доз. Вони приблизно рівномірно збалансовані, як і 3 категорії тривалості лікування - 24 години, 48 годин або 72 години.

Особливості експресії генів (Рис. 3.3). По суті, це анонімізовані характеристики, відомі від "g-0" до "g-771". Значення числові, тому подивимось на плотність для перших 4-х призначених генів у якості пропозиції.

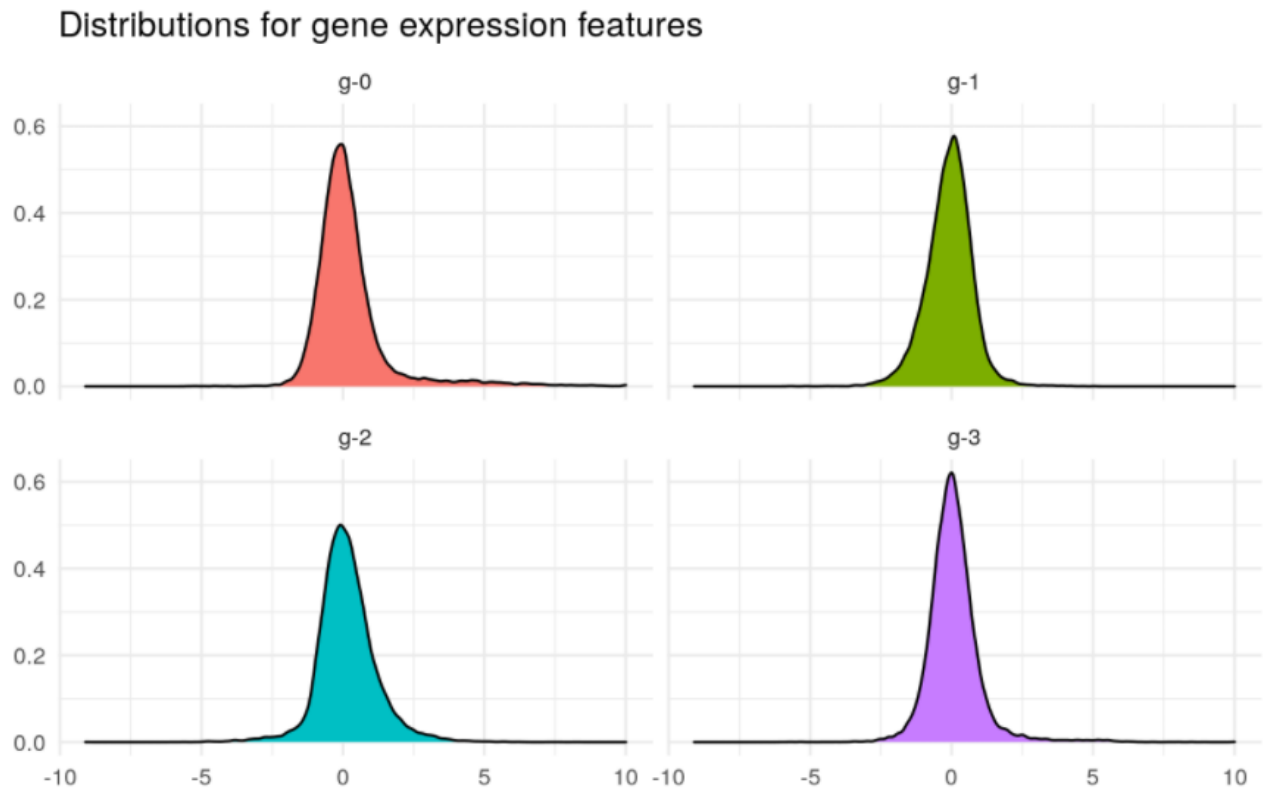


Рисунок 3.3 – особливості експресії генів

Ці розподіли виглядають досить нормально, що добре. У деяких з них є трохи перекіс, але нічого, що не повинно вимагати перетворення.

Особливості життєздатності клітин (Рис. 3.4). Подібно до ознак гена, ознаки життєздатності клітин анонімні, позначені від “c-0” до “c-99”; 100 особливостей. Їх розподіл виглядає наступним чином.

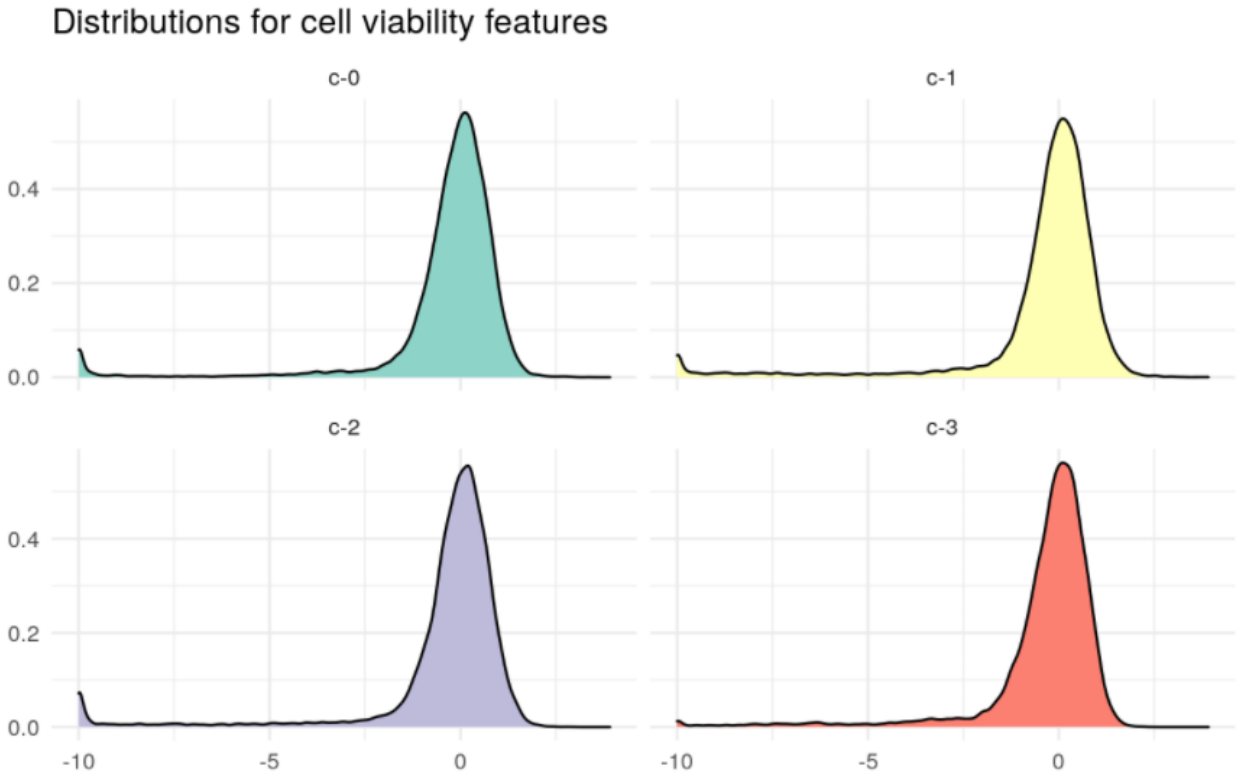


Рисунок 3.4 – особливості життєздатності клітин

Також досить нормально, але з помітними горбами навколо значень -10.

Цільові змінні. Всі таргети є двійковими стовпцями, що вказують на те, чи реагує певний тип клітин

Нашою проблемою є проблема класифікації за кількома мітками, і тому рядки (тобто зразки ліків) можуть мати кілька міток (тобто активними можуть бути більш одного цільового класу). Спочатку розглянемо розподіл того (Рис. 3.5), скільки цільових класів може бути одночасно активними. на препарат чи ні.

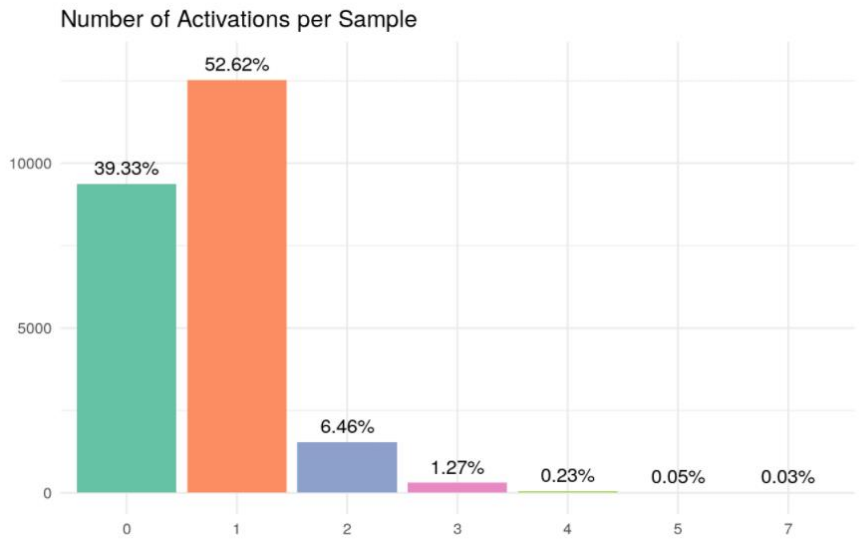


Рисунок 3.5 – кількість цільових класів

Близько 39% навчальних зразків взагалі не мають приміток MoA (наприклад, всі цільові класи мають нульові значення). Це дозволяє пояснити розріджений цільової блок даних, якщо 40% з них повністю порожні.

Найбільша група, з трохи більше 50%, має рівно 1 анотацію MoA (наприклад, один клас = одне значення "1" в своєму рядку).

Для більш ніж 1 анотації MoA ми бачимо хвіст, який простягається до 7 одночасних MoA (для 0.03% випадків). Тільки для 2-х MoA трохи вище 5%, а для 3-х - нижче 1%. Кожен другий випадок значно рідше. Відзначимо, що з 6 цільовими класами взагалі немає жодного випадку.

3.2.2 Візуалізація багатофункціонального взаємодії

Тепер, коли у нас з'явилося набагато краще уявлення про те, як ведуть себе індивідуальні змінні, подивимось на їх взаємодію. Будемо використовувати той же порядок наборів особливостей, що і вище.

Особливості лікування (Рис. 3.6). Порівняння 3-х функцій лікування вимагає граневої сітки. Два об'єкти охоплюють горизонтальну і вертикальну осі сітки, 3-й визначає ділянку всередині кожної огранки.

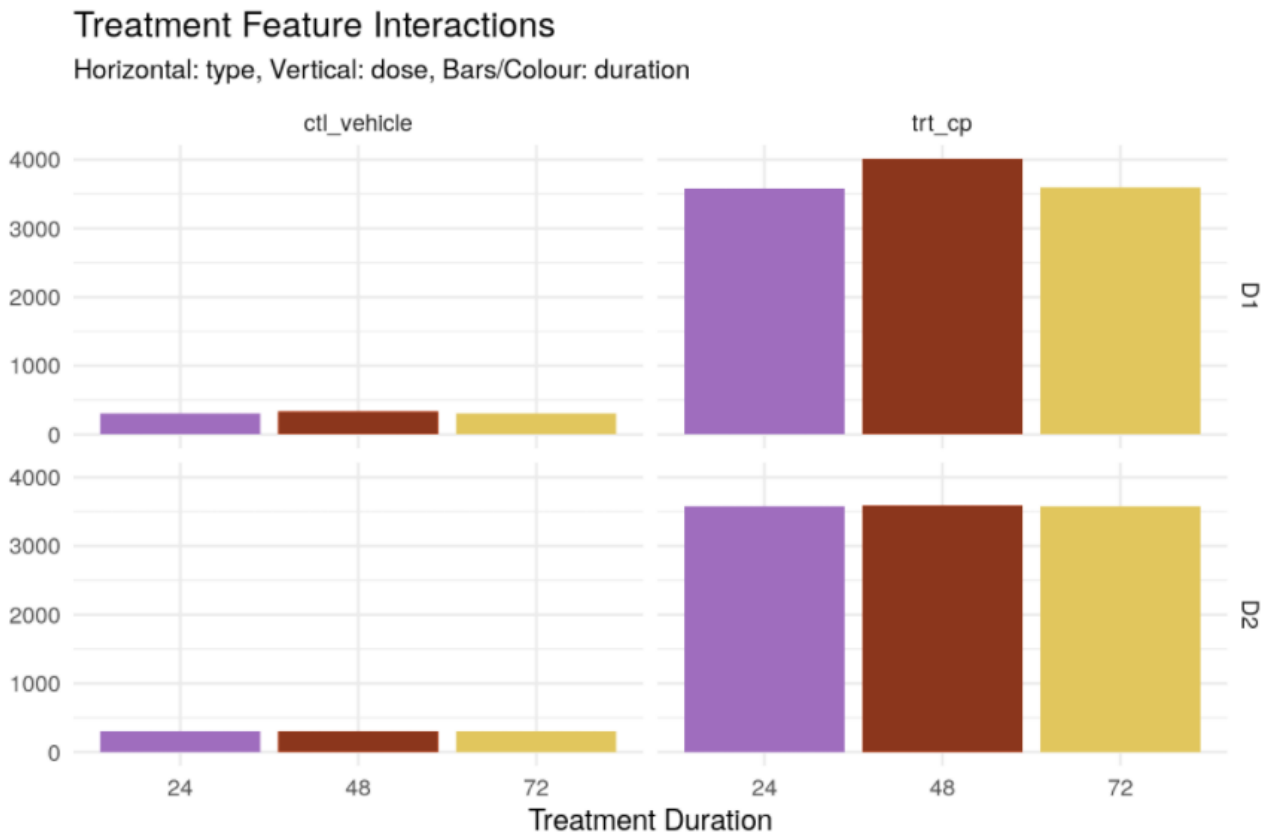


Рисунок 3.6 – гранева сітка особливостей лікування

Отримана картина відповідає загальному вигляду, який ми бачили вище. Контрольні процедури також рідкісні в залежності від дозування і тривалості.

Однак помітною відмінністю є дещо вищий відсоток 48-годинного лікування при дозах D1 (як контрольних, так і комбінованих) в порівнянні зі значно більш рівномірно розподіленими стовбцями D2.

Особливості експресії генів (Рис. 3.7). Ось де все може заплутатися, з 772 особливостями в цій групі. З іншого боку, який би метод тут не працював, він повинен працювати і для меншого набору ознак життєздатності осередків. Виходячи з цього огляду, має бути достатньо детально розглянути підмножину функцій.

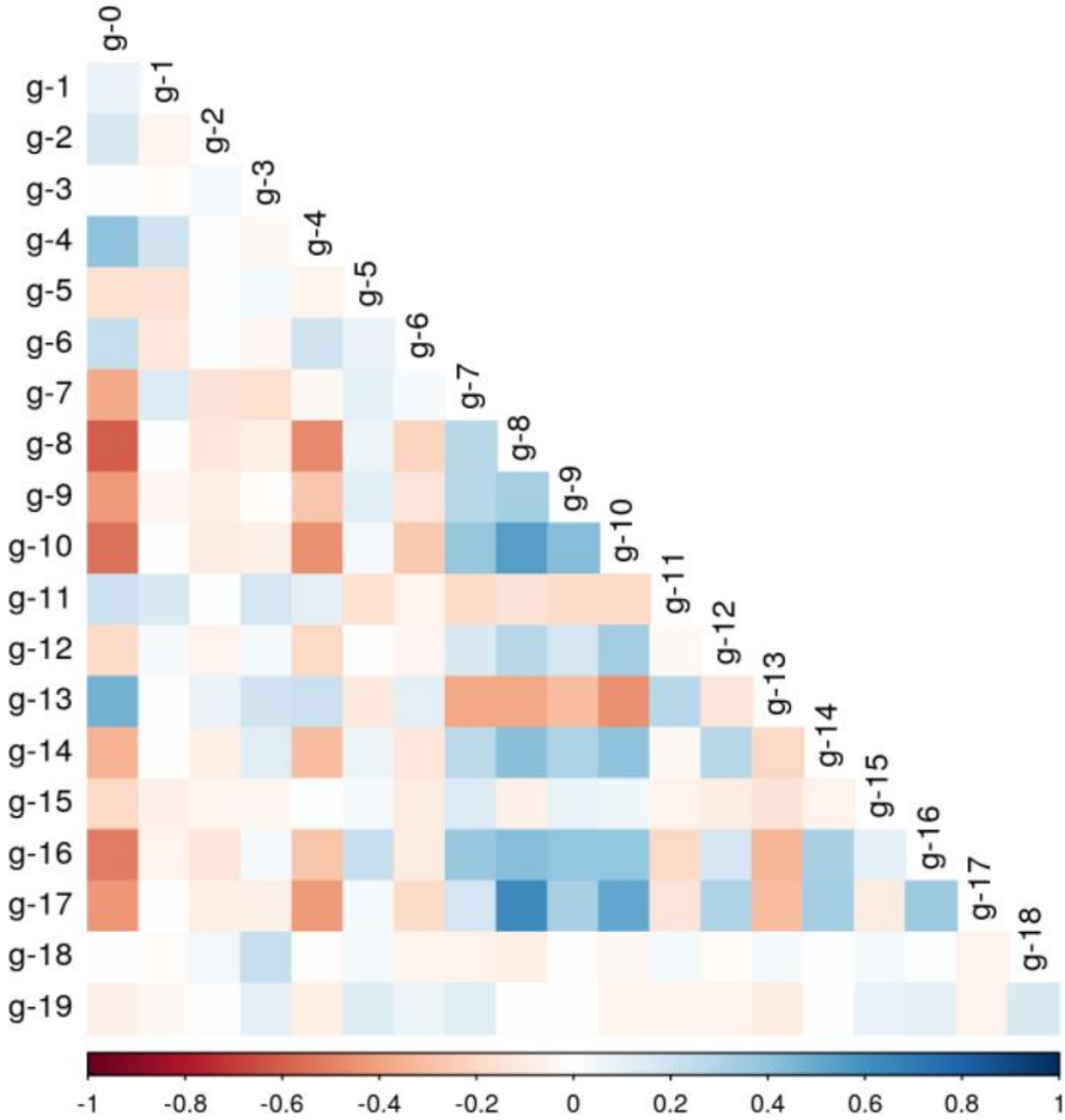


Рисунок 3.7 – кореляційна матриця експресії генів

Тут у нас перші 20 генів. Певні сильніші кореляції очевидні: наприклад, "g-0" проти "g-8" (антикореляція), або "g-8" проти "g-17".

До ознак без значної кореляції відносяться "g-18", "g-19", а також "g-2" і "g-3".

Особливості життєздатності клітин (Рис. 3.8). Тепер ми можемо зробити той самий аналіз особливостей життєздатності клітин. Тут у нас менше функцій (лише 100). Зробимо оглядовий графік кореляції.

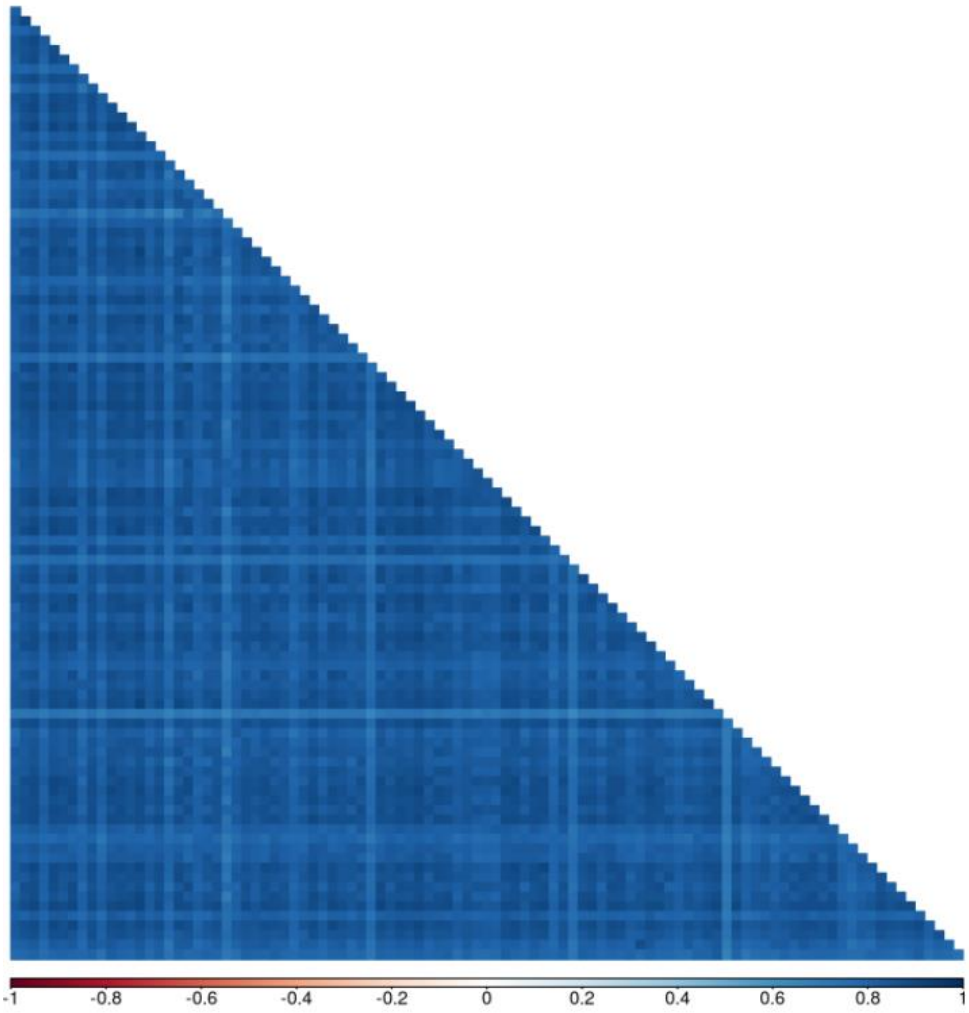


Рисунок 3.8 – кореляційна матриця життєздатності клітин

З огляду на те, що ми бачили вище для розподілів, зміщених в сторону -10, ця картина не зовсім несподівана. Варто подумати про видалення цих крайніх негативних значень, щоб більше дізнатися про співвідношення між рештою точками даних.

На цьому етапі досить поглянути на перші 10 ознак (Рис. 3.9). Тут виберемо варіант відображення, який відображає значення коефіцієнтів безпосередньо, з колірним кодуванням, еквівалентної матриці наведеної вище.

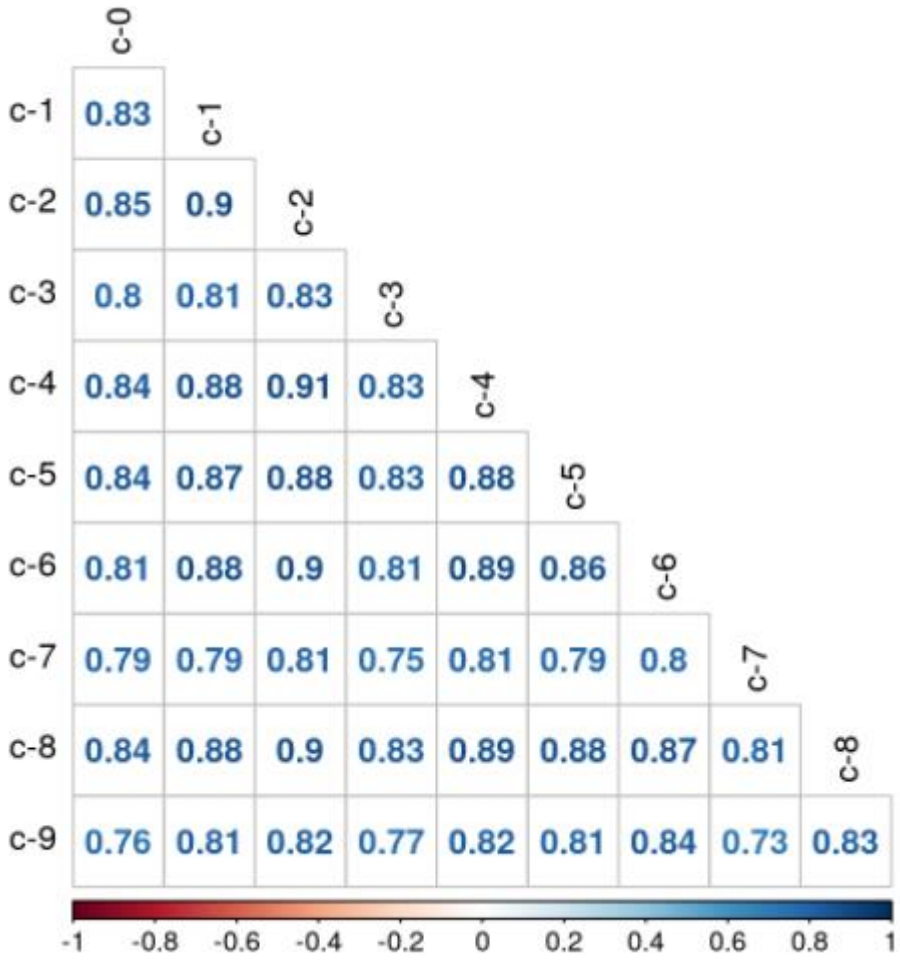


Рисунок 3.9 – кореляційна матриця життєздатності клітин

Значення варіюються від 0,75 до 0,9. Це досить сильні кореляції.

3.2.3 Зменшення розмірності за допомогою PCA

З огляду на значну кількість кореляцій в гені і, особливо, в особливості клітини, протестуємо метод зменшення розмірності PCA, щоб подивитися, наскільки ми можемо зменшити наш простір ознак.

Почнемо з особливостей гена, потім подивимося на особливості клітини.

Характеристики гена

Подивимось яку кількість інформації ми втрачаємо при зміні розмірності (Рис. 3.10).

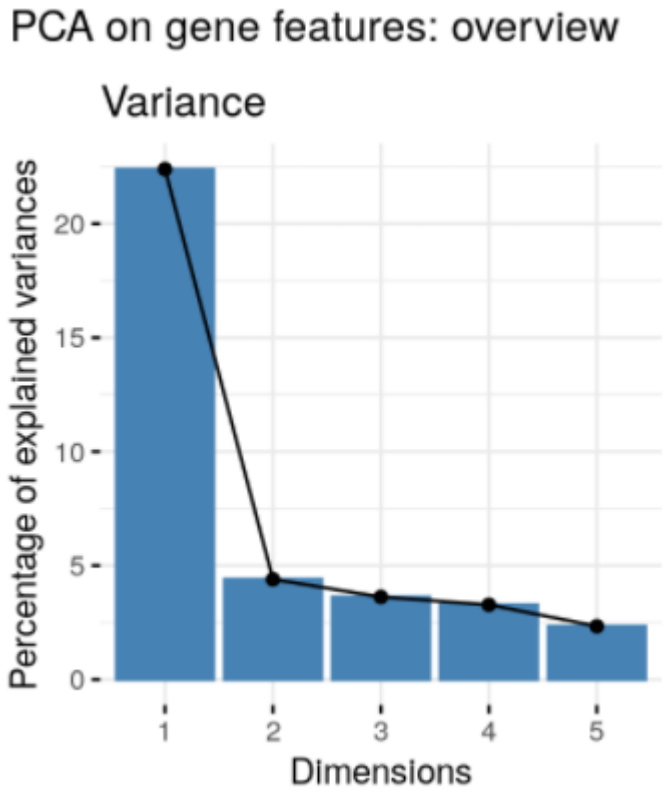


Рисунок 3.10 – залежність втрати інформації від розмірності характеристик генів

Похибка при композиції до 1 змінної становить близько 25%, в той час як при більших розмірностях вона становить менше 5%, що є помітним зниженням.

Розглянемо 15 кращих змінних (Рис. 3.11) в кожному вимірі (пунктирна горизонтальна лінія показує очікуване значення для рівномірного розподілу).

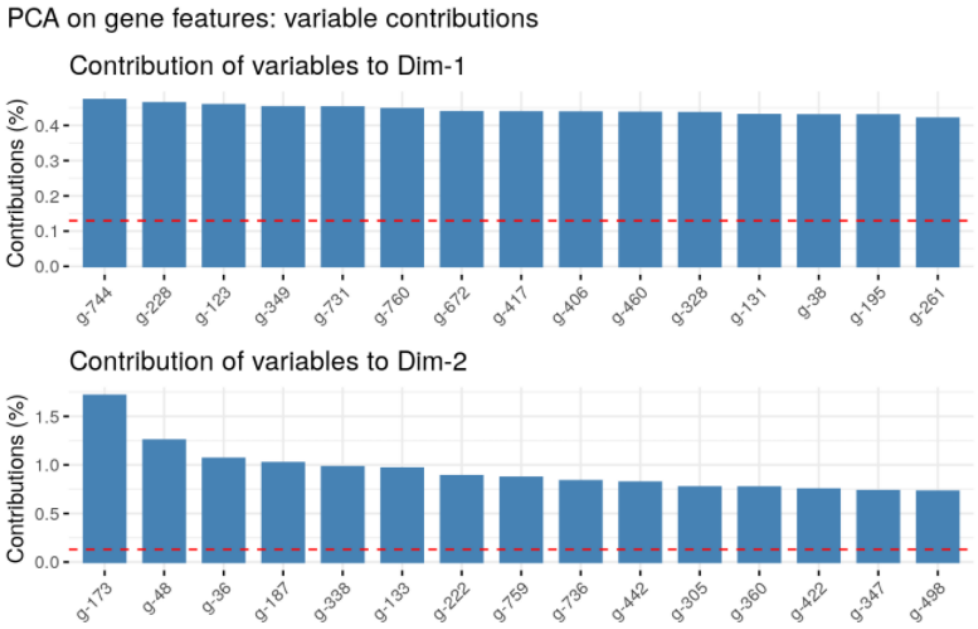


Рисунок 3.11 – вплив змінних на вимір після їх композиції

Характеристики клітини

Тепер ми можемо зробити це ж саме, що з особливостями клітин (Рис. 3.12).

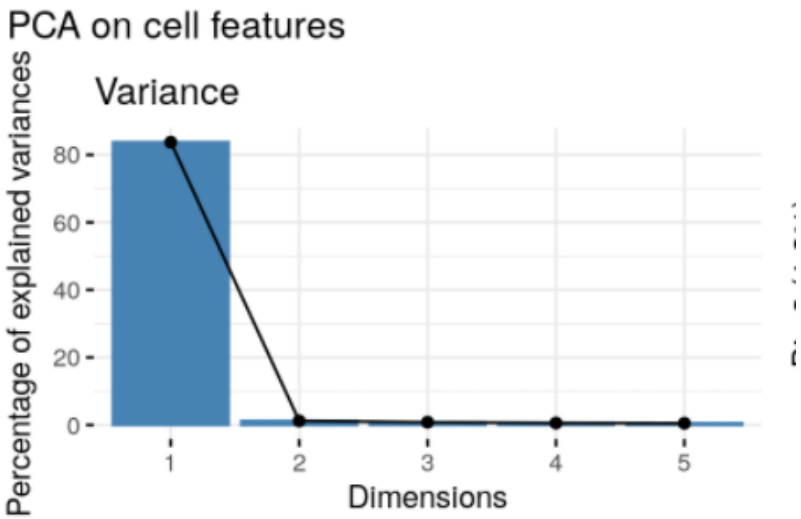


Рисунок 3.12 – залежність втрати інформації від розмірності характеристик клітин

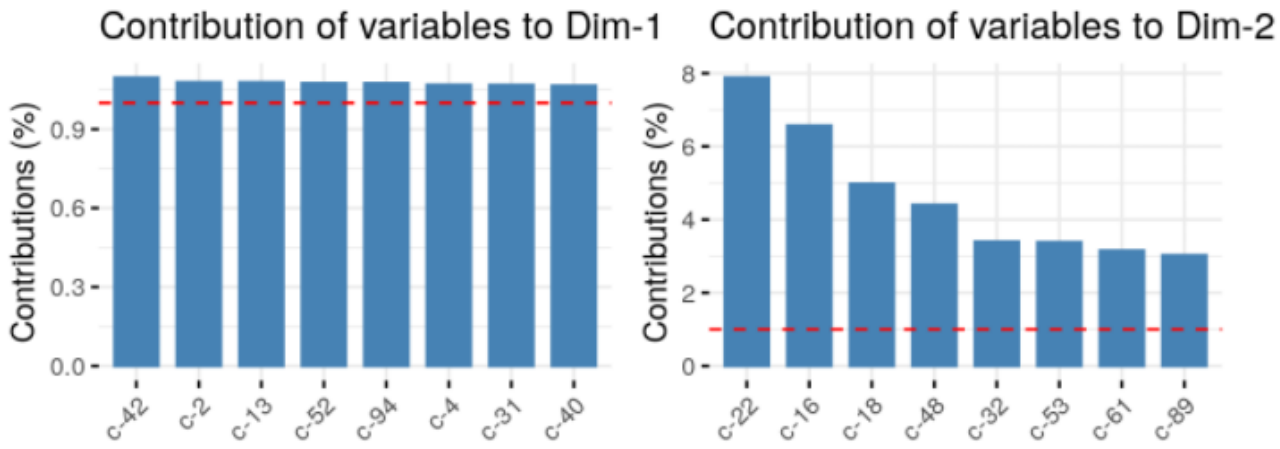


Рисунок 3.13 – вплив змінних на вимір після їх композиції

3.3 Тренування моделі та кодування даних

3.3.1 Препроцесінг даних

По-перше, нам потрібно зробити деяку препроцесінг, щоб отримати дані в потрібному вигляді. Нейронні мережі відносяться до "прискіпливим" моделям, в тому сенсі, що вони можуть працювати тільки з числовими входами (крім, скажімо, простих дерев рішень) без пропущених значень. Крім того, їх стиль навчання, заснований на градієнтному спуску, працює найкраще, якщо вхідні дані стандартизовані. Тут будемо використовувати простий розподіл "train / test".

Поділ стратифікований за *cp_type*, який, безумовно, має найбільший дисбаланс і вплив особливостей лікування Тому щоб збалансувати вибірку, візьмемо декілька наборів всі рядків з «cp_vehicle» (копіювання).

Тепер ми визначимо «пайплайн препроцесінгу». Це складається з наступних кроків:

- Кодуємо ознаки *cp_type* і *cp_dose* як цілі числа. Так як у цих двох категорій є тільки 2 рівня кожен, то використаємо кодування за мітками.

- Нормалізація функції *cp_time*. У цьому кодуванні описується тривалість, тому ми можемо розглядати її як числову характеристику.
- PCA. Тут ми застосовуємо його до ознак гена і клітини окремо.

3.3.2 Тренування моделі

Для нашої задачі було вирішено взяти NN з критерієм оптимізації Adam. Розглянемо розмірність і кількість шарів в моделі (Рис. 3.14).

```
NeuralNetwork(
  (input_layer): Linear(in_features=658, out_features=700, bias=True)
  (relu1): ReLU()
  (dropout1): Dropout(p=0.5, inplace=False)
  (hidden_layer1): Linear(in_features=700, out_features=1000, bias=True)
  (relu2): ReLU()
  (hidden_layer2): Linear(in_features=1000, out_features=800, bias=True)
  (relu3): ReLU()
  (hidden_layer3): Linear(in_features=800, out_features=500, bias=True)
  (relu4): ReLU()
  (hidden_layer4): Linear(in_features=500, out_features=200, bias=True)
  (relu5): ReLU()
  (output_layer): Linear(in_features=200, out_features=206, bias=True)
)
```

Рисунок 3.14 – опис структури моделі

Щоб запобігти перенавчанню, побудуємо графік похибок на тестовій та навчальній виборці в залежності від епохи(крок) навчання (Рис. 3.15).

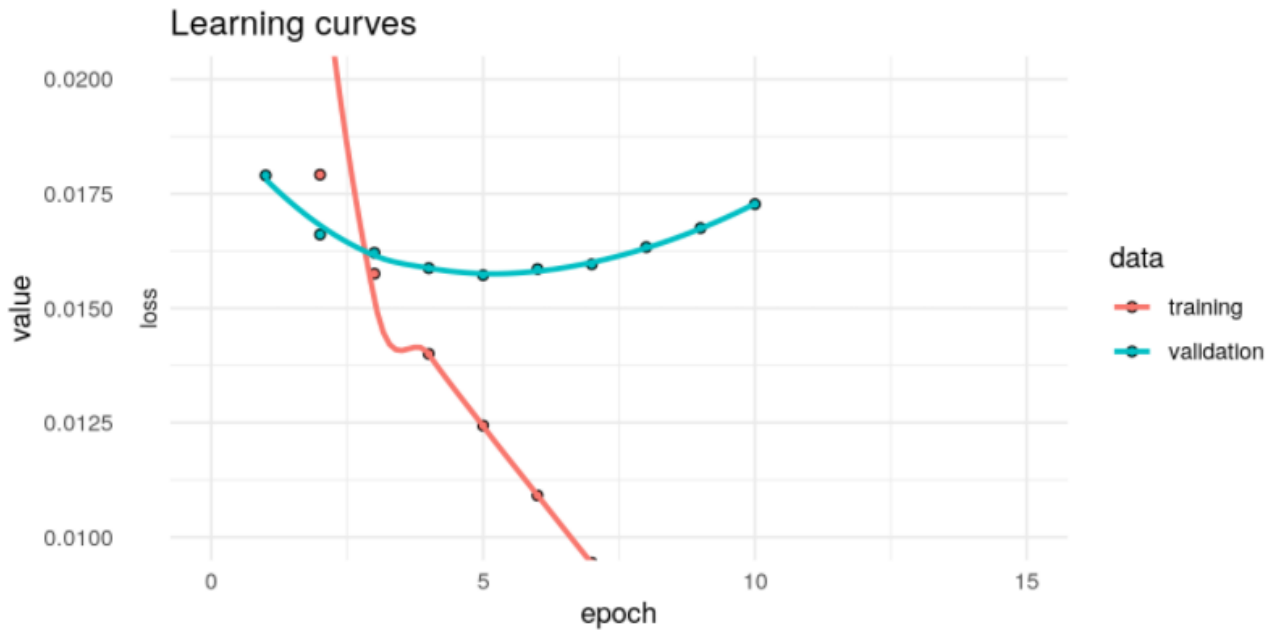


Рисунок 3.15 – залежність похибок від епохи навчання

Як можна зрозуміти, навчання було зупинено на 5-му кроці.

3.4 Висновки до третього розділу

В даному розділі було повністю розібрано весь пайплайн проекту від аналізу сирих даних до побудови моделі класифікації MoA. З основних проблем даних була висока незбалансованість таргетів та деяких ознак, висока кореляція характеристик генів і клітин. Остаточна оцінка моделі на тестових даних становить 0.016, що досить непогано.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого. Головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Далі наведено маркетинговий аналіз стартап проекту.

В межах цього етапу:

а) розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;

б) аналізуються ринкові можливості щодо його реалізації;

в) на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

4.1. Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

а) зміст ідеї (що пропонується);

б) можливі напрямки застосування;

в) основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);

г) чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці (таблиця 4.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	**Напрямки застосування**	**Вигоди для користувача**
Дана комплексна система дозволяє розвязати проблему пошуку нових ліків. На основі результатів класифікації відбувається розв’язання цієї проблеми –прогоноз механізму дії на людину різної кількості амінокислот.	Створення нових ліків	Зменшеня собівартотості ліків
	2. Аналіз нових ліків на стан здоров’я людини	Покращеня якості ліків

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

а) визначення переліку техніко-економічних властивостей та характеристик ідеї;

б) визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

в) проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (табл. 4.2).

Таблиця 4.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	**Техніко-економічні характеристики ідеї**	**(потенційні) товари/концепції конкурентів**		**W (слабка сторона)**	**N (нейтральна сторона)**	**S (сильна сторона)**
		Мій проект	**FORecast 4u**			
1.	Точність прогнозування	Застосування кращої моделі	Відсутнє прогнозування			+
2.	Ризики невірного прогнозу	Існують, через велику кількість факторів	Відсутні через відсутність прогноз у		+	

Продовження таблиці 4.2

3.	Доступність, зручність	Обмежені функції:6 побудова моделі і прогнозування	Власний інтерфейс		+	

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 4.3):

1) за якою технологією буде виготовлено товар згідно ідеї проекту?
2) чи існують такі технології, чи їх потрібно розробити/доробити?
3) чи доступні такі технології авторам проекту?

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Створення системи конструювання ліків з хімічних речовин	Використання мови програмування Python	Наявна	Доступна
		Pytorch	Наявна	Доступна
		Використання мови програмування HTML	Наявна	Доступна
		Docker	Наявна	Наявна
Обрана технологія реалізації ідеї проекту: мова програмування Python. Бібліотека Pytorch				

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як Python 3.4 та Pytorch через їх доступність та безкоштовність.

Визначення ринкових можливостей, які можна використати під час 77 ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

Таблиця 4.4 - Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	15000000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Ліцензії, регуляторні
5	Специфічні вимоги до стандартизації та сертифікації	+
6	Середня норма рентабельності в галузі (або по ринку), %	350

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект.

За результатами аналізу таблиці 4.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та сформовано орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Потреба в відокремленні сигналу від шуму побудови прогнозів.	Аналітики, аналітичні відділи, лікарі	Велика кількість даних	Простота використання, висока точність
2.	Створення якісного прогнозу	Користувачі додатку	Цікавить простота у використанні, низька ціна підтримки системи	Швидкість обробки, низька ціна

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6, 4.7).

Таблиця 4.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок продуктів з кращими характеристиками	Передбачити додаткові переваги власного програмного продукту (ПП) для того, щоб повідомити про них саме після виходу на ринок конкурентів. Вдосконалення технічних моментів власного продукту. Обрати нову цільову аудиторію і зосередитися на ній: зниження цін.
2	Зміна потреб користувачів	Користувачам необхідне програмне забезпечення з іншим функціоналом	Передбачити можливість додавання нового функціоналу до створюваного ПП

Таблиця 4.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Конкуренція	Відсутність аналогічного продукту для вітчизняного користувача.	Адаптація програмного продукту до вітчизняних особливостей.

Продовження таблиці 4.7

2	Поява нових методів прогнозування	З’являться нові методи, що будуть швидше та ефективніше прогнозувати показники	Покращити ПП додаванням нового функціоналу, розширення можливостей
3	Поява нових методів моделювання	З’являться нові методи, що будуть швидше, та більш точно моделювати процеси	Покращити ПП додаванням нового функціоналу, розширення можливостей

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 4.8).

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
1. Вказати тип конкуренції - монополія	На ринку присутні декілька компаній-конкурентів, але їх товар дещо відрізняється між собою.	Підтримка якості продукту та постійні нововведення, вдосконалення.

Продовження таблиці 4.8

2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу ПП таким чином, щоб можна було легко переробити даний ПП для використання у галузях інших країн.
3. За галузевою ознакою - міжгалузева	Продукт може використовуватись для різних галузей	Постійне вдосконалення продукту, що не має прив'язки до сфери
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між видами ПП, їх особливостями.	Створити ПП, враховуючи недоліки конкурентів
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПП, щоб собівартість була нижчою	Удосконалення моделі. Використання більш дешевих технологій для розробки, ніж використовують конкуренти, але тільки якщо ці технології відповідають необхідним вимогам якості.
6. За інтенсивністю - не марочна	Бренд присутній, але його роль незначна	Реклама, участь у конференціях, семінарах.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 4.9).

Таблиця 4.9 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	**Прямі конкуренти в галузі**	**Потенційні конкуренти**	**Постачальники**	**Клієнти**	**Товари-замінники**
	Навести перелік прямих конкурентів	*Визначити бар'єри входження в ринок*	*Визначити фактори сили постачальників*	*Визначити фактори сили споживачів*	*Фактори загроз з боку замінників*
	SAS Matlab	Наявність вже існуючих рішень	-	Контроль якості продукту	Наявність більш широкого функціоналу, зручнішого інтерфейсу та авторитет (перевірена якість)
Висновки:	Досить інтенсивна конкурентна боротьба з вже закріпившимися на ринку гравцями	Є можливості виходу на ринок, але є і конкуренти. Строки – 18 місяців.	-	Клієнти диктують умови роботи на ринку: зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для побудови моделей і прогнозів.	Необхідно випускати ПЗ не гірше, ніж у конкурентів та розширяти функціонал.

За результатами аналізу було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок

щодо характеристик, які повинен мати проект, щоб бути конкурентоспроможним на ринку.

Цей висновок був врахований при формулюванні переліку факторів конкурентоспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в таблиці, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. 4.6, 4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлено у (табл. 4.10).

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	**Фактор конкурентоспроможності**	**Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)**
1	Багатофункціональність	Жоден конкурент не є настільки багатункціональним, не здатен на прогноз, чистку шумів та класифікацію сигналів
2	Якість	Можливість використання системи без складних імплементація в існуючу екосистему автомобіля
3	Висока якість прогнозу, велика кількість допоміжних статистичних даних	Робота з клієнтами – великими компаніями та окремими спеціалістами

За визначеними факторами конкурентоспроможності (табл. 4.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 - Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	**Фактор конкурентоспроможності**	**Бали 1-20**	**Рейтинг товарів-конкурентів у порівнянні з … (назва підприємства)**						
			–3	–2	–1	0	+1	+2	+3
1	Якість аналізу	15					*		
2	Простота використання	20			*				
3	Орієнтованість на кінцевого споживача	7					*		

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 4.11). Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 4.12 - SWOT-аналіз стартап-проекту

Сильні сторони:	**Слабкі сторони:**
Точність аналізу Простота використання Автономність	Потрібен час для навчання системи Інтерфейс користувача

Продовження таблиці 4.12

Можливості:	**Загрози:**
Аналіз емоцій людини на основі біосигналів Застосування системи для попередження серцевих нападів	Конкуренція

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 4.13).

Таблиця 4.13 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	**Альтернатива (орієнтовний комплекс заходів) ринкової поведінки**	**Ймовірність отримання ресурсів**	**Строки реалізації**
1	Ліцензування алгоритму і патентів	85%	6 місяців
2	Створення ПП з подальшим з подальшою інтеграцією в систему автомобіля	75%	18 місяців
3	Створення окремого додатку і девайса, який буде аксесуаром для керма	55%	16 місяців

Після аналізу було обрано альтернативу №1.

4.3 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 4.14).

Таблиця 4.14 - Вибір цільових груп потенційних споживачів

№ п/п	**Опис профілю цільової групи потенційних клієнтів**	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит в межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**
1	Компанії (українські та міжнародні) діяльність яких пов’язана з фінансово-економічною сферами.	Висока	Високий	Сильна	Складно
2	Приватні підприємства міського та міжнародного рівня, діяльність яких повязана з фінансово-економічною сферами.	Висока	Високий	Сильна	Складно

Продовження таблиці 4.14

3	Приватні підприємства, обласного рівня.	Помірна	Помірний	Помірна	Середня складність
4	Підприємства регіонального характеру	Помірна	Слабкий	Слабка	Просто
5	ФОП, які діють у фінансово-економічній сферах.	Слабка	Слабкий	Слабка	Просто
Які цільові групи обрано:1,2,3					

За результатами аналізу потенційних груп споживачів було обрано цільові групу, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 4.15).

Таблиця 4.15 - Визначення базової стратегії розвитку

№ п/п	**Обрана альтернатива розвитку проекту**	**Стратегія охоплення ринку**	**Ключові конкурентоспроможні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку***
1		Визначити потреби кожної з груп, розробити відповідно до них стратегії приваблення клієнтів та маркетингової комунікації	Цінова політика, універсальність продукту (миттєве практичне застосування), орієнтованість на кінцевого споживача	Стратегія диференціації

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 4.16).

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

№ п/п	**Чи є проект «першопрохідцем» на ринку?**	**Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?**	**Чи буде компанія копіювати основні характеристики товару конкурента, і які?**	**Стратегія конкурентної поведінки***
1	«Першопроходець»	Шукати нових	Ні	Стратегія заняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (таблиця 4.16) розроблено стратегію позиціонування (таблиця 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Легкість розуміння, зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для побудови моделей і прогнозів.	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування; Практичність та точність результату

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап- компанії на ринку.

4.4 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього підсумовано результати попереднього аналізу конкурентоспроможності товару (таблиця 4.18). Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Швидкість отримання результату	Швидка побудова моделі та створення прогнозу	Відсутність необхідності звертатися до сторонньої особи/компанії для побудови моделі та прогнозу. Дані компанії-користувача, якими оперує ПП, не передаються третім особам, чого вимагає політика безпеки багатьох компаній.
2	Зручність застосування	Не потрібно мати глибоких знань, для того щоб побудувати модель та спрогнозувати показники	ПП сам обирає необхідний та оптимальний метод для побудови моделі та прогнозу. Не потрібно мати глибоких знань у прогнозуванні для того, щоб користуватися ПП
3	Практичність та точність результату	Користувач отримує точні (з малою похибкою розбіжності) результати.	Користувач на виході роботи ПП отримує модель та прогноз, котрі відповідають необхідним показникам достовірності та точності. Отриманий прогноз можна використовувати для створення стратегії розвитку підприємства.

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 4.19).

1-й рівень При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 4.19 - Опис трьох рівнів моделі товару

Рівні товару	**Сутність та складові**		
I. Товар за задумом	Зручність та швидкість отримання практичного результату щодо побудови моделі та прогнозування процесів.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. функція побудови моделі процесу 2. функція побудови прогнозу		
	Якість: достовірність побудови математичної моделі, достовірність побудови прогнозу		
	Пакування: відсутнє		
	Марка: StatLabs «Forec»		
III. Товар із підкріпленням	До продажу: відсутнє		
	Після продажу: персональна підтримка в обслуговуванні за додаткову платню.		
Вихідний код та математична модель будуть закриті. На ідею зареєстровано патент.			

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання за допомогою ноу-хау. Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни

відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 4.20). Аналіз проведено експертним методом.

Таблиця 4.20 - Визначення меж встановлення ціни

№ п/п	**Рівень цін на товари-замінники**	**Рівень цін на товари-аналоги**	**Рівень доходів цільової групи споживачів**	**Верхня та нижня межі встановлення ціни на товар/послугу**
1	1800$	3500$	У всіх трьох груп високий рівень доходів	Базова покупка 1000$ Подальша персональна підтримка в обслуговуванні 150$/місяць

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 4.21).

Таблиця 4.21 - Формування системи збуту

№ п/п	**Специфіка закупівельної поведінки цільових клієнтів**	**Функції збуту, які має виконувати постачальник товару**	**Глибина каналу збуту**	**Оптимальна система збуту**
1	Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть отримати вигоду та	Встановлення контактів із споживачами і підтримання їх. Формування попиту і стимулювання збуту.	Один (від виробника одразу	Прямий канал збуту до споживача, мінімізувати збутові

Продовження таблиці 4.21

	покращити дохідність. Вони цікавляться сучасними розробками та інноваційними рішеннями, тому відвідують конференції, інтернет-конференції, семінари.	Дослідницька робота зі збору маркетингової інформації. Доробка товару, виходячи з потреб конкретного покупця.	спожива чу)	витрати розвиток маркетинго вого спілкування із споживачем

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 4.22).

Таблиця 4.22 - Концепція маркетингових комунікацій

№ п/п	**Специфіка поведінки цільових клієнтів**	**Канали комунікацій, якими користуються цільові клієнти**	**Ключові позиції, обрані для позиціонування**	**Завдання рекламного повідомлення**	**Концепція рекламного звернення**
	Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть отримати	Конференції, інтернет-конференції, семінари, огляд професійної літератури,	Позиція на основі порівняння фірми з товарами конкурентів;	- Створення репутації фірмі — виробнику чи посереднику; - збільшення чистого прибутку та	Шукаєте вірний шлях для розвитку вашої компанії? Досить даремно

Продовження таблиці 4.22

	вигоду та покращити дохідність. Вони цікавляться сучасними розробками та інноваційними рішеннями, тому відвідують конференції, інтернет-конференції, семінари.	інтернет, періодичні видання у різноманітних (профільних) галузях.	Відмінні особливості споживача	рентабельності фірми; - збільшення потоків покупців та обсягів продажу; стабілізація обсягів продажу в період зменшення попиту та загального спаду ділової активності.	гаяти час на вгадування вірної стратегії! Користуйтесь «Forec» і світле майбутнє вашій компанії забезпечено!

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

4.5 Висновки до четвертого розділу

В даному розділі було проведено аналіз програмного продукту у якості стартап-проекту. Можна зазначити, що у проекту є можливість

комерціалізації, оскільки ринок потребує якісний продукт, що надає автоматичну побудову лікарських речовин.

На ринку наявна монополістична конкуренція, існує декілька фірм-конкурентів, але їх товар дещо відрізняється, тому вихід на ринок не буде легким і потребує грамотної стратегії виходу. Для впровадження ринкової реалізації проекту слід обрати альтернативу, яка передбачає розробку програмного продукту з подальшим розповсюдженням ліцензій та права на використання за певну роялті.

ВИСНОВКИ ПО РОБОТІ ТА РЕКОМЕНДАЦІЇ ЩОДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У даній роботі було розглянуто дані по експресії генів та клітин та побудовано модель нейронної мережі ж критерієм оптимізації Adam для прогнозування механізмом дії клітин (MoA).

У першому розділі було розглянуто різні підходи очистки даних для подальшого покращення якості тренування моделей. А саме способи очистки від викидів, рішення проблеми сильної кореляції між змінними, а також різні методики нормалізації та стандартизації даних.

У другому розділі було розглянуто весь етап побудови і навчання нейронної мережі з критерієм оптимізації Adam і способи перевірки точності моделі.

У третьому розділі був аналіз даних та пошук найкращих гіперпараметрів для моделі. З основних проблем даних була сильна незбалансованість цільових класів, висока кореляція багатьох змінних як генів, так і клітин, з чи нам допоміг метод головних компонент (PCA). З допомогою, якого нам значно вдалося зменшити розмірність матриці для тренування, зберігши 95% інформативності початкових даних. Нам пощастило, що в даних не було пропущених значень. Точність моделі становить 0.016, що розраховувалася за перехресною ентропією на тестових даних.

Дане дослідження зможе допомогти сконструювати білкову мішень, пов'язану з хворобою, і розробити молекулу, здатну модулювати цю білкову мішень. У майбутніх дослідженнях планується спробувати навчання на рекурентних нейронних мережах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. Москва: Мир, 1992. 184 с.
2. Michael A. Nielsen. Neural Networks and Deep Learning. URL: http://neuralnetworksanddeeplearning.com
3. Mechanisms of Action (MoA) Prediction. URL: https://www.kaggle.com/c/lish-moa/overview.
4. Pytorch documentation. URL: https://pytorch.org/docs/stable/index.html.
5. Adam Optimization Algorithm. URL: https://towardsdatascience.com/adam-optimization-algorithm-1cdc9b12724a.
6. How to Remove Outliers for Machine Learning. URL: https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data.
7. Principal component analysis: a review and recent developments. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202#d3e3131.
8. Хайкин С. Нейронные сети. Полный курс / под ред. Хайкина С. Москва: Вильямс, 2006. 1104 с.
9. Hertz J., Krogh A. and Palmer R.G. Introduction to the Theory of Neural Computation. London: Mass, 1991. 111 p.
10. Каллан Р. Основные концепции нейронных сетей / за ред. Роберта Каллана. Москва: Вильямс, 2001. 287 с.
11. Rosenblatt R. Principles of Neurodynamic. New York: Spartan Books, 1962. 230 p.
12. Круглов В.В. Искусственные нейронные сети. Теория и пратика / под ред. Круглова В.В., Борисова В.В. Москва: Телеком, 2002. 382 с.
13. How the backpropagation algorithm works. URL: http://neuralnetworksanddeeplearning.com/chap2.html.

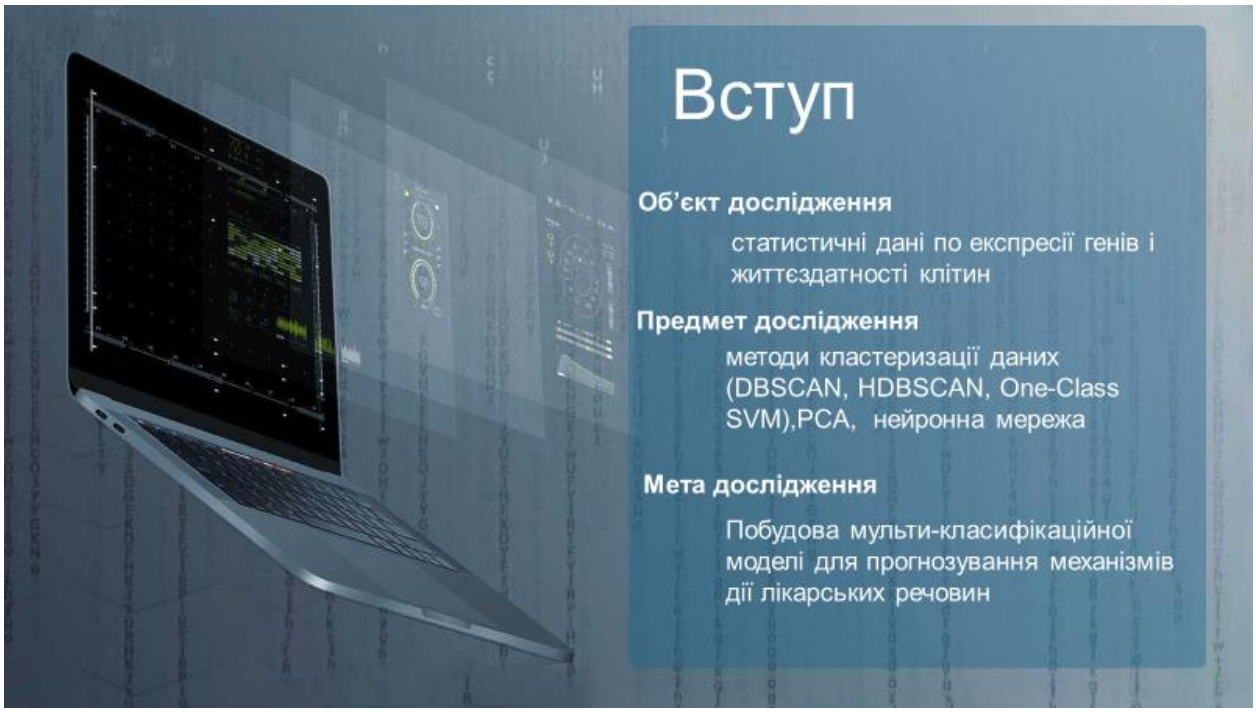
14. A Step by Step Backpropagation Example. URL: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/.

15. Zaccone G. Deep Learning with TensorFlow. Birmingham: Packt Publishing, 2017. 300p.

16. Raschka S. Python Machine Learning. Birmingham: Packt Publishing, 2005. 456 p

17. Locascio N. Fundamentals of Deep Learning. Sebastopol: O'Reilly Media Inc, 2017. 298 p.

ДОДАТОК А. Презентаційні матеріали




Постановка задачі
01
Огляд і аналіз існуючих методик очистки даних та моделей мульти-класифікації
02
Провести дослідження даних по експресії генів і життєздатності клітин
03
Підготовка даних до навчання моделі та її побудова
04
Проаналізувати отримані результати


Актуальність роботи
A
Просування розробки ліків за рахунок поліпшення алгоритмів прогнозування MoA
B
Представлення результатів аналізу для подальшого дослідження алгоритмів прогнозування MoA

Набір даних

Дані

Цей набір даних містить інформацію про експресії генів і життєздатності клітин. Джерелом даних є The Connectivity Map, проект в рамках Broad інституту MIT і Гарварду https://clue.io/

875 змінних

Кожен запис містить:

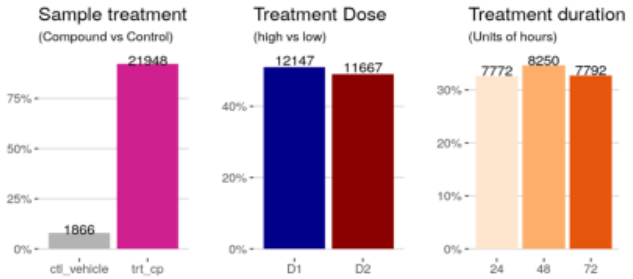
- дані про експресію гена (772 змінних)
- дані про життєздатність клітин (100 змінних)
- Тип обробки молекули: сполукою чи з контрольним збуренням
- тривалість лікування
- дозування лікування

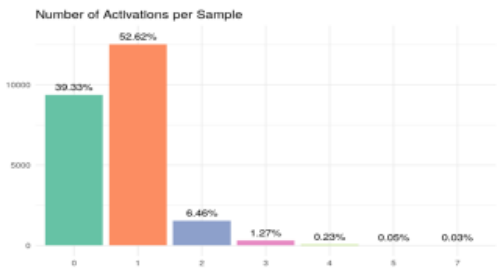
~25тис. записів

207 вихідних змінних

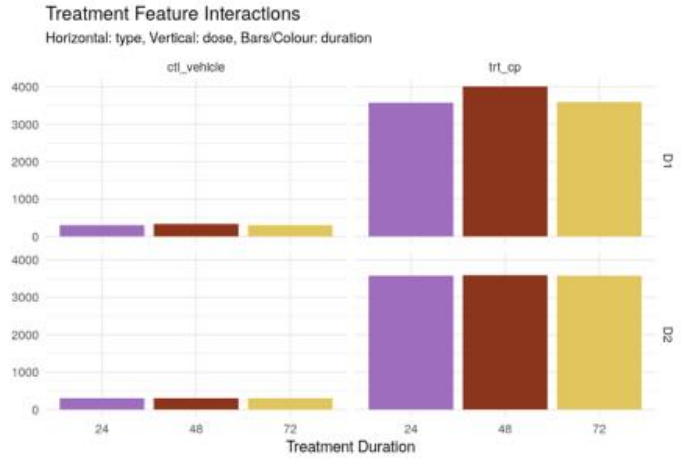
Двійкові цільові показники MoA

Аналіз індивідуальних особливостей





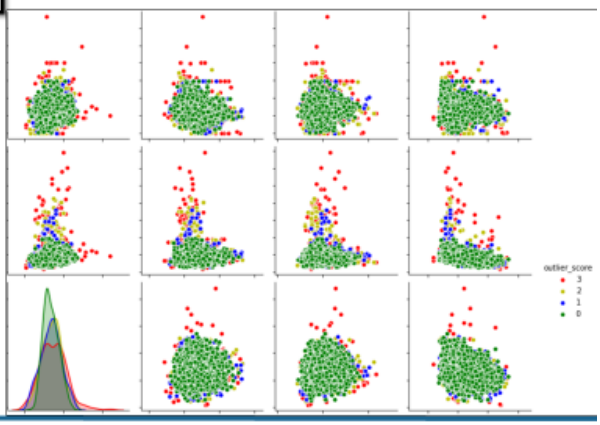
Аналіз багатофункціональної взаємодії



Пошук відхилених значень з допомогою методів кластеризації

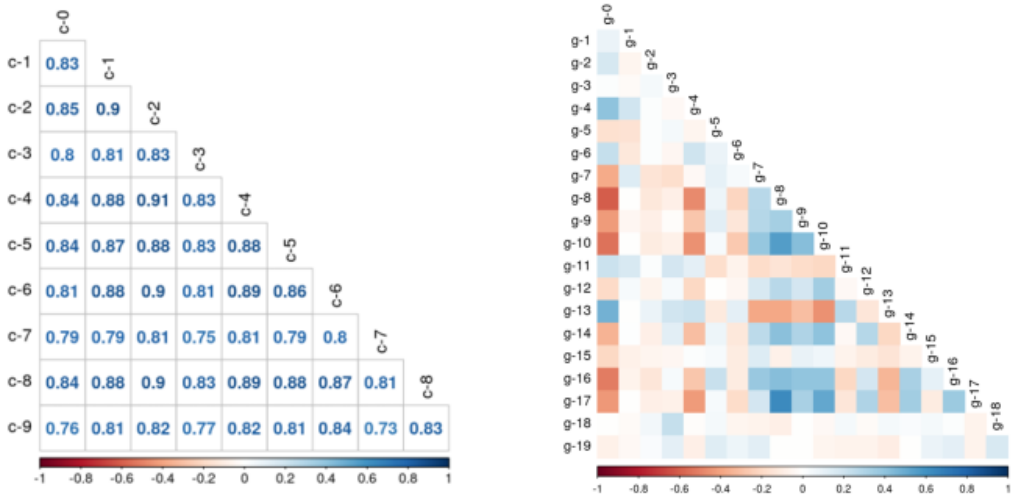
- Однокласовий SVM, DBSCAN, HDBSCAN
- За поріг було взято не більше 10% відхилень при використанні кожного методу окремо

0	0.762272
1	0.148701
3	0.040807
2	0.036265

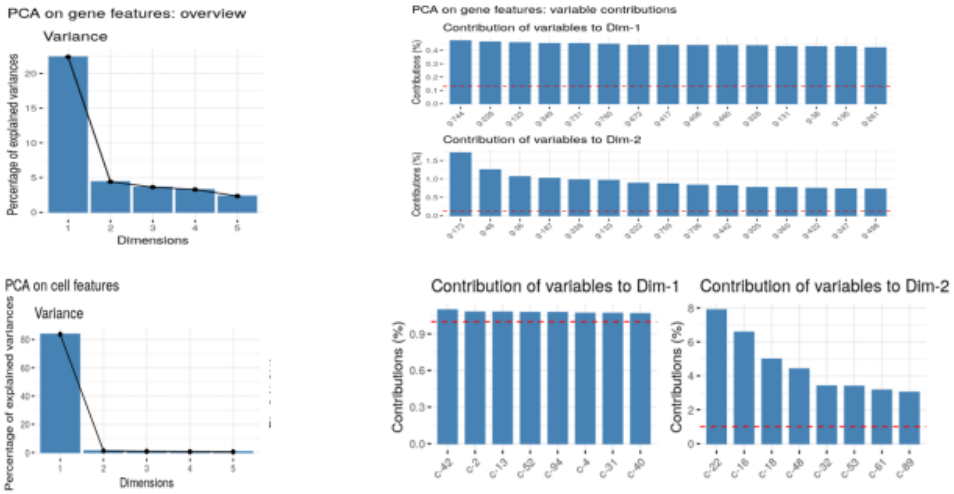


Було вирішено відкинути значення з трьома методами, що вирішили що ці значення є відхиленнями. Внаслідок чого, було втрачено 4% даних.

Особливості життєздатності клітин та експресії генів



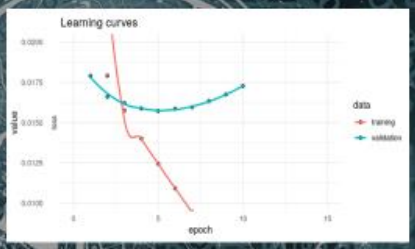
Зменшення розмірності за допомогою PCA



Нейронна мережа

- Вхідний шар нормілизований за StandartScaler
- 4 прихованих шари з активаційною функціями RELU)
- Вихідний шар з Sigmod функцією
- Критерій оптимізації Adam (розмір кроку навчання – 10^{-4})
- функція втрат – бінарна кросентропія

Залежність похибок від епохи навчання



Поділ даних

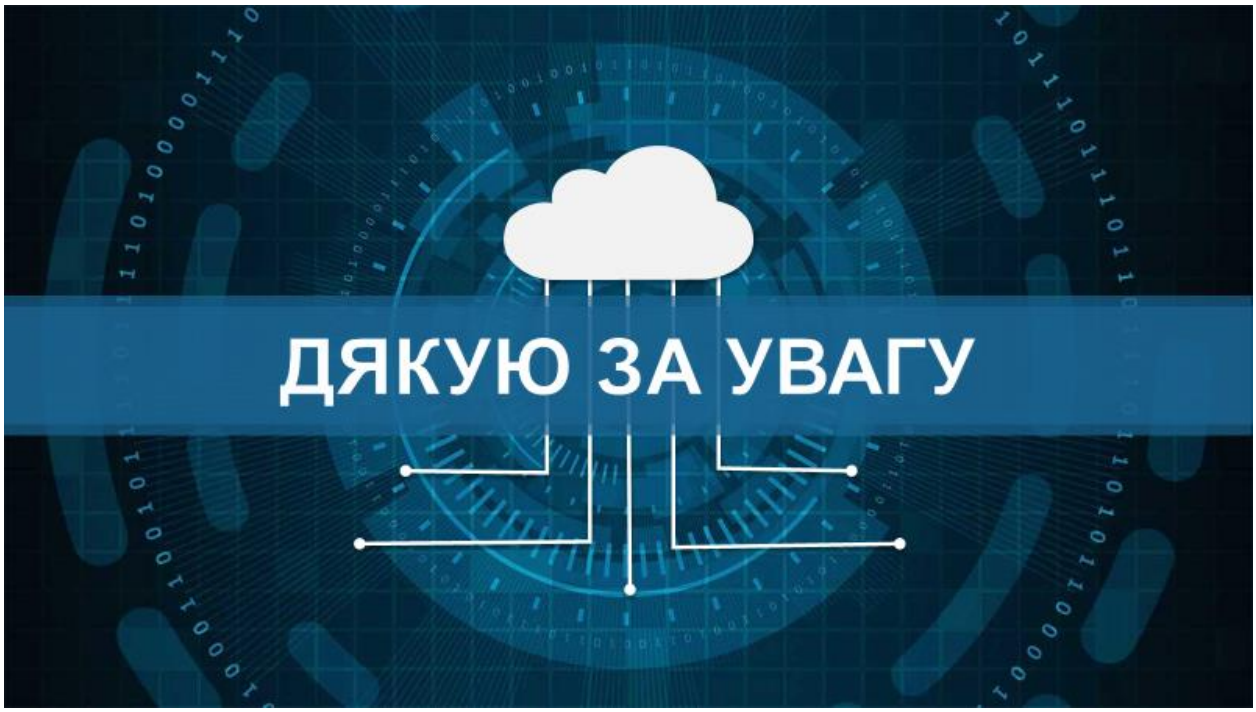
- навчальна – 80%
- тестова – 20%

оцінка моделі на тестових даних:
0.016

Висновки

- Було розглянуто різні підходи очистки даних для подальшого покращення якості тренування моделей. А саме способи очистки від викидів, рішення проблеми сильної кореляції між змінними, а також різні методики нормалізації та стандартизації даних
- Розглянуто весь етап побудови і навчання нейронної мережі з критерієм оптимізації Adam і способи перевірки точності моделі
- Розроблено програмний продукт для прогнозування механізмів дії лікарських речовин

Подальші дослідження



ДОДАТОК Б. Лістинг

```
import seaborn as sns

import warnings

import plotly.graph_objs as go

from plotly.offline import init_notebook_mode, iplot

from plotly import tools

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import OneHotEncoder




# In[3]:




warnings.filterwarnings("ignore")




# In[4]:




import pandas as pd



# data load
```

```
features_train_df = pd.read_csv('./data/train_features.csv')

features_test_df = pd.read_csv('./data/test_features.csv')



train_targets_scored_df = pd.read_csv('./data/train_targets_scored.csv')



features_train_df.head()




# In[5]:




train_targets_scored_df.head()




# In[6]:




features_train_df[features_train_df.isin(['unknown', '?', ''])] = np.nan




# In[7]:
```

```
#Check missing values



Missing_df = pd.DataFrame(features_train_df.isnull().sum() / features_train_df.count())



# rename and sort by column

Missing_df.columns = ['missing_ratio']

Missing_df = Missing_df.sort_values(by=['missing_ratio'], ascending=False)



Missing_df[Missing_df.missing_ratio > 0]




# In[ ]:




from pandas_profiling import ProfileReport

profile_features = ProfileReport(features_train_df, title='Features_train Report',minimal=True,
html={'style':{'full_width':True}})

profile_features.to_file(output_file="features.html")




# In[19]:




profile_targets = ProfileReport(train_targets_scored_df, title='Targets_train Report',
html={'style':{'full_width':True}})
```

```
profile_targets.to_file(output_file="targets.html")




# In[ ]:




# train_targets_nonscored_df = pd.read_csv('./data/train_targets_nonscored.csv')

# train_targets_nonscored_df.head()




# ## Data Preprocessing



# ### Outlier finding



# In[10]:




data_features = features_train_df.copy()

data_features.drop(columns=['sig_id', 'cp_type', 'cp_time', 'cp_dose'], inplace=True)




# In[12]:
```

```
scaler = StandardScaler()
scaled_data = pd.DataFrame(
    data=scaler.fit_transform(data_features),
    columns=data_features.columns
)



# In[8]:



def anomalies_report(outliers):

    print("Total number of outliers: {}\nPercentage of outliers:   {:.2f}%".format(
             sum(outliers), 100*sum(outliers)/len(outliers)))



# #### DBSCAN

# In[14]:



from sklearn.cluster import DBSCAN

outlier_percentage = 1.
```

```
num_clusters = []

anomaly_percentage = []



eps = 30

eps_history = [eps]

while outlier_percentage > 0.1:

    model = DBSCAN(eps=eps, n_jobs=-1).fit(scaled_data)

    labels = model.labels_

    num_clusters.append(len(np.unique(labels)) - 1)

    labels = np.array([1 if label == -1 else 0 for label in labels])

    outlier_percentage = sum(labels==1) / len(labels)

    eps_history.append(eps)

    anomaly_percentage.append(outlier_percentage)

    print(outlier_percentage, eps)

    eps += 1







# In[21]:



model = DBSCAN(30, n_jobs=-1)

model.fit(scaled_data)
```

```
density_outlier = np.array([1 if label == -1 else 0 for label in model.labels_])




# #### HDBSCAN


# In[13]:




from hdbscan import HDBSCAN


clusterer = HDBSCAN(min_cluster_size=50).fit(scaled_data)


threshold = pd.Series(clusterer.outlier_scores_).quantile(0.9)

hdb_outliers = np.array([1 if i > threshold else 0 for i in clusterer.outlier_scores_])




# #### SVM


# In[15]:




from sklearn.svm import OneClassSVM


one_class_svm = OneClassSVM(nu=0.1, gamma='auto')
```

```
one_class_svm.fit(scaled_data)

svm_outliers = one_class_svm.predict(scaled_data)

svm_outliers = np.array([1 if label == -1 else 0 for label in svm_outliers])




# #### IsolationForest



# In[16]:




from sklearn.ensemble import IsolationForest



isolation_forest = IsolationForest(n_estimators=100, contamination=0.1,

                                   max_features=1.0, bootstrap=True, behaviour="new")

isolation_forest.fit(scaled_data)



isolation_outliers = isolation_forest.predict(scaled_data)

isolation_outliers = np.array([1 if label == -1 else 0 for label in isolation_outliers])




# In[18]:




anomalies_report(isolation_outliers)
```

```
# In[22]:



summary = np.concatenate((
    [density_outlier],
    [hdb_outliers],
    [svm_outliers],
    [isolation_outliers]
))

summary = pd.DataFrame(
    summary.T,
    columns=['dbscan', 'hdbscan', 'svm', 'isolation']
)
summary.head()



# In[23]:



outliers_score_model_based = summary.sum(axis=1)
```

```
# In[24]:



outliers_score_model_based.value_counts()/ len(outliers_score_model_based)



# In[28]:



features_train_df



# In[31]:



features_train_df



# In[33]:



features_train_df.reset_index(inplace=True)

features_train_df['outliers_score'] = outliers_score_model_based
```

```
features_train_df = features_train_df.loc[lambda x: x.outliers_score !=3].reset_index()

features_train_df.drop(columns=['outliers_score', 'level_0', 'index'], inplace=True)

features_train_df.head()




# In[35]:




features_train_df.to_csv('features_train_outliers_filt.csv', index=False)




# In[9]:




features_train_df = pd.read_csv('features_train_outliers_filt.csv')

features_train_df.head()




# ### Corelation

# In[10]:




col_ignore = ['sig_id', 'cp_type', 'cp_time', 'cp_dose']
```

```
corr_matrix = features_train_df.corr()

lower = corr_matrix.where(np.tril(np.ones(corr_matrix.shape), k=-1).astype(np.bool))



high_corr = [

    column for column in lower.columns if any((lower[column] > 0.6)|(lower[column] < -0.6))

]



other_features = [col for col in features_train_df.columns if col not in high_corr and col not
in col_ignore]




# In[35]:




features_train_df.iloc[:, 776:]




# In[41]:




def corr_plot(df):

    '''

    Plot correlation matrix for dataframe

    '''
```

```
    f = plt.figure(figsize=(19, 15))
    plt.matshow(df.corr(), fignum=f.number)
#     plt.xticks(range(df.shape[1]), df.columns, fontsize=14, rotation=45)
#     plt.yticks(range(df.shape[1]), df.columns, fontsize=14)
    cb = plt.colorbar()
    cb.ax.tick_params(labelsize=14)
    plt.title('Correlation Matrix', fontsize=16)
    plt.show()



# In[42]:



'''
`g` features
'''
corr_plot(features_train_df.iloc[:, 4:776])



# In[43]:



'''
`c` features
```

```
'''

corr_plot(features_train_df.iloc[:, 776:])




# In[ ]:









# In[64]:



features_train_df.cp_type.value_counts()



# Balance `cp_type` in train data

# In[74]:



cp_type_trt_cp = features_train_df.loc[lambda x: x.cp_type == 'trt_cp']

cp_type_ctl_vehicle = features_train_df.loc[lambda x: x.cp_type == 'ctl_vehicle']
```

```
# In[75]:



diff_int = round(cp_type_trt_cp.shape[0] / cp_type_ctl_vehicle.shape[0])

diff_float = round(cp_type_trt_cp.shape[0] / cp_type_ctl_vehicle.shape[0], 1) - diff_int



# In[76]:



features_train_df_scaled = cp_type_trt_cp.append([cp_type_ctl_vehicle]*diff_int,
ignore_index=True)

features_train_df_scaled =
features_train_df_scaled.append(cp_type_ctl_vehicle[:int(cp_type_ctl_vehicle.shape[0]*diff_flo
at)],

                          ignore_index=True)



features_train_df_scaled.cp_type.value_counts()



# ### Encoding



# In[80]:
```

```
pd.get_dummies(features_train_df_scaled[['cp_type', 'cp_dose']], prefix_sep='_')



# In[81]:



'''

Encoding categorical data

'''



features_train_df_onehot  =  pd.get_dummies(features_train_df_scaled[['cp_type',  'cp_dose']],
prefix_sep='_')

features_train_df_scaled = features_train_df_scaled.join(features_train_df_onehot)

features_train_df_scaled.drop(columns=['cp_type', 'cp_dose'], inplace=True)



# In[82]:



'''

Encoding numerical data

'''



std_scaler = StandardScaler()
```

```
features_train_df_scaled.iloc[:,                                1:-4]                                =
std_scaler.fit_transform(features_train_df_scaled.iloc[:, 1:-4])

features_train_df_scaled.head()
```

```
# In[84]:
```

```
features_train_df_scaled.to_csv('features_train_df_scaled_before_pca.csv', index=False)
```

```
# ### PCA
```

```
# In[88]:
```

```
features_train_df_scaled.iloc[:, 2:774].columns
```

```
# In[94]:
```

```
features_train_df_scaled.iloc[:, 774:874].columns
```

```
# In[95]:



from sklearn.decomposition import PCA


pca_g = PCA(0.95)

pca_c = PCA(0.95)


g = pca_g.fit_transform(features_train_df_scaled.iloc[:, 2:774])

for i in range(g.shape[1]):

    features_train_df_scaled[f'g_pca_{i}'] = g[:,i]


c = pca_c.fit_transform(features_train_df_scaled.iloc[:, 774:874])

for i in range(c.shape[1]):

    features_train_df_scaled[f'c_pca_{i}'] = c[:,i]

features_train_df_scaled.drop(columns=list(features_train_df_scaled.iloc[:, 2:874].columns),

                      inplace=True)



# In[97]:



features_train_df_scaled.head()
```

```
# In[98]:



features_train_df_scaled.to_csv('features_train_df_scaled_after_pca.csv', index=False)



# In[102]:



features_train_df_scaled



# ### Targets

# In[104]:



train_df = pd.merge(features_train_df_scaled,
        train_targets_scored_df,
        'left',
        on=['sig_id'])
```

```
# In[105]:



train_df.to_csv('./data/train.csv', index=False)
```