

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Звіт до лабораторної роботи №3

Студента групи ТТП-41
Маркова Максима Юрійовича

викладач: Панченко Тарас Володимирович

Анотація

У цій роботі були розроблені та застосовані моделі для аналізу та прогнозування параметрів дорожніх робіт на основі різноманітних даних. Основна мета роботи полягала в розробці моделей для передбачення серйозності дорожніх робіт, тривалості їх виконання, а також дистанції, на якій проводяться ці роботи, за допомогою машинного навчання.

Основні етапи роботи:

- Розробка моделей для передбачення ступеня серйозності дорожніх робіт за допомогою алгоритмів Logistic Regression, Random Forest Classifier та XGB Classifier. Моделі працюють на масштабованому датафреймі, підготовленому за допомогою StandardScaler().
- Кластеризація даних для аналізу залежностей між параметрами, такими як дистанція, температура, швидкість вітру та вологість повітря. Для цього використовувався алгоритм K-Means.
- Розробка моделей для прогнозування тривалості та дистанції дорожніх робіт із застосуванням Linear Regression, Random Forest Regressor та XGB Regressor.
- Пошук асоціативних правил між числовими параметрами та параметрами точок інтересу (POI) за допомогою алгоритму Apriori, для чого була використана бінарна матриця для введення даних.

Для реалізації були обрані такі методи та інструменти:

- Logistic Regression, Random Forest Classifier, XGB Classifier для класифікаційних задач.
- K-Means для кластеризації даних та виявлення закономірностей.
- Linear Regression, Random Forest Regressor, XGB Regressor для побудови регресійних моделей.
- Алгоритм Apriori для пошуку асоціативних правил між числовими та категоріальними параметрами.

Отримані результати можуть бути використані для оптимізації планування дорожніх робіт, прогнозування їх тривалості та серйозності, а також для

подальшого аналізу поведінки різних параметрів, що впливають на ефективність цих робіт.

Вступ

Тема роботи: «Аналіз дорожніх робіт на основі даних».

Мета роботи – розробити моделі для прогнозування та класифікації параметрів дорожніх робіт, таких як ступінь серйозності, тривалість та дистанція робіт, з використанням сучасних методів машинного навчання. Робота також включає пошук асоціативних правил між параметрами, що характеризують дорожні роботи, для подальшого аналізу та оптимізації процесів.

Лабораторна робота складається з наступних етапів:

1. Попередня обробка даних
 - a. Збір та підготовка даних, включаючи масштабування та трансформацію числових параметрів.
 - b. Використання StandardScaler для масштабування даних, що забезпечує коректне навчання моделей.
2. Розробка моделей для передбачення ступеня серйозності дорожніх робіт
 - a. Використання логістичної регресії, Random Forest Classifier та XGB Classifier для класифікації дорожніх робіт за серйозністю.
 - b. Оцінка точності моделей за допомогою метрик, таких як точність, точність, recall та F1-score.
3. Кластеризація даних
 - a. Застосування алгоритму K-Means для кластеризації даних і виявлення залежностей між параметрами, такими як дистанція, температура, швидкість вітру та вологість повітря.
 - b. Вивчення кластерів, які можуть вказувати на різні типи дорожніх робіт та умови їх виконання.
4. Прогнозування тривалості та дистанції дорожніх робіт
 - a. Розробка моделей регресії для прогнозування тривалості та дистанції робіт, використовуючи Linear Regression, Random Forest Regressor та XGB Regressor.
 - b. Оцінка ефективності моделей на основі метрик, таких як

середнє абсолютне відхилення (MAE) та корінь середньоквадратичної помилки (RMSE).

5. Пошук асоціативних правил

- a. Використання алгоритму Apriori для виявлення частих наборів даних між різними параметрами дорожніх робіт.
- b. Формулювання асоціативних правил між числовими параметрами, такими як температура, швидкість вітру та вологість, а також параметрами точок інтересу (POI).
- c. Фільтрація отриманих правил для знаходження найбільш значущих залежностей, які можуть бути використані для оптимізації процесу планування дорожніх робіт.

6. Візуалізація та аналіз результатів

- a. Візуалізація результатів аналізу даних, побудова графіків для зображення асоціативних правил, кластерів та прогнозних моделей.
- b. Інтерпретація знайдених залежностей для подальшого використання в системах планування та оптимізації дорожніх робіт.

1. Теоретична частина

1.1 Алгоритм Apriori

Apriori — алгоритм глибинного аналізу даних щодо частих одиниць у множинах і машинного навчання щодо асоціативних правил, що застосовується переважно до баз даних транзакцій. Алгоритм ідентифікує елементи/одиниці, що часто повторюються у базі, і розширює їх список до все більших множин з дотриманням правила достатньої частотності. Визначені алгоритмом множини частих одиниць можна використати для визначення правил асоціювання, по яких стають помітними загальні тенденції в базі даних.

В задачі аналізу ринкового кошику одиницями є пропоновані товари, а покупка являє собою транзакцію, яка містить куплені предмети (одиниці). Алгоритм при цьому визначає кореляції такого виду: якщо хтось купує шампунь і лосьйон для гоління, у 90% випадків купується також і піна для гоління.

Дані, що надаються до аналізу, складаються з таблиці транзакцій (на рядках), в якій перераховуються будь-які бінарні одиниці (у колонках). Алгоритм Apriori знаходить співвідношення між множинами одиниць, які зустрічаються у великій частині транзакцій. Правила асоціювання, які отримуються в результаті, мають форму $A \rightarrow B$ при цьому A і B є множинами одиниць, а правило стверджує, що коли у великій частині транзакцій зустрічається множина одиниць A , то там часто зустрічається і множина одиниць B .

1.2 Опис алгоритму Apriori

Алгоритм Apriori було запропоновано Агравалом і Срікантом в 1994 році. Apriori застосовується до баз даних транзакцій (наприклад, наборів товарів, куплених клієнтами, або відвідуваності вебсайту). Кожна транзакція розглядається як множина елементів. Маючи заданий поріг C , алгоритм Apriori ідентифікує множину елементів, які є підмножинами принаймні C транзакцій в базі даних.

Апріорі використовує підхід «знизу вгору», за якого список частих підмножин розширюється по одному елементу за раз (крок, відомий як генерування кандидатів); відтак групи кандидатів перевіряються на основі

наявних даних. Алгоритм завершує роботу, коли подальших успішних розширень знайти неможливо.

Apriori використовує пошук у ширину та структуру геш-дерева для ефективного підрахунку елементів-кандидатів множини. Він генерує множини елементів-кандидатів довжиною k з множин довжиною $k-1$. Потім він відсікає кандидатів, які є нечастими підмножинами. Відповідно до леми низхідного змикання (англ. downward closure lemma), множина кандидатів містить усі множини елементів довжини k , які часто зустрічаються. Після цього він сканує базу даних транзакцій, щоб визначити множини елементів, які часто зустрічаються серед кандидатів.

Псевдокод для алгоритму наведено на рис. 1 для бази даних транзакцій T і допоміжного порога ϵ . Застосовується звичайна нотація теорії множин, хоча слід відзначити, що T є мультимножиною. C_k — це множина кандидатів для рівня k . З кожним кроком, як припускається, алгоритм генерує набори кандидатів з великих множин попереднього рівня, дотримуючись леми низхідного закриття. $\text{count}[c]$ звертається до поля структури даних, яка являє собою множину кандидатів c , спочатку вона приймається рівною нулю.

```

Apriori( $T, \epsilon$ )
   $L_1 \leftarrow \{\text{large 1-itemsets}\}$ 
   $k \leftarrow 2$ 
  while  $L_{k-1} \neq \emptyset$ 
     $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \not\subseteq L_{k-1}\}$ 
    for transactions  $t \in T$ 
       $C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$ 
      for candidates  $c \in C_t$ 
         $\text{count}[c] \leftarrow \text{count}[c] + 1$ 
     $L_k \leftarrow \{c \mid c \in C_k \wedge \text{count}[c] \geq \epsilon\}$ 
     $k \leftarrow k + 1$ 
  return  $\bigcup_k L_k$ 

```

Рисунок 1 - псевдокод для алгоритму Apriori

1.3 Обмеження алгоритму Apriori

Алгоритм Аpriori страждає від низької ефективності та компромісів, що призвело до виникнення інших алгоритмів. Утворення кандидата породжує велику кількість підмножин (алгоритм намагається завантажити в набір кандидатів якомога більше даних перед кожним скануванням). Пошук підмножин проходом від низу до верху (по суті обхід в ширину підмножини решітки) знаходить будь-яку максимальну підмножину S тільки після того, як будуть знайдено її $2^{|S|} - 1$ власних підмножин.

Алгоритм сканує базу даних дуже багато разів, що суттєво скорочує швидкодію. Тому, для того щоб алгоритм працював швидко, потрібно, щоб база постійно знаходилась у пам'яті.

Також часова та просторова складність алгоритму є дуже високими.

1.4 Основні метрики алгоритму Apriori

Support

Support елемента x - це не що інше, як відношення числа транзакцій з товаром x до загального числа транзакцій.

Confidence

Confidence ($x \Rightarrow y$) позначають можливість купівлі товару y при купівлі товару x . У цьому методі враховується популярність товару x .

Lift

Lift ($x \Rightarrow y$) — це «цікавість» або ймовірність покупки товару y при купівлі товару x . На відміну від confidence ($x \Rightarrow y$), в цьому методі враховується популярність товару y .

Якщо $\text{lift}(x \Rightarrow y) = 1$, то кореляції в наборі товарів немає.

Якщо $\text{lift}(x \Rightarrow y) > 1$, кореляція в наборі товарів позитивна, тобто ймовірність спільної покупки товарів x і y вище.

Якщо $\text{lift}(x \Rightarrow y) < 1$, кореляція в наборі товарів негативна, тобто сумісна покупка товарів x і y маловірогідна.

$$\begin{aligned}
 \text{Rule } X \Rightarrow Y & \begin{cases} \text{Support} = \frac{\text{Frequency}(X,Y)}{N} \\ \text{Confidence} = \frac{\text{Frequency}(X,Y)}{\text{Frequency}(X)} \\ \text{Lift} = \frac{\text{Support}}{\text{Support}(X) * \text{Support}(Y)} \end{cases}
 \end{aligned}$$

Рисунок 2 - формули Support, Lift та Confidence для асоціативного правила $X \Rightarrow Y$

1.5 One-hot encoding

One Hot Encoding — це метод перетворення категоріальних змінних у двійковий формат. Він створює нові стовпці для кожної категорії, де 1 означає, що категорія присутня, а 0 означає, що її немає. Основна мета One Hot Encoding — забезпечити ефективне використання категоріальних даних у моделях машинного навчання.








Car_model					
			Car_model_BMW	Car_model_Lexus	Car_model_Toyota
0	Toyota		0.0	0.0	1.0
1	Lexus		0.0	1.0	0.0
2	BMW		1.0	0.0	0.0
3	Toyota		0.0	0.0	1.0

Рисунок 3 - приклад One-hot encoding

1.6 Алгоритм K-Means

Алгоритм K-Means — популярний метод кластеризації, — впорядкування множини об'єктів у порівняно однорідні групи. Винайдений в 1950-х роках математиком Гуго Штайнгаузом і майже одночасно Стюартом Ллойдом. Особливу популярність отримав після виходу роботи МакКвіна (1967).

Мета методу — розділити n спостережень на k кластерів, так щоб кожне спостереження належало до кластера з найближчим до нього середнім

значенням. Метод базується на мінімізації суми квадратів відстаней між кожним спостереженням та центром його кластера, тобто функції $\sum_{i=1}^N d(x_i, m_j(x_i))^2$, де d — метрика, x_i — i -ий об'єкт даних, а $m_j(x_i)$ — центр кластера, якому на j -ій ітерації приписаний елемент x_i .

1.7 Опис алгоритму K-Means

Маємо масив спостережень (об'єктів), кожен з яких має певні значення за рядом ознак. Відповідно до цих значень об'єкт розташовується у багатовимірному просторі.

1. Дослідник визначає кількість кластерів k , що необхідно утворити
2. Випадковим чином обирається k спостережень, які на цьому кроці вважаються центрами кластерів
3. Кожне спостереження «приписується» до одного з k кластерів — того, відстань до якого найкоротша
4. Розраховується новий центр кожного кластера як елемент, ознаки якого розраховуються як середнє арифметичне ознак об'єктів, що входять у цей кластер
5. Відбувається така кількість ітерацій (повторюються кроки 3-4), поки кластерні центри стануть стійкими (тобто при кожній ітерації в кожен кластер потрапляють одні й ті самі об'єкти), дисперсія всередині кластера буде мінімізована, а між кластерами — максимізована.

Вибір кількості кластерів робиться на основі дослідницької гіпотези. Якщо її немає, то рекомендують спочатку створити 2 кластери, далі 3, 4, 5, порівнюючи отримані результати.

1.8 Переваги та недоліки алгоритму K-Means

Головні переваги методу k -середніх — його простота та швидкість виконання. Метод k -середніх більш зручний для кластеризації великої кількості спостережень, ніж метод ієрархічного кластерного аналізу (у якому дендограми стають перевантаженими і втрачають наочність).

Одним із недоліків простого методу є порушення умови зв'язності елементів одного кластера, тому розвиваються різні модифікації методу, а також його нечіткі аналоги, у яких на першій стадії алгоритму допускається приналежність одного елемента множини до декількох кластерів (із різним ступенем приналежності).

Попри очевидні переваги методу, він має суттєві недоліки:

- Результат класифікації сильно залежить від початкових позицій кластерних центрів
- Алгоритм чутливий до викидів, які можуть викривлювати середнє
- Кількість кластерів має бути заздалегідь визначена дослідником

1.9 Elbow Method

У кластеризації K-Means ми починаємо з випадкової ініціалізації k кластерів і ітеративного коригування цих кластерів, поки вони не стабілізуються в точці рівноваги. Однак перш ніж ми зможемо це зробити, нам потрібно вирішити, скільки кластерів (k) ми повинні використовувати.

Метод ліктя допомагає нам знайти це оптимальне значення k . Ось як це працює:

1. Ми “пробігаємо” діапазон значень k , як правило, від 1 до n (де n — параметр, який ми вибираємо).
2. Для кожного k ми обчислюємо суму квадратів у межах кластера (WCSS; Within-Cluster Sum of Squares).

WCSS вимірює, наскільки добре точки даних згруповані навколо відповідних центроїдів. Він визначається як сума квадратів відстаней між кожною точкою та її центроїдом кластера:

$$WCSS = \sum_{i=1}^k \sum_{j=1}^{n_i} d(x_j^{(i)}, c_i)^2, \text{ де } d(x_j^{(i)}, c_i)^2 \text{ представляє відстань між } j\text{-ю}$$

точкою даних $x_j^{(i)}$ у кластері i та центроїд c_i цього кластера.

Метод ліктя працює за наступною схемою:

- Ми обчислюємо міру відстані під назвою WCSS (сума квадратів у кластері). Це говорить нам про те, наскільки розподілені точки даних у кожному кластері.
- Ми пробуємо різні значення k (кількість кластерів). Для кожного k ми запускаємо K-Means і обчислюємо WCSS.
- Ми будуємо графік з k на осі X і WCSS на осі Y.
- Визначення точки ліктя: коли ми збільшуємо k , WCSS зазвичай зменшується, оскільки ми створюємо більше кластерів, які мають тенденцію фіксувати більше варіацій даних. Однак настає момент, коли додавання додаткових кластерів призводить лише до незначного зниження WCSS. Тут ми спостерігаємо форму «ліктя» на графіку.

Перед elbow point: збільшення k значно зменшує WCSS, вказуючи на те, що нові кластери ефективно охоплюють більше мінливості даних.

Після elbow point: додавання додаткових кластерів призводить до мінімального зниження WCSS, що свідчить про те, що ці додаткові кластери можуть бути непотрібними.

Мета полягає в тому, щоб визначити точку, де швидкість зниження WCSS різко змінюється, вказуючи на те, що додавання більшої кількості кластерів (поза цією точкою) дає меншу віддачу. Ця точка «ліктя» вказує на оптимальну кількість кластерів.

Спотворення (Distortion)

Спотворення вимірює середню квадратну відстань між кожною точкою даних та центром кластера, до якого вона належить. Це показник того, наскільки добре кластери відображають дані. Менше значення спотворення вказує на кращу кластеризацію.

$$Distortion = \frac{1}{n} \sum_{i=1}^n d(x_i, c_j)^2, \text{ де:}$$

- x_i – i -та точка даних,
- c_j – центр кластера, до якого належить

- $d(x_i, c_j)$ – евклідова відстань між точкою даних i її центром кластера.

1.10 Silhouette Analysis

Silhouette Analysis є одним із багатьох алгоритмів для визначення оптимальної кількості кластерів для алгоритму K-Means. В алгоритмі Silhouette ми припускаємо, що дані вже кластеризовано в k кластерів за допомогою техніки кластеризації. Потім для кожної точки даних ми визначаємо наступне:

$C(i)$ - Кластер, призначений i -й точці даних

$|C(i)|$ – Кількість точок даних у кластері, призначених i -й точці даних

$a(i)$ – Показує, наскільки добре i -та точка даних присвоєна її кластеру (середня відстань між кожною точкою в кластері).

$$a(i) = \frac{1}{|C(i)|-1} \sum_{C(i), i \neq j} d(i, j)$$

$b(i)$ – визначається як середня розбіжність з найближчим кластером, який не є даним кластером

$$b(i) = \min_{i \neq j} \left(\frac{1}{|C(j)|} \sum_{j \in C(j)} d(i, j) \right)$$

На рис. 4 схематично зображено величини $a(i)$ та $b(i)$.

Silhouette Score $s(i)$ визначається як $s(i) = \frac{b(i)-a(i)}{\max(b(i), a(i))}$. Ми визначаємо середнє значення $s(i)$ для кожного значення k і значення k , яке має максимальне значення $s(i)$, вважається оптимальною кількістю кластерів для алгоритму.

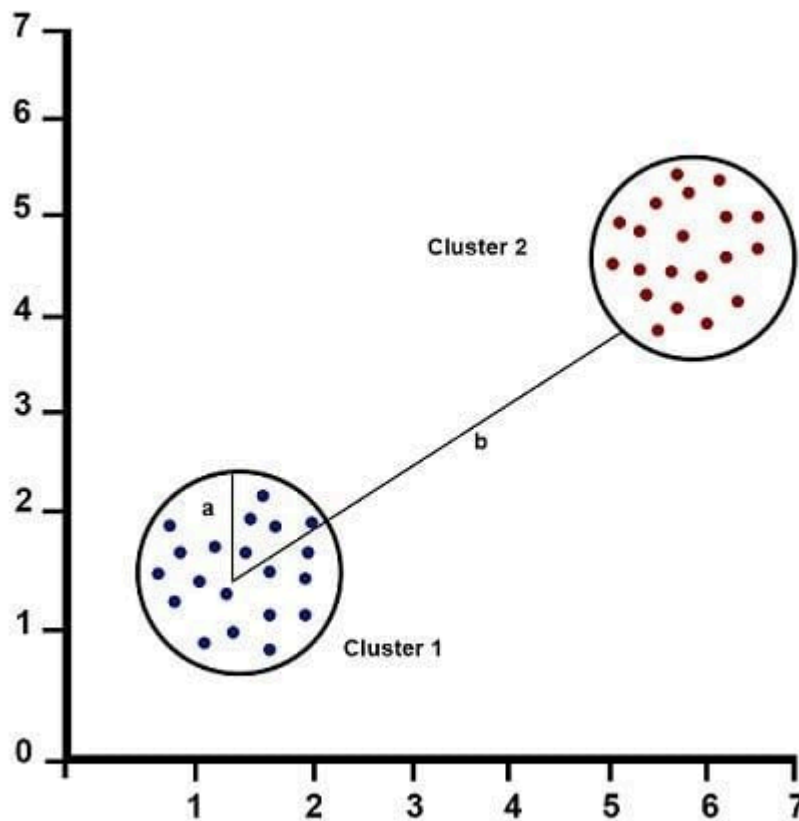


Рисунок 4 - величини *Silhouette Score*

Значення *Silhouette Score* коливається від -1 до 1. Нижче наведено інтерпретацію *Silhouette Score*.

- 1: Точки ідеально розподілені в кластері, і кластери легко розрізнити.
- 0: Кластери перекриваються.
- 1: точки неправильно розподілені в кластері.

1.11 Principal Component Analysis

PCA — це статистичний метод, який був введений математиком Карлом Пірсоном у 1901 році. Він працює шляхом перетворення даних з високою вимірністю в простір з нижчою вимірністю, максимально зберігаючи дисперсію (або розкидання) даних у новому просторі. Це допомагає зберегти найбільш важливі закономірності та зв'язки в даних.

Примітка: він віддає перевагу напрямкам, де дані змінюються найбільше (адже більше змін = більше корисної інформації).

Кроки методу PCA:

1. Стандартизація даних

Необхідно, щоб усі ознаки мали однаковий масштаб, інакше змінні з більшими значеннями (наприклад, зарплата 0–100000) будуть домінувати над змінними з меншими значеннями (наприклад, вік 0–100).

Формула стандартизації:

$$Z = \frac{X - \mu}{\sigma}, \text{ де } \mu - \text{середнє значення кожної ознаки, } \sigma - \text{стандартне}$$

відхилення кожної ознаки.

2. Обчислення коваріаційної матриці

Необхідно визначити, як ознаки змінюються разом. Це робиться за допомогою коваріаційної матриці, що показує взаємозв'язок між парами змінних.

Формула для обчислення коваріації між двома змінними

$$\text{cov}(x_1, x_2) = \frac{\sum_{i=1}^n (x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2)}{n-1}$$

Результат може бути:

- Додатним: обидві змінні збільшуються разом.
- Від'ємним: одна змінна збільшується, а інша зменшується.
- Нульовим: відсутній лінійний взаємозв'язок.

3. Обчислення власних значень та власних векторів

РСА визначає нові осі (напрямки), уздовж яких дані мають найбільшу дисперсію.

Перша головна компонента (PC1) – напрямок найбільшого розкиду даних.

Друга головна компонента (PC2) – наступний найважливіший напрямок, перпендикулярний до PC1.

Для цього використовуються власні вектори та власні значення.

Якщо матриця A є квадратною, то власний вектор X та власне значення λ задовольняють рівняння $AX = \lambda X$.

Це означає, що:

- A лише масштабує X без зміни його напрямку.
- Власні вектори визначають "стабільні напрями" матриці A .

Рівняння можна записати у вигляді $(A - \lambda I)X = 0$, де I – одинична матриця. Визначник цього рівняння має дорівнювати нулю: $|A - \lambda I| = 0$

Це рівняння називається характеристичним рівнянням, а його розв'язання дає власні значення λ , після чого знаходяться відповідні власні вектори X .

4. Вибір головних компонент і перетворення даних

Зберігаються лише ті головні компоненти, які пояснюють найбільшу частину дисперсії (наприклад, 95%). Дані проєктуються на ці компоненти, що дозволяє зменшити кількість вимірів без значної втрати інформації.

РСА є некерованим методом машинного навчання, оскільки не потребує міток цільових змінних. Його часто використовують для аналізу даних, візуалізації та підвищення ефективності моделей машинного навчання.

1.12 Візуалізація роботи РСА

Припустимо, що є набір даних з двома ознаками: радіус (x-вісь) та площа (y-вісь), який зображено на рис. 5.

РСА знаходить напрямки найбільшої варіативності:

- PC_1 (перша головна компонента) – напрямок, уздовж якого дані мають найбільшу дисперсію.
- PC_2 (друга головна компонента) – напрямок, перпендикулярний до PC_1 , який містить менше інформації.

Червоні штриховані лінії вказують на розкид даних уздовж різних напрямків. Оскільки розкид уздовж PC_1 більший, ніж уздовж PC_2 , це означає, що PC_1 несе більше корисної інформації.

Дані (сині точки) проєктуються на PC_1 , що ефективно зменшує вимірність набору даних з двох (радіус, площа) до одного (PC_1).

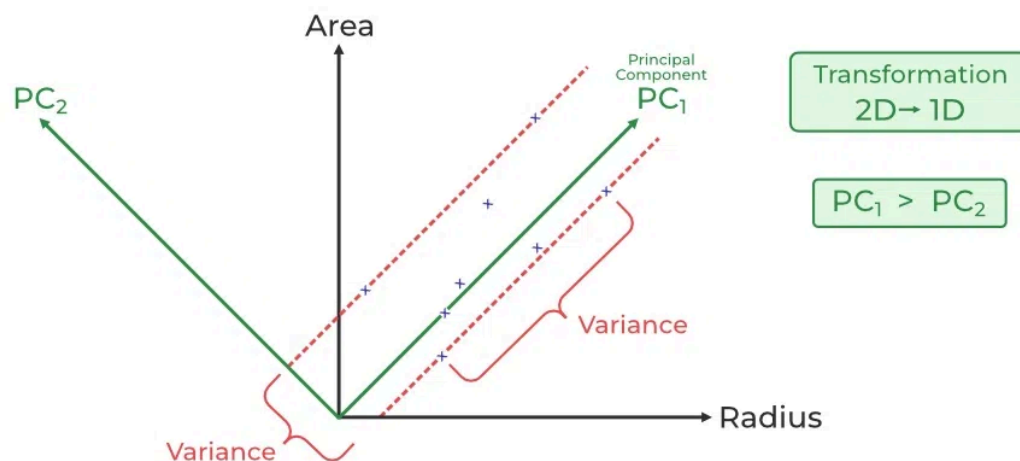


Рисунок 5 - 2D -> 1D трансформація за допомогою PCA

1.13 Метод лінійної регресії

Метод лінійної регресії — це метод моделювання залежності між скалярною змінною y та векторною (у загальному випадку) змінною X . У разі, якщо змінна X також є скаляром, регресію називають простою.

При використанні лінійної регресії взаємозв'язок між даними моделюється за допомогою лінійних функцій, а невідомі параметри моделі оцінюються за вхідними даними. Подібно до інших методів регресійного аналізу лінійна регресія повертає розподіл умовної ймовірності y в залежності від X , а не розподіл спільної ймовірності y та X , що стосується області мультиваріативного аналізу.

Загальна лінійна регресійна модель має вигляд:

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u, \text{ де:}$$

- y — залежна пояснювана змінна,
- (x_1, x_2, \dots, x_k) — незалежні пояснювальні змінні,
- u — випадкова похибка, розподіл якої в загальному випадку залежить від незалежних змінних, але математичне сподівання якої дорівнює нулеві.

Згідно з цією моделлю, математичне сподівання залежної змінної є лінійною функцією незалежних змінних: $E(y) = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u$

Вектор параметрів $(\beta_1, \beta_2, \dots, \beta_k)$ є невідомим і задача лінійної регресії полягає у пошуку цих параметрів на основі деяких експериментальних значень у та (x_1, x_2, \dots, x_k) . Тобто для деяких n експериментів мають бути відомими значення $\{x_{i1}, \dots, x_{ik}\}$, $n \in [1, n]$ незалежних змінних і відповідні їм значення y_i залежної змінної.

Згідно з означенням моделі для кожного експериментального випадку залежність між змінними визначається формулою:

$y = \beta_0 + \beta_1 x_{1,i} + \dots + \beta_k x_{k,i} + u_i$, або, у матричних позначеннях:

$y = X\beta + u$, де:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1K} \\ 1 & x_{21} & \cdots & x_{2K} \\ \vdots & \ddots & \vdots & \\ 1 & x_{n1} & \cdots & x_{nK} \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_K \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}.$$

На основі цих даних потрібно оцінити значення параметрів $(\beta_0, \beta_1, \dots, \beta_k)$, а також розподіл випадкової величини u .

Залежно від об'єктів, що досліджуються за допомогою лінійної регресії, та конкретних цілей дослідження можуть використовуватися різні методи оцінки невідомих параметрів. Найпопулярнішим є звичайний метод найменших квадратів. Він приймає за оцінку параметра значення, що мінімізують суму квадратів залишків по всіх спостереженнях:

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n \left| y_i - \beta_0 - \sum_{j=1}^K X_{ij} \beta_j \right|^2 = \arg \min_{\beta} \|y - X\beta\|^2.$$

Метод найменших квадратів можна застосувати у будь-яких задачах, в яких ранг матриці X рівний кількості її стовпців. Також цей метод дає простий аналітичний вираз для оцінки параметрів: $\hat{\beta} = (X'X)^{-1}X'y$.

У випадку класичної моделі лінійної регресії оцінка методу найменших квадратів є незміщеною, змістовною і найкращою лінійною незміщеною оцінкою.

У випадку коли деякі з умов класичної лінійної регресії не виконуються метод найменших квадратів може не бути оптимальним. Так для узагальненої моделі лінійної регресії, де $V(u|X) = \sigma^2 W$ (W — відома додатноозначена матриця) найкращою лінійною незміщеною оцінкою є оцінка, що одержується так званим узагальненим методом найменших квадратів:

$$\hat{\beta} = (X^T W^{-1} X)^{-1} X^T W^{-1} y.$$

Серед інших методів оцінювання існує метод найменших модулів, що знаходить мінімум суми не квадратів відхилень, а їх абсолютних значень:

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n \left| y_i - \beta_0 - \sum_{j=1}^K X_{ij} \beta_j \right|.$$

1.14 Метод логістичної регресії

Логістична регресія — це метод моделювання залежності між бінарною (дискретною) залежною змінною та одним або кількома незалежними змінними. Логістична регресія застосовується, коли результат є категоріальним, а саме для оцінки ймовірності належності об'єкта до певної категорії (наприклад, ймовірність настання певної події).

При використанні логістичної регресії взаємозв'язок між змінними моделюється за допомогою функції логістичної функції, яка дає значення в інтервалі від 0 до 1, що є інтерпретованим як ймовірність.

Загальна логістична регресійна модель має вигляд:

$$P(y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1x_1+\dots+\beta_kx_k)}}, \text{ де:}$$

- $P(y = 1|X)$ — ймовірність того, що залежна змінна y набуде значення 1 (наприклад, подія станеться),
- β_0 — вільний параметр,
- $(\beta_1, \beta_2, \dots, \beta_k)$ — параметри моделі (ваги), які потрібно оцінити,
- (x_1, x_2, \dots, x_k) — незалежні пояснювальні змінні.

Математичне сподівання залежної змінної (яке є ймовірністю події) в логістичній регресії є функцією лінійної комбінації незалежних змінних через логіт-функцію: $E(y|X) = \frac{1}{1+e^{-(\beta_0+\beta_1x_1+\dots+\beta_kx_k)}}$.

Як і в лінійній регресії, задача логістичної регресії полягає в оцінці невідомих параметрів $(\beta_1, \beta_2, \dots, \beta_k)$, що мінімізують функцію втрат на основі спостережуваних даних X і y . Проте в даному випадку, через бінарну природу залежної змінної, найпоширенішим методом оцінки параметрів є метод максимальної ймовірності.

Метод максимальної вірогідності

Метод максимальної ймовірності є підходом до оцінки параметрів логістичної регресії. Він полягає в максимізації ймовірності спостережуваних даних за допомогою підбору параметрів моделі, які максимізують функцію правдоподібності. Вона для логістичної регресії має вигляд:

$$L(\beta_0, \beta_1, \dots, \beta_K) = \prod_{i=1}^n P(y_i|x_i)^{y_i} (1 - P(y_i|x_i))^{(1-y_i)}$$

де:

- $P(y_i, x_i)$ — ймовірність того, що спостережуване значення y_i дорівнює

1, обчислене через логістичну функцію,

- y_i — спостережуване значення залежної змінної для i -го спостереження.

Логарифм цієї функції дає логарифмічну функцію правдоподібності, яку часто максимізують для оцінки параметрів:

$$\log L(\beta_0, \beta_1, \dots, \beta_K) = \sum_{i=1}^n [y_i \log P(y_i|x_i) + (1 - y_i) \log(1 - P(y_i|x_i))]$$

Мультиноміальна логістична регресія

Мультиноміальна логістична регресія застосовується для задач, де залежна змінна має більше ніж два класи, і кожен клас є відокремленим від інших. Класичним прикладом є ситуація, коли необхідно класифікувати об'єкти в одну з кількох категорій.

Замість того, щоб мати лише одну функцію ймовірності для двох класів, у мультиноміальній логістичній регресії модель оцінює $K-1$ ймовірностей для K класів.

Формула для оцінки ймовірності для кожного класу має вигляд:

$$P(y = k|X) = \frac{e^{\beta_k^T X}}{1 + \sum_{j=1}^{K-1} e^{\beta_j^T X}}$$

де:

- k — це поточний клас,
- X — це вектор вхідних змінних,
- β_k — це вектор параметрів для класу k .

1.15 Random Forest Regression

Метод Random Forest Regression базується на ансамблевому підході, де кілька моделей дерев рішень використовуються для отримання більш стабільних і точних прогнозів. Кожне дерево в ансамблі працює незалежно, і прогноз кожного дерева комбінується для отримання фінального результату.

Кожне дерево в Random Forest є регресійним деревом, яке намагається прогнозувати значення залежної змінної y на основі набору незалежних змінних $X = (x_1, x_2, \dots, x_p)$.

Для побудови дерева використовуються такі кроки:

- Вибір найбільш значущої ознаки: На кожному етапі побудови дерева вибирається ознака, яка найбільше зменшує непевність у прогнозі, зазвичай за допомогою критерію середнього квадратичного відхилення (MSE).
- Розподіл на підгрупи: Дані розподіляються на дві групи (підвузли), де кожен вузол намагається мінімізувати MSE:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2, \text{ де } y_i \text{ — фактичне значення для } i\text{-го}$$

спостереження, а \hat{y} — прогнозоване значення для цього спостереження.

- Прогноз на основі дерева: Кожен лист дерева надає прогноз на основі середнього значення залежної змінної y для всіх спостережень, які потрапили в цей лист.

Random Forest використовує ліс з T дерев. Кожне дерево генерується за допомогою випадкового підбору даних та ознак. Для кожного дерева, побудованого на основі підмножини даних D_t , прогноз є середнім значенням результатів дерева:

$$\hat{y}_t = \frac{1}{n_t} \sum_{i=1}^{n_t} y_i^{(t)}, \text{ де:}$$

- \hat{y}_t — прогноз, отриманий від t-го дерева;
- $y_i^{(t)}$ — прогноз для ііі-го спостереження в t-му дереві;
- n_t — кількість спостережень в t-му дереві.

Фінальний прогноз \hat{y} для Random Forest є середнім прогнозом усіх дерев:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T \hat{y}_t = \frac{1}{T} \sum_{i=1}^T \left(\frac{1}{n_t} \sum_{i=1}^{n_t} y_i^{(t)} \right), \text{ де } T \text{ — кількість дерев у лісі.}$$

Random Forest має можливість оцінити важливість кожної ознаки в процесі класифікації. Важливість ознаки вимірюється через вплив цієї ознаки на точність моделі. Це можна оцінити через зміни в точності при випадковому перемішуванні значень цієї ознаки.

Математично, важливість ознаки j для моделі можна визначити через:

$$Importance(j) = \frac{1}{T} \sum_{i=1}^T \Delta Accuracy_j^{(t)}, \text{ де } \Delta Accuracy_j^{(t)} \text{ — зміна точності } t\text{-го}$$

дерева при випадковому перемішуванні значень ознаки j .

1.16 Random Forest Classifier

Кожне дерево в Random Forest є класичним деревом рішень. Процес побудови кожного дерева включає кілька етапів:

1. Вибір найбільш значущої ознаки: на кожному етапі побудови дерева вибирається ознака, яка найкраще розділяє спостереження на класи. Для цього зазвичай використовують такі критерії, як індекс Джині або ентропія.
2. Розподіл на підгрупи (вузли): дерева створюють підвузли, де кожен вузол

намагається розділити дані таким чином, щоб зменшити непевність у класах. Для кожного поділу обирається та ознака, що мінімізує значення обраного критерію, наприклад, індекс Джині:

$$Gini = 1 - \sum_{k=1}^K p_k^2, \text{ де } p_k \text{ — ймовірність спостереження належати класу } k.$$

Ентропія:

$$H(y) = - \sum_{k=1}^K p_k \log(p_k), \text{ де } p_k \text{ — ймовірність належності до класу } k.$$

3. Прогноз для кожного дерева: Для кожного дерева t , яке робить прогноз для спостереження x_i , прогноз $y_i^{(t)}$ може бути одним із класів $\{c_1, c_2, \dots, c_K\}$, де K — кількість класів. Прогноз для Random Forest формується шляхом голосування, де кожне дерево віддає свій голос (клас), а клас, який отримав більшість голосів, стає фінальним прогнозом:

$$\hat{y} = \text{mode}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T), \text{ де:}$$

- \hat{y}_t — прогноз t -го дерева;
- mode — функція, яка вибирає найбільш частий клас серед прогнозів.

1.17 XGBoost Regression

XGBoost Regression використовується для регресії, тобто для прогнозування безперервних числових значень.

XGBoost (Extreme Gradient Boosting) — це метод градієнтного бустингу, який побудований на основі лісу рішень, створених за допомогою рішень дерев. У XGBoost кожне нове дерево в лісі створюється для виправлення помилок попередніх дерев та має на меті мінімізувати функцію втрат за

допомогою градієнтного спуску.

Для XGBoost Regression використовують середньоквадратичну помилку (MSE) як функцію втрат:

$$L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \text{ де:}$$

- N — кількість спостережень,
- y_i — фактичне значення,
- \hat{y}_i — передбачене значення.

Крім того, XGBoost використовує регуляризацію для уникнення перенавчання, що дає йому перевагу над іншими алгоритмами градієнтного бустингу.

$$\text{Regularized Loss} = L + \lambda \sum_{t=1}^T ||w_t||^2, \text{ де } \lambda \text{ — параметр регуляризації, } w_t \text{ —}$$

коефіцієнти моделі, T — кількість дерев у моделі.

Алгоритм побудови моделі

- Початкова модель: Початкове передбачення $\hat{y}_i^{(0)}$ встановлюється як середнє значення цільової змінної y .
- Додавання дерев: Наступні дерева додаються за допомогою градієнтного спуску. Для кожного дерева мінімізується градієнт функції втрат по відношенню до поточного передбачення.
- Градієнтний спуск: Для кожного дерева знаходиться градієнт помилки попереднього дерева, і нове дерево навчиться виправляти цю помилку.

Для кожного дерева t , передбачення коригується:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \gamma * f_t(x_i), \text{ де } f_t(x_i) \text{ — передбачення } t\text{-го}$$

дерева для ознаки x_i , γ — коефіцієнт коригування.

1.18 XGBoost Classifier

XGBoost Classifier використовується для класифікаційних задач, де метою є передбачити категоріальні (дискретні) вихідні значення.

У XGBoost Classifier використовується функція втрат, яка відповідає типу задачі класифікації.

Для бінарної класифікації застосовують логістичну втрату:

$$L = - [y * \log(p) + (1 - y)\log(1 - p)], \text{ де:}$$

- y — фактична мітка класу (0 або 1),
- p — ймовірність того, що об'єкт належить до класу 1.

Для мультикласової класифікації застосовують втрату категоріальної крос-ентропії (Categorical Cross-Entropy Loss):

$$L = - \sum_{c=1}^C y_c \log(p_c), \text{ де:}$$

- C — кількість класів,
- y_c — фактична ймовірність належності до класу c ,
- p_c — передбачена ймовірність для класу c .

2. Реалізація

2.1 Моделі для передбачення ступеня серйозності дорожніх робіт на основі даних

Використовуються три основні моделі:

1. Логістична регресія — класична модель для багатокласової класифікації, яка застосовує логістичну функцію для передбачення ймовірностей приналежності події до конкретного класу серйозності.
2. Random Forest — ансамблева модель, що складається з багатьох дерев рішень, кожне з яких дає прогноз, а остаточний результат визначається за допомогою голосування.
3. XGBoost — потужна модель, що використовує градієнтний бустинг для поліпшення точності передбачень через поступове вдосконалення попередніх моделей.

Для кожної з цих моделей проводиться навчання на тренувальних даних, після чого оцінюється їхня ефективність за допомогою метрик, таких як точність (accuracy), точність (precision), повнота (recall) та F1-метрика. Оскільки різні характеристики в даних можуть мати різні масштаби, застосовується масштабування з використанням StandardScaler. Окрім цього, для кращого розуміння помилок моделей будуються матриці сплутаності, що дозволяє оцінити, наскільки добре моделі передбачають різні рівні серйозності дорожніх подій.

♦ Model: Logistic Regression

Accuracy: 0.9093

Precision: 0.8886

Recall: 0.9093

F1-score: 0.8922

||| Classification Report:

	precision	recall	f1-score	support
0	1.00	0.00	0.00	183
1	0.92	0.98	0.95	204583
2	0.34	0.04	0.07	2531
3	0.58	0.29	0.39	19923
accuracy			0.91	227220
macro avg	0.71	0.33	0.35	227220
weighted avg	0.89	0.91	0.89	227220

♦ Model: Random Forest

Accuracy: 0.9857

Precision: 0.9856

Recall: 0.9857

F1-score: 0.9852

||| Classification Report:

	precision	recall	f1-score	support
0	0.84	0.40	0.54	183
1	0.99	1.00	0.99	204583
2	0.97	0.83	0.89	2531
3	0.99	0.88	0.93	19923
accuracy			0.99	227220
macro avg	0.94	0.78	0.84	227220
weighted avg	0.99	0.99	0.99	227220

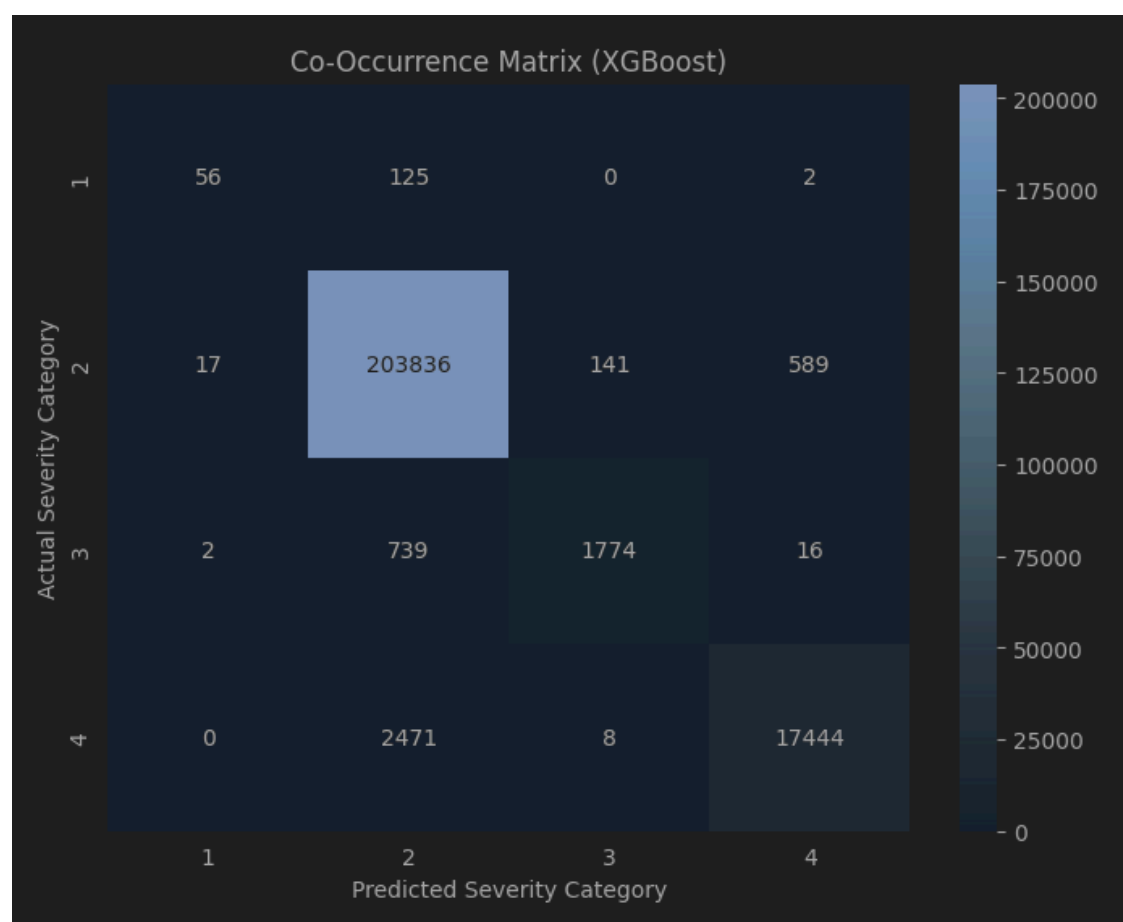
```

♦ Model: XGBoost
Accuracy: 0.9819
Precision: 0.9815
Recall: 0.9819
F1-score: 0.9812

||| Classification Report:

```

	precision	recall	f1-score	support
0	0.75	0.31	0.43	183
1	0.98	1.00	0.99	204583
2	0.92	0.70	0.80	2531
3	0.97	0.88	0.92	19923
accuracy			0.98	227220
macro avg	0.90	0.72	0.78	227220
weighted avg	0.98	0.98	0.98	227220



2.2 Кластеризація даних

Використовується метод K-Means для виявлення природних груп (кластерів) у даних, що стосуються різних характеристик, таких як відстань, температура,

швидкість вітру та вологість.

Основні етапи виконання кластеризації:

- Фільтрація даних: Спочатку з набору даних виключаються всі рядки, де швидкість вітру перевищує 50 миль на годину. Це дозволяє зосередитися на подіях з меншими значеннями швидкості вітру, що можуть бути більш релевантними для аналізу.
- Масштабування даних: Оскільки різні ознаки (відстань, температура, швидкість вітру, вологість) мають різні одиниці виміру, для коректного порівняння їх значення стандартизується за допомогою `StandardScaler`. Це перетворює всі ознаки на масштаб з нульовим середнім значенням та одиничним стандартним відхиленням.

Вибір оптимальної кількості кластерів:

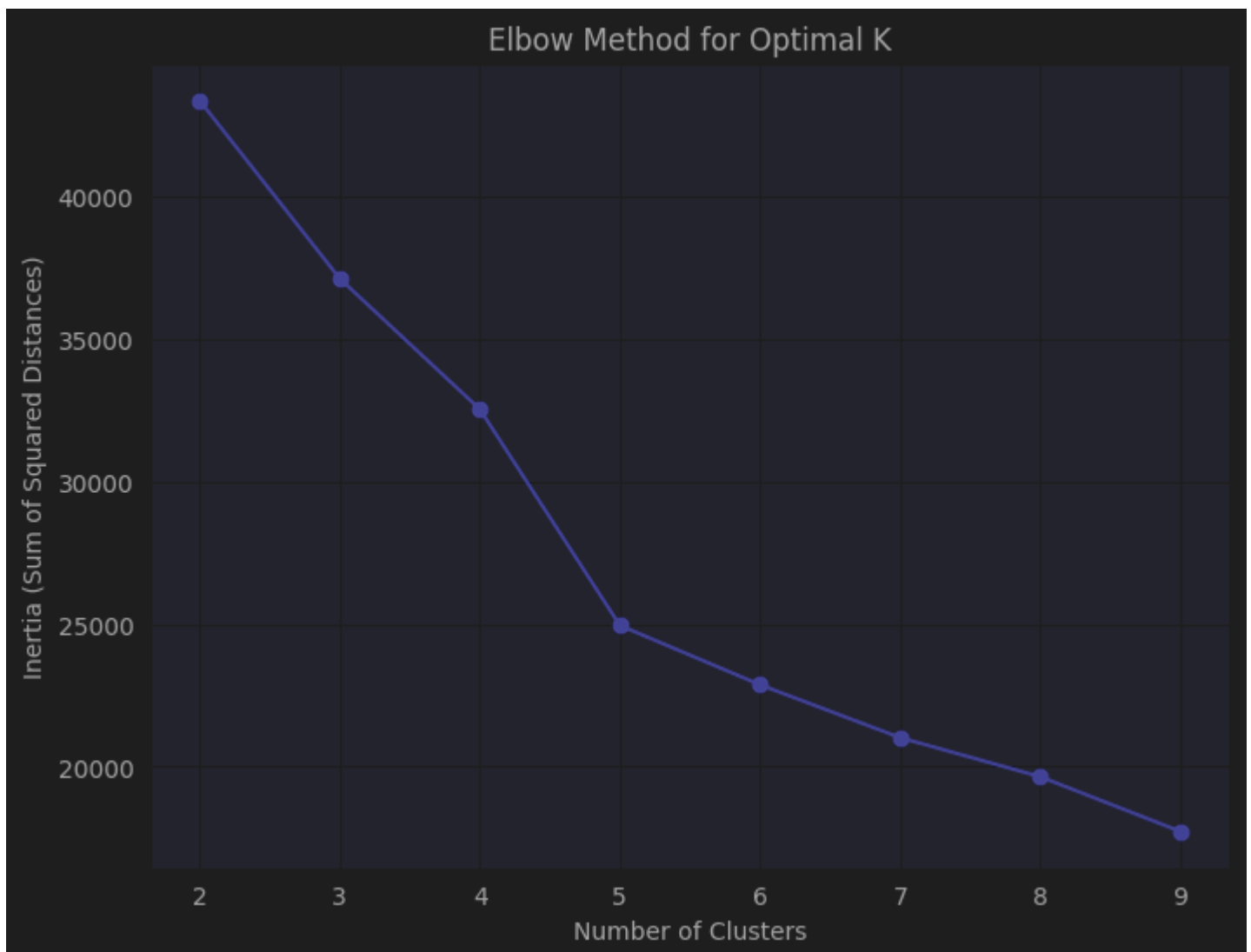
- Метод ліктя: Для того, щоб визначити найкращу кількість кластерів, проводиться обчислення інерції для різних значень кількості кластерів (від 2 до 10). Інерція визначає, наскільки добре дані підходять до обраних кластерів — чим менша інерція, тим краще.
- Метод силуету: Цей метод дає змогу оцінити, наскільки добре кластеризовані дані, порівнюючи схожість об'єкта з його власними кластерами та з іншими. Зазвичай вибирається кількість кластерів, що дає найкращий показник силуету.

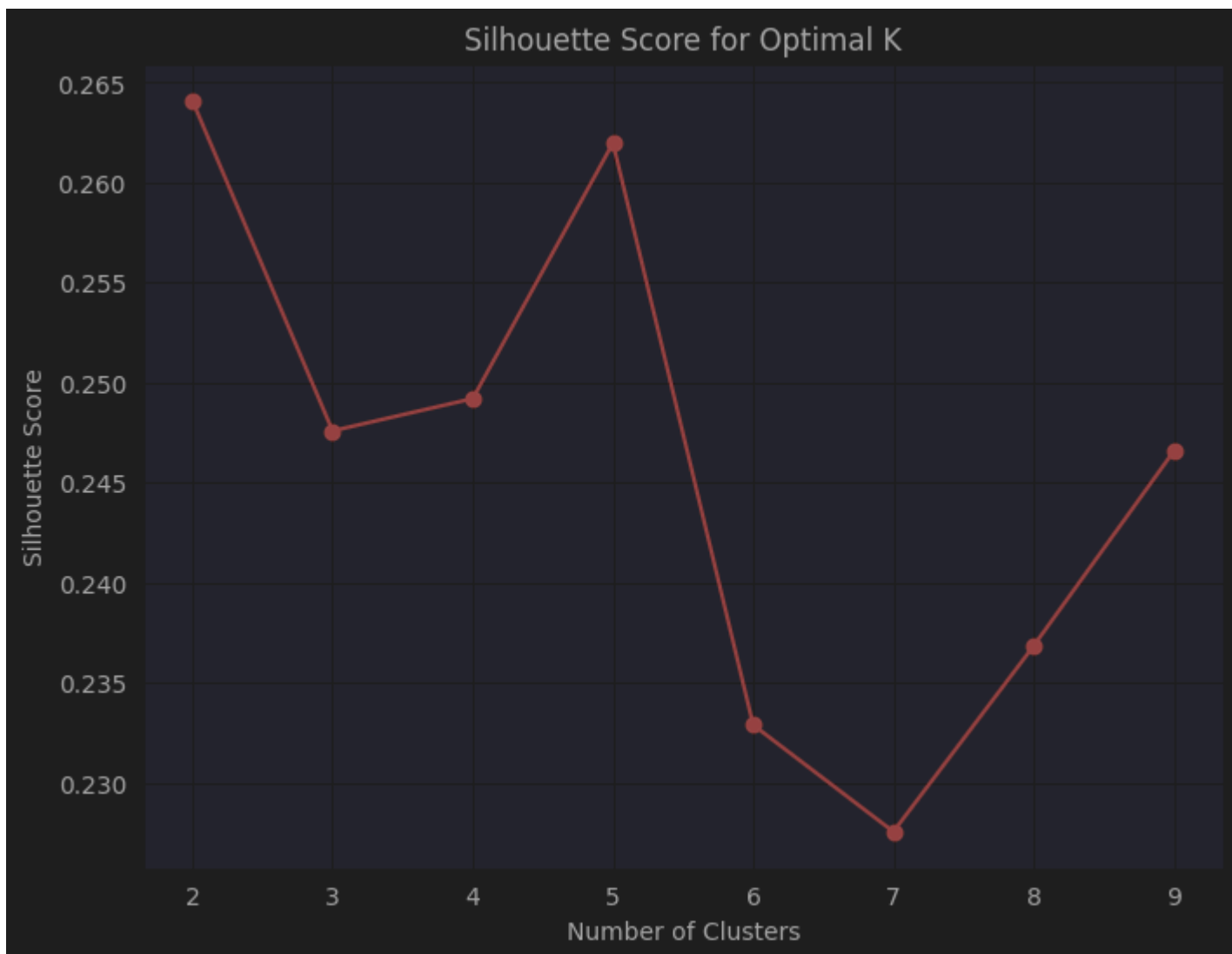
Навчання моделі K-Means: Після визначення оптимальної кількості кластерів (у даному випадку вибрано 5 кластерів) проводиться кластеризація даних. Для кожного об'єкта (даних про дорожні роботи) на основі його характеристик визначається належність до одного з кластерів.

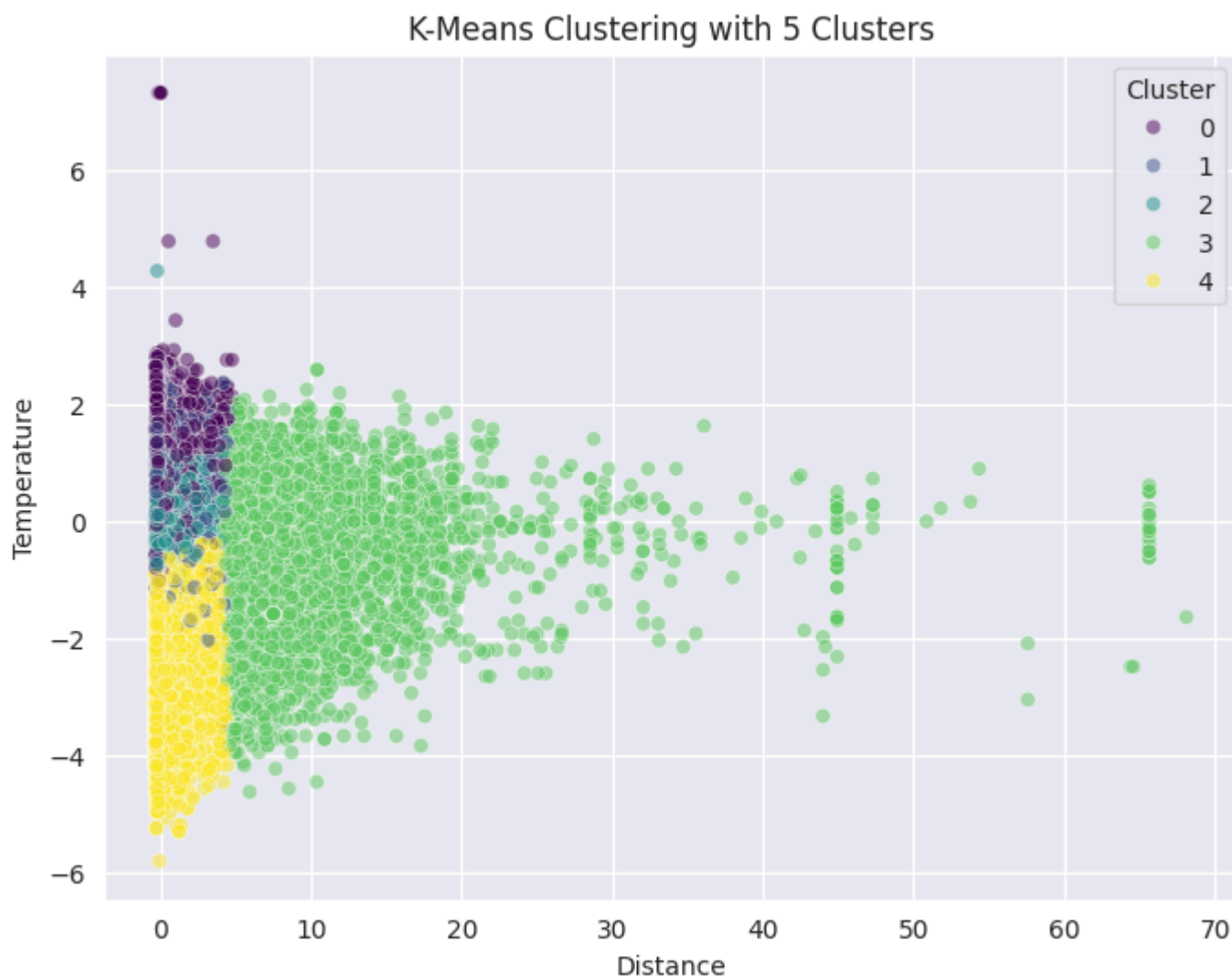
Візуалізація кластерів:

- Спочатку будуються графіки для кожної пари ознак (наприклад, відстань і температура, відстань і швидкість вітру), де кольори точок відображають належність до конкретного кластера.
- Потім застосовується PCA (Головні компоненти), щоб зменшити кількість вимірів і спростити візуалізацію, зображаючи дані в двовимірному просторі. Таким чином можна побачити, як добре розподілені кластери в новому просторі.

Аналіз компонент PCA: Виводиться таблиця, що показує внесок кожної ознаки у побудову головних компонент. Це дає уявлення про те, які ознаки найважливіші для класифікації даних у новому просторі.







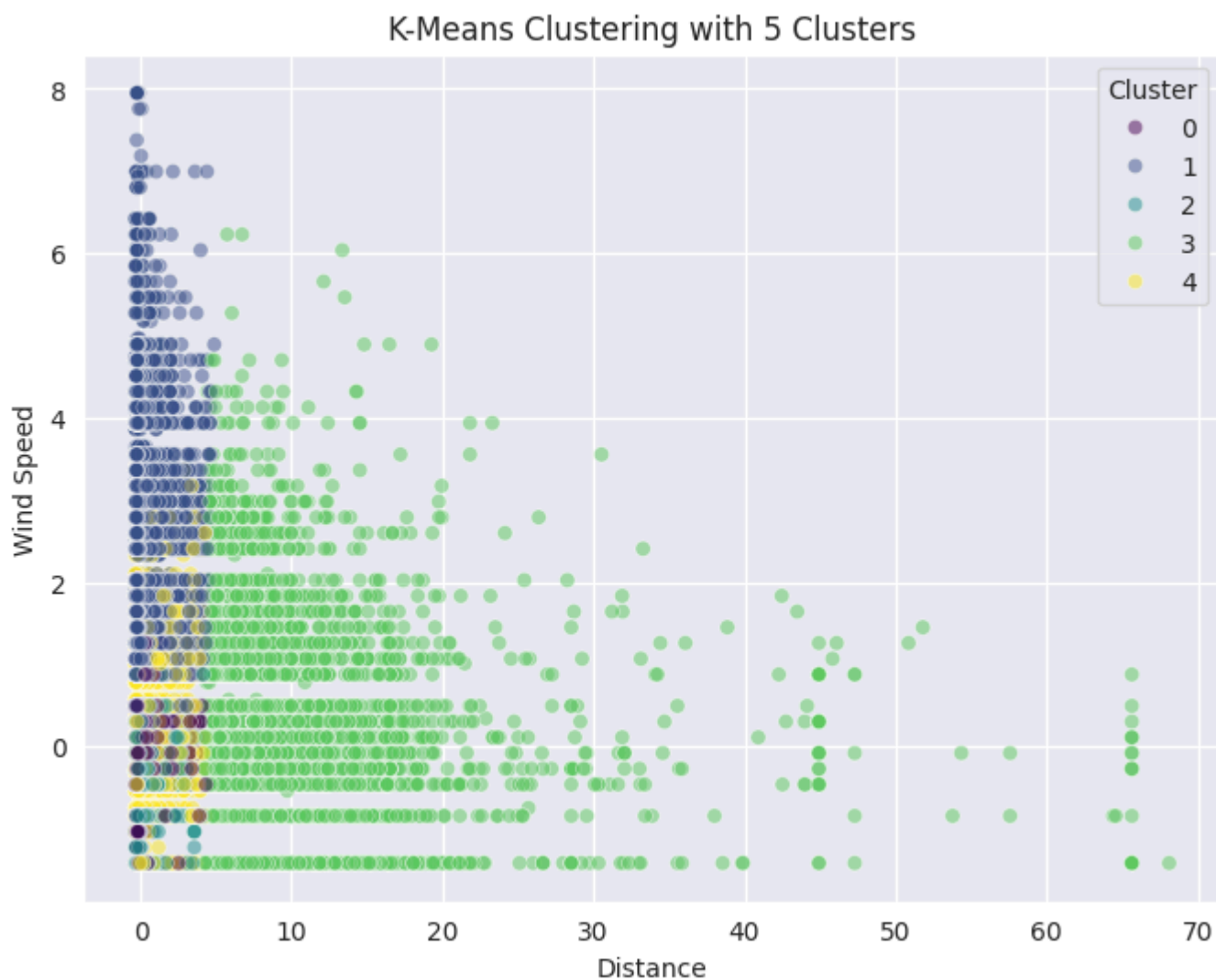
Бачимо, що всі чотири кластери, крім зеленого (№3) мають приблизно однакове максимальну протяжність дорожніх робіт (~ 5 од.)

Жовтий кластер (№4) має температури < 0 од.

Кластери №1 та №2 мають температури ~ 0 од.

Бачимо, що при збільшенні дистанції мінімальна температура збільшується майже лінійно.

І аналогічно, при збільшенні дистанції максимальна температура зменшується майже лінійно.



Бачимо, що значення швидкості вітру можна дискретизувати.

При збільшенні дистанції максимальна швидкість вітру зменшується приблизно лінійно.

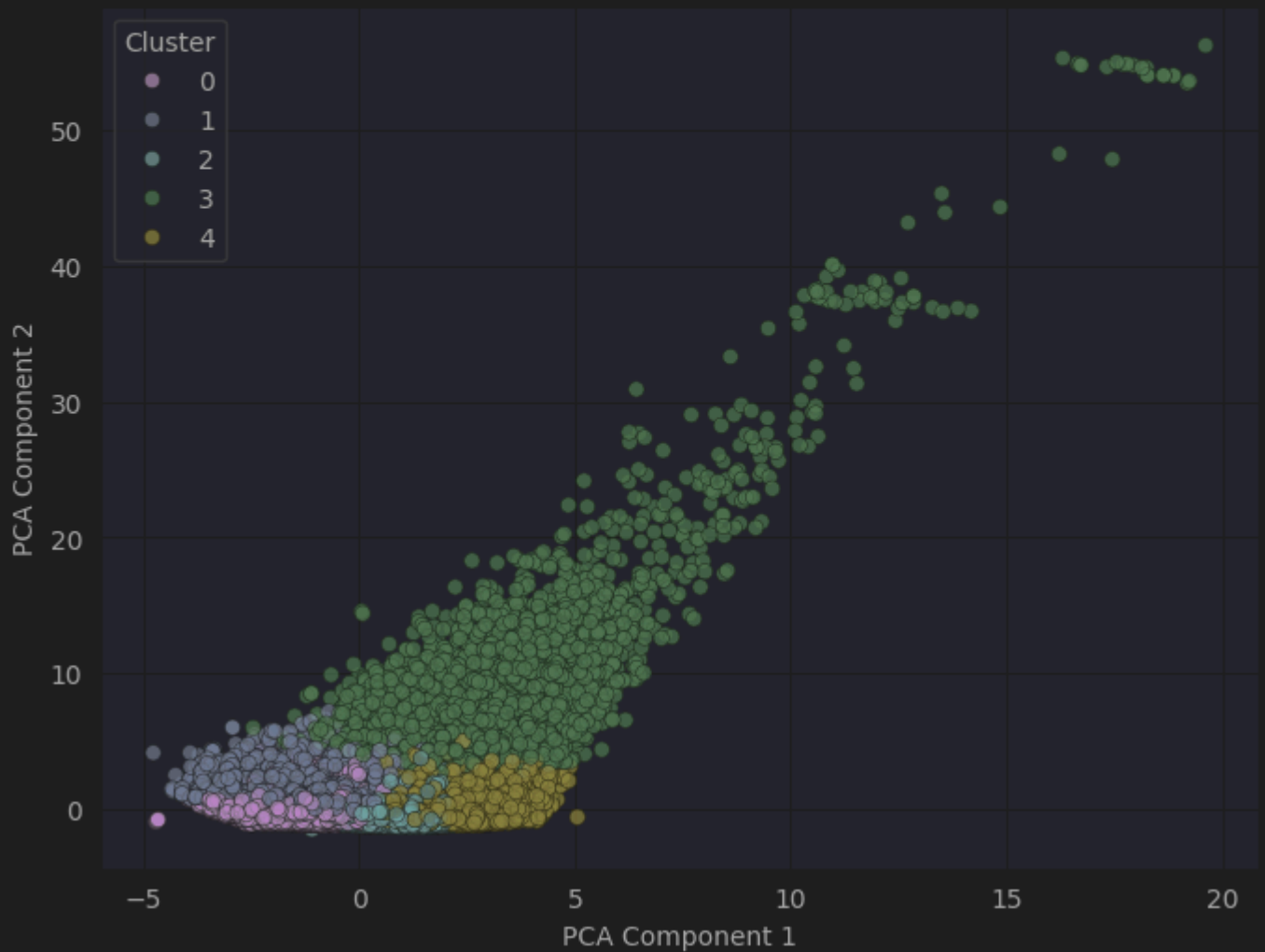


Знову бачимо, що всі чотири кластери, крім зеленого (№3) мають приблизно однакове максимальну протяжність дорожніх робіт (~ 5 од.).

Feature contributions to each principal component:

	Distance(mi)	Temperature(F)	Wind_Speed(mph)	Humidity(%)
PC1	0.25737116	-0.61767636	-0.36796312	0.64563081
PC2	0.83542192	-0.07447984	0.53544203	-0.09912016

K-Means Clustering with 5 Clusters (PCA Reduced to 2D)



PC1 (Головна компонента 1):

Позитивно впливає вологість (+0.646) та дистанція (+0.257) → більші значення PC1 відповідають місцям з високою вологістю та більшою дистанцією.

Негативно впливає температура (-0.617) → зменшення PC1 пов'язане з вищою температурою.

PC2 (Головна компонента 2):

Сильний позитивний вплив має дистанція (+0.835) → зростання PC2 асоційоване з віддаленістю.

Швидкість вітру (+0.535) також позитивно впливає → отже, дистанція та вітер корелюють.

Температура (-0.074) та вологість (-0.099) мають слабкий вплив.

Бачимо лінійну тенденцію в зеленому кластері. При збільшенні PC1 майже лінійно збільшується і PC2.

2.3 Моделі для передбачення тривалості дорожніх робіт на основі даних

У коді виконується побудова та оцінка моделей для прогнозування тривалості будівельних робіт.

1. Підготовка даних:

- Відбір релевантних ознак та розділення на навчальну та тестову вибірки.

2. Навчання моделей:

- Використовуються три підходи: лінійна регресія, випадковий ліс, XGBoost.
- Паралельне навчання моделей для прискорення обчислень.

3. Оцінка якості:

- Визначаються метрики MAE, RMSE та коефіцієнт детермінації R^2 .
- Розраховується довірчий інтервал для прогнозів.

4. Візуалізація результатів:

- Порівняння прогнозованих значень із фактичними.

- Аналіз розподілу помилок (залишків).

Завдання — знайти найкращу модель для точного передбачення тривалості робіт.

Linear Regression:

MAE: 758.9198

RMSE: 1721.1312

R^2 : 0.0135

95% CI: ± 20.0172

Random Forest Regressor:

MAE: 659.1461

RMSE: 1548.7430

R^2 : 0.2012

95% CI: ± 17.9868

XGBoost Regressor:

MAE: 657.0739

RMSE: 1568.8675

R^2 : 0.1803

95% CI: ± 18.2454

Бачимо, що Random Forest Regressor є найкращою моделлю:

- Найвище R^2 (0.2012) (найбільша частина варіації залежної змінної пояснена моделлю).
- Найменший RMSE (1548.74), що вказує на кращу узагальнюючу здатність.
- Найменший довірчий інтервал з рівнем довіри 0.95, що свідчить про стабільність прогнозів.

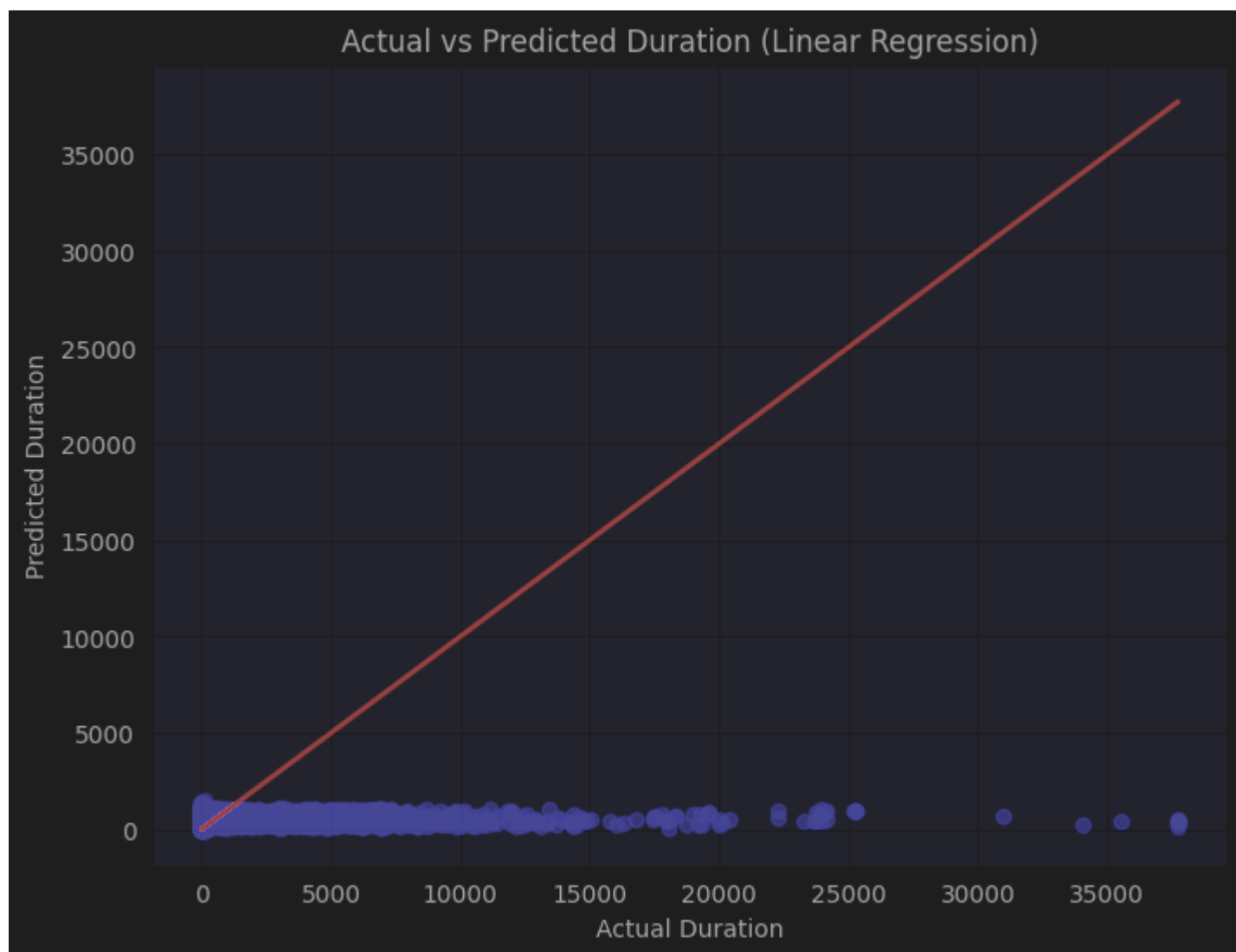
Linear Regression є найгіршою:

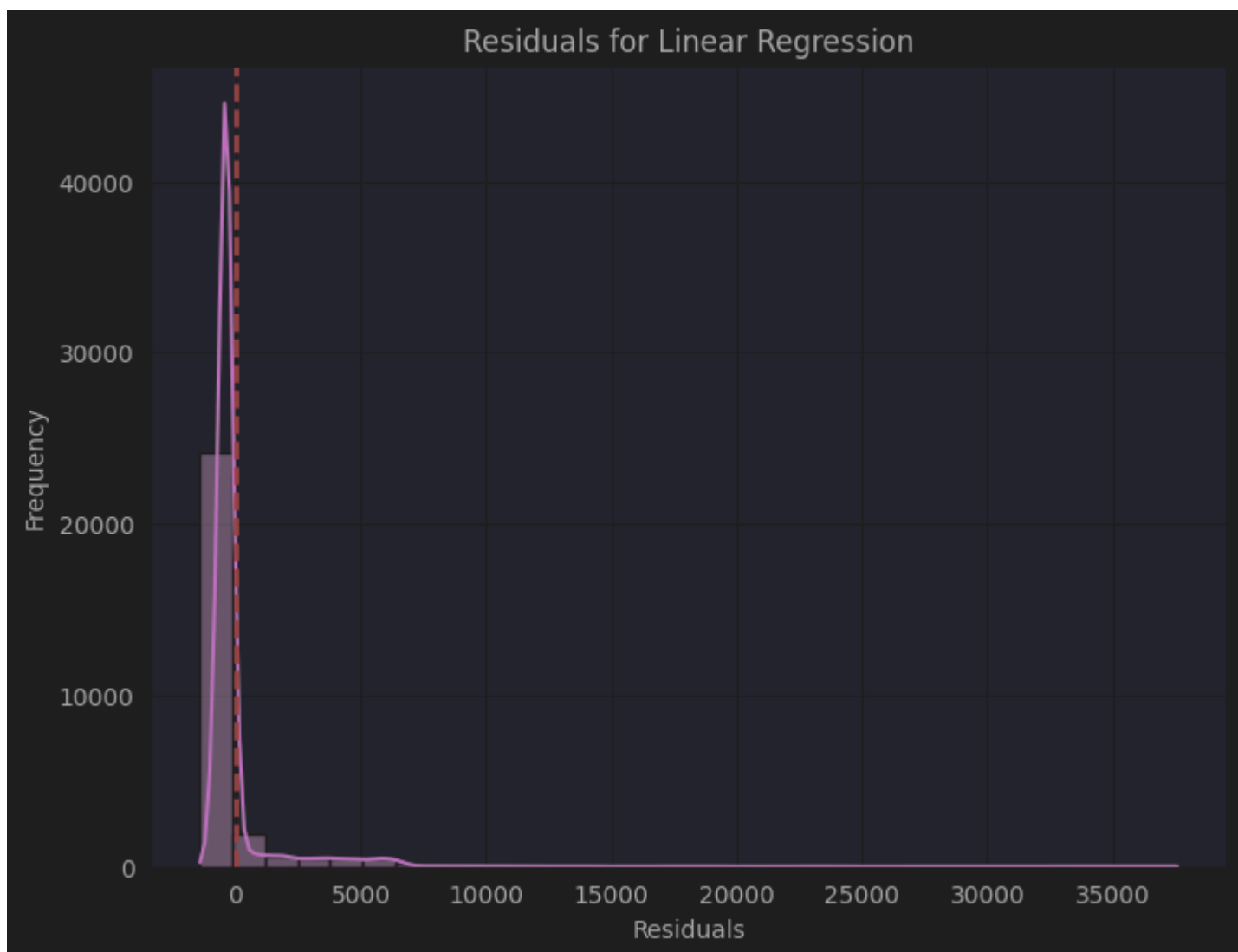
- Дуже низький R^2 (0.0135) означає, що модель майже не пояснює

варіацію даних.

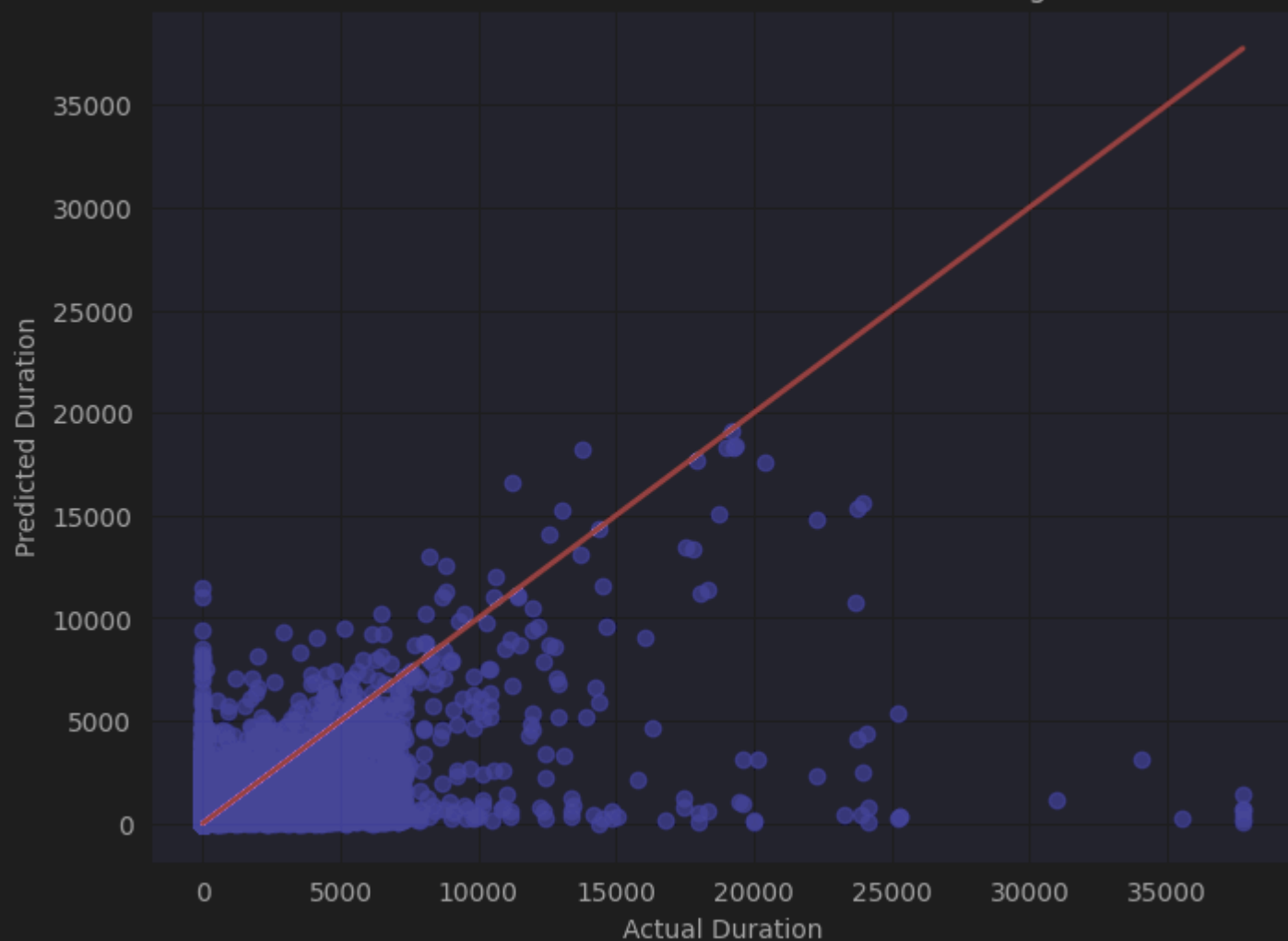
- Найвищі значення MAE та RMSE.
- Найширший довірчий інтервал, що вказує на нестабільні прогнози.

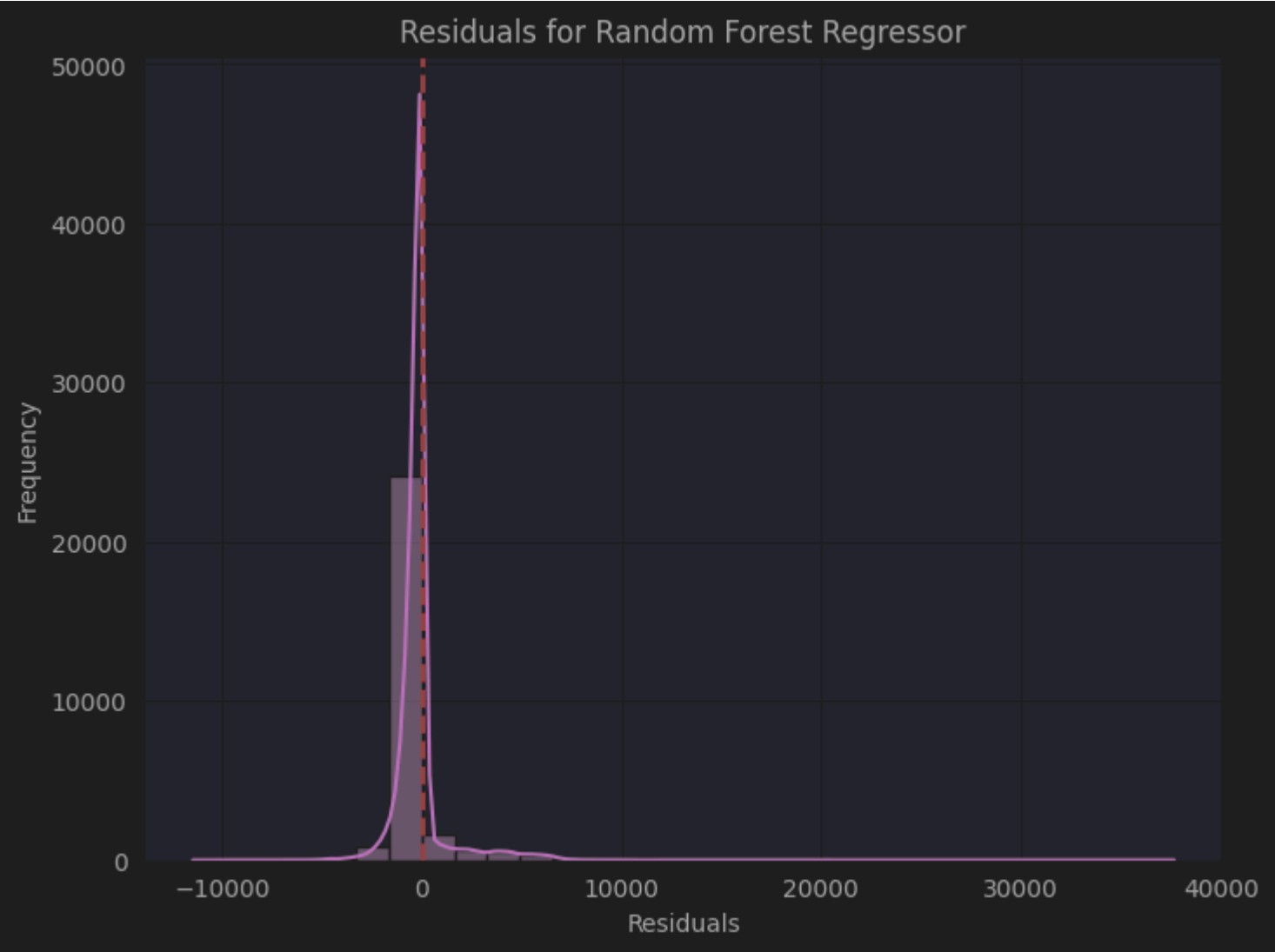
XGBoost Regressor має подібну ефективність до Random Forest, але трохи гірші показники RMSE та довірчого інтервалу.



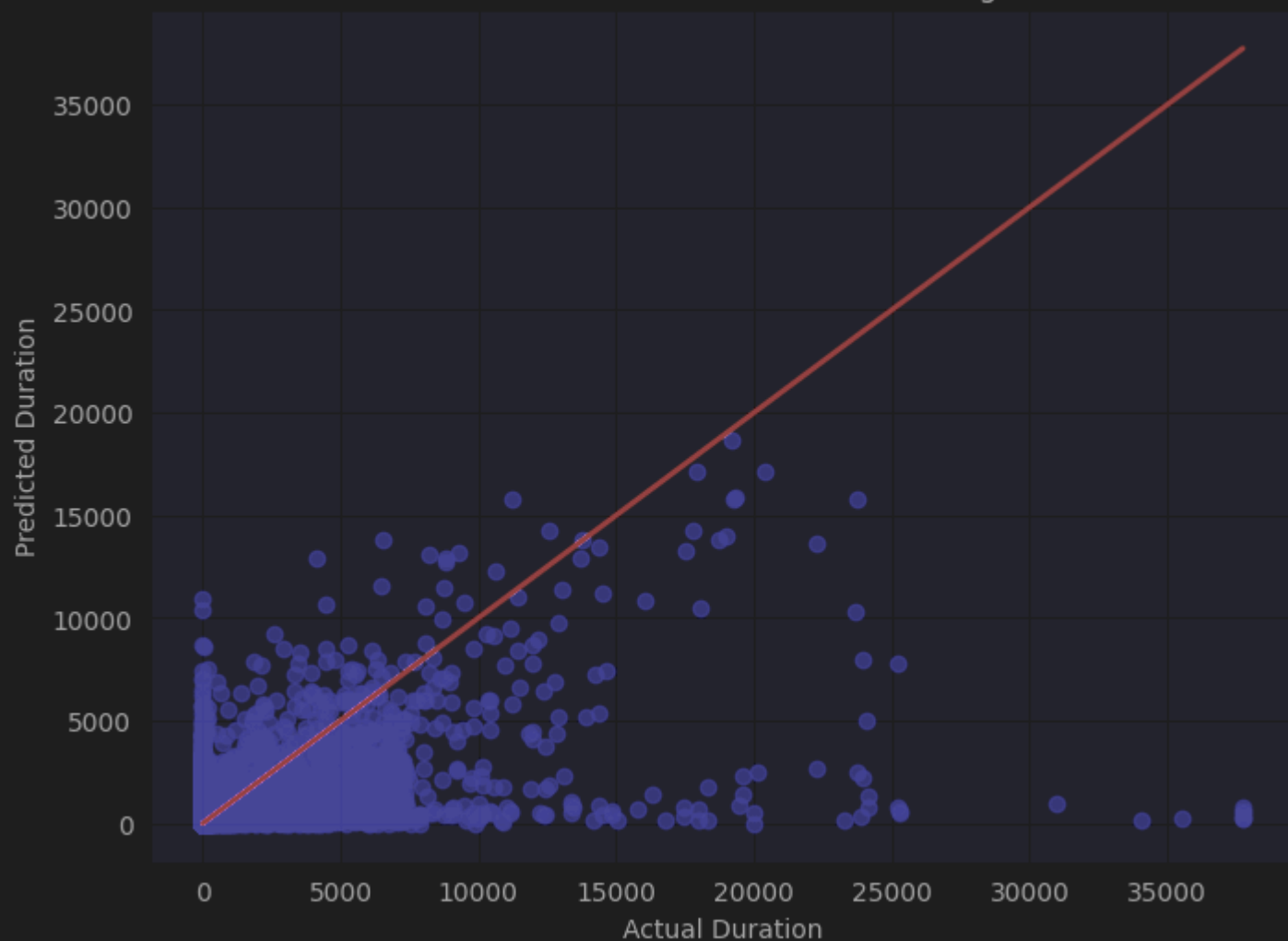


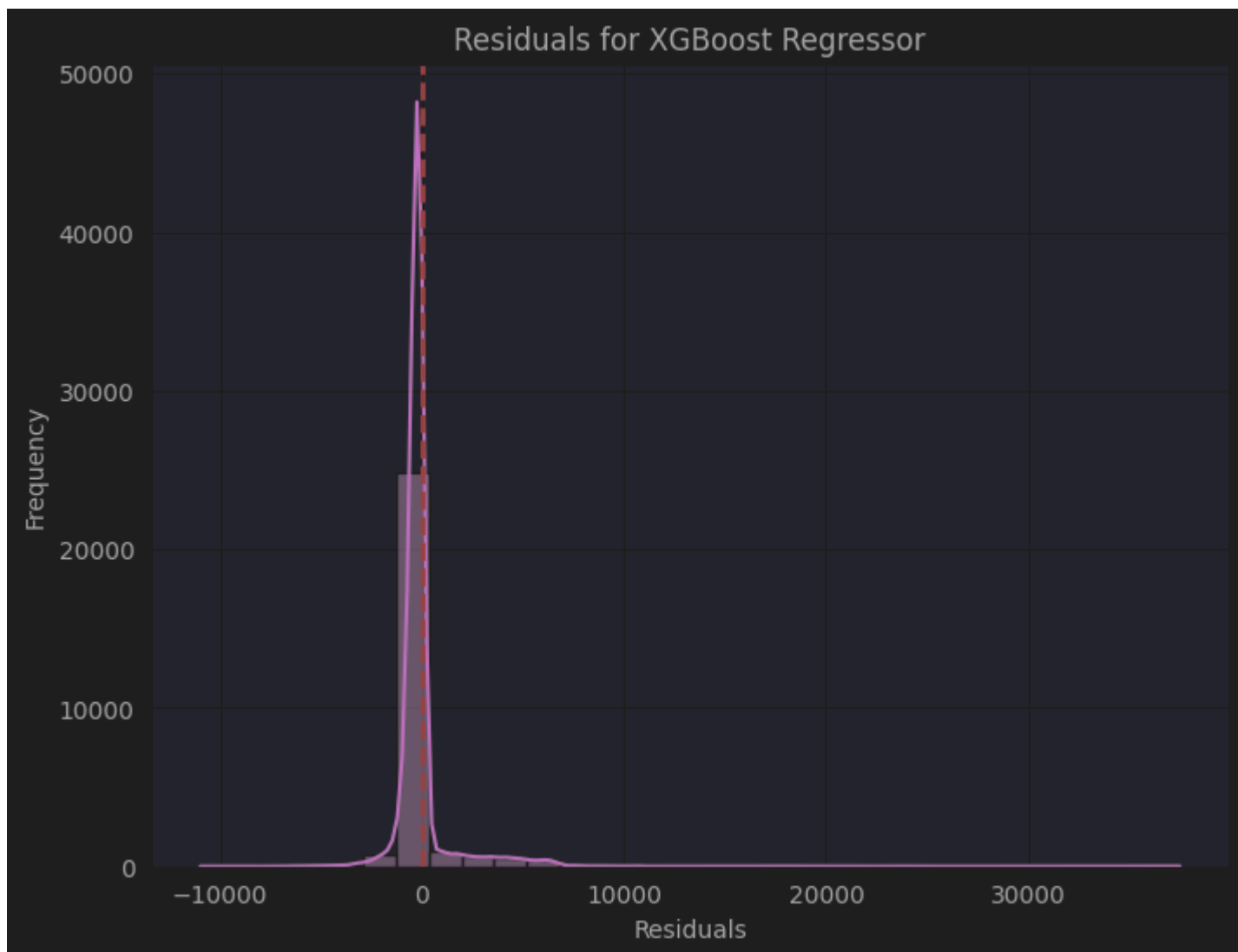
Actual vs Predicted Duration (Random Forest Regressor)





Actual vs Predicted Duration (XGBoost Regressor)





2.4 Моделі для передбачення дистанції дорожніх робіт на основі даних

Основні етапи побудови прогнозової моделі

1. Підготовка та попередня обробка даних

- Виключення нерелевантних ознак, таких як унікальні ідентифікатори (ID, Start_Time, End_Time), оскільки вони не несуть корисної інформації для прогнозування.
- Розділення вибірки на ознаки (X) та цільову змінну (y), де y — це прогнозована відстань (Distance(mi)).
- Розподіл даних на навчальну (80%) та тестову (20%) вибірки для оцінки моделей.

2. Вибір моделей та навчання

Для прогнозування було відібрано три алгоритми:

- Лінійна регресія – базовий метод, що моделює залежність між змінними через лінійне рівняння.
- Random Forest Regressor – ансамблевий метод, який поєднує кілька дерев рішень, що дозволяє виявити складніші залежності між ознаками.
- XGBoost Regressor – градієнтний бустинг, що ітеративно покращує модель, коригуючи помилки попередніх ітерацій.

Кожна модель навчається окремо, а після навчання прогнозує дистанцію дорожніх робіт для тестової вибірки.

3. Оцінка якості передбачень

Для оцінки точності моделей використовуються наступні метрики:

- MAE (Mean Absolute Error) – середня абсолютна похибка, яка показує середнє відхилення передбачених значень від реальних.
- RMSE (Root Mean Squared Error) – корінь середньоквадратичної похибки, що враховує великі помилки з більшою вагою.
- R^2 (коефіцієнт детермінації) – оцінює, наскільки добре модель пояснює варіацію цільової змінної.
- MedAE (Median Absolute Error) – медіанна абсолютна похибка, що менше залежить від викидів.
- Довірчий інтервал – визначає діапазон можливих відхилень прогнозів від реальних значень.

4. Аналіз результатів та вибір найкращої моделі

- Лінійна регресія демонструє найгірші результати, оскільки вона не враховує нелінійні зв'язки між змінними.
- Random Forest забезпечує кращу точність, зменшуючи похибки у порівнянні з лінійною моделлю.
- XGBoost має схожу якість прогнозів із Random Forest, але зазвичай працює швидше та ефективніше для великих обсягів даних.

5. Візуалізація та аналіз похибок

Для кожної моделі будується:

- Графік "фактичне значення vs прогноз" – відображає, наскільки близько передбачені значення до реальних.
- Розподіл залишків (residuals plot) – показує відхилення передбачених значень від реальних, що дозволяє оцінити, чи є систематичні помилки в прогнозах.

Linear Regression:

MAE: 0.5626

RMSE: 1.5621

R^2 : 0.1008

MedAE: 0.3046

95% CI: ± 0.0182

Random Forest Regressor:

MAE: 0.3561

RMSE: 1.2693

R^2 : 0.4063

MedAE: 0.0846

95% CI: ± 0.0148

XGBoost Regressor:

MAE: 0.4090

RMSE: 1.3074

R^2 : 0.3701

MedAE: 0.1410

95% CI: ± 0.0152

Лінійна регресія

- Найгірші показники серед трьох моделей.
- Висока MAE (0.5626) та RMSE (1.5621) означають, що модель робить значні помилки у прогнозах.
- $R^2 = 0.1008$ свідчить про слабку пояснювальну здатність – лише $\approx 10\%$ варіації у відстанях пояснюється ознаками.
- Високий MedAE (0.3046) говорить про суттєві помилки навіть у медіанному випадку.

Random Forest Regressor

- Найкращі результати серед усіх моделей.
- Значно нижчий MAE (0.3561) та RMSE (1.2693) у порівнянні з

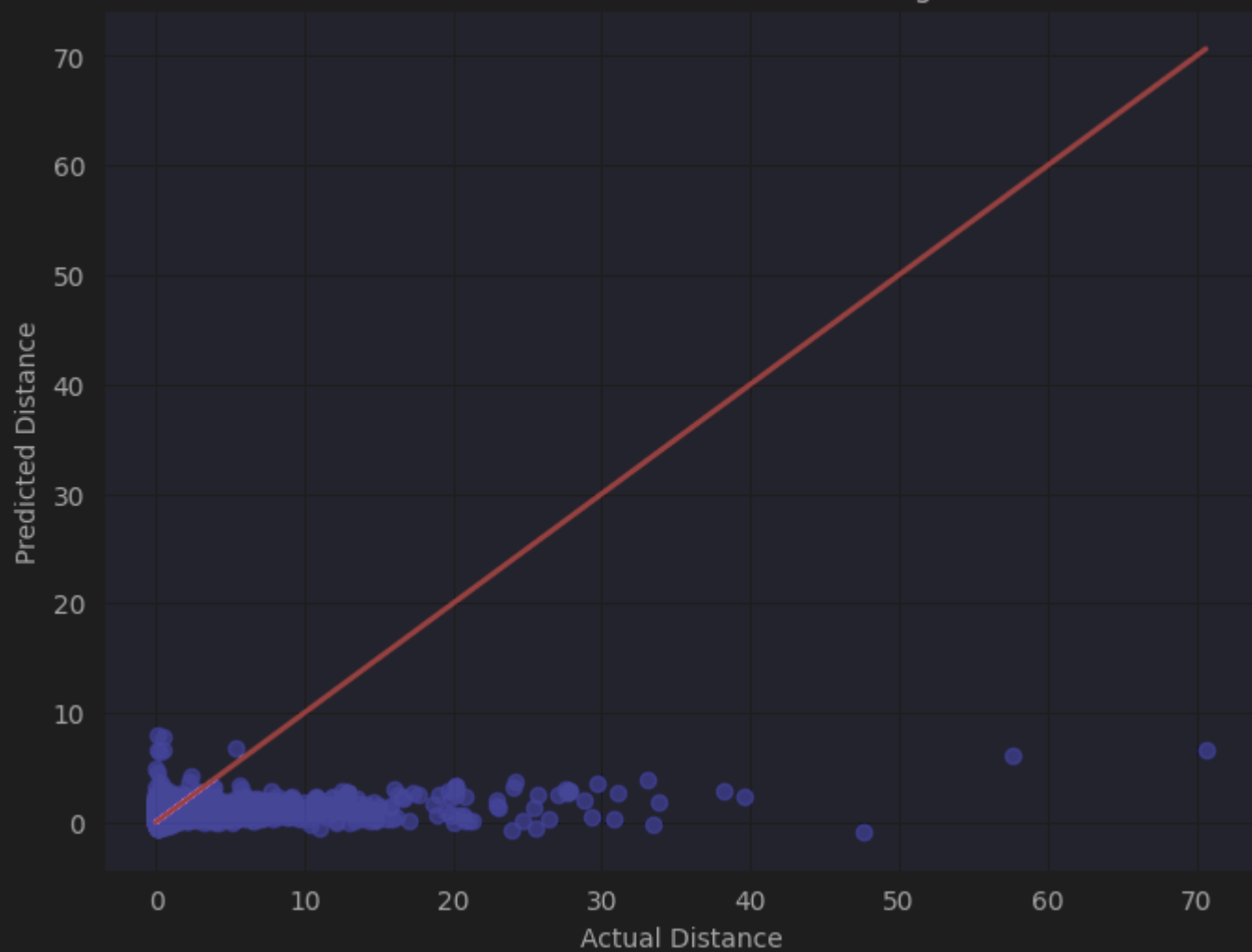
іншими методами.

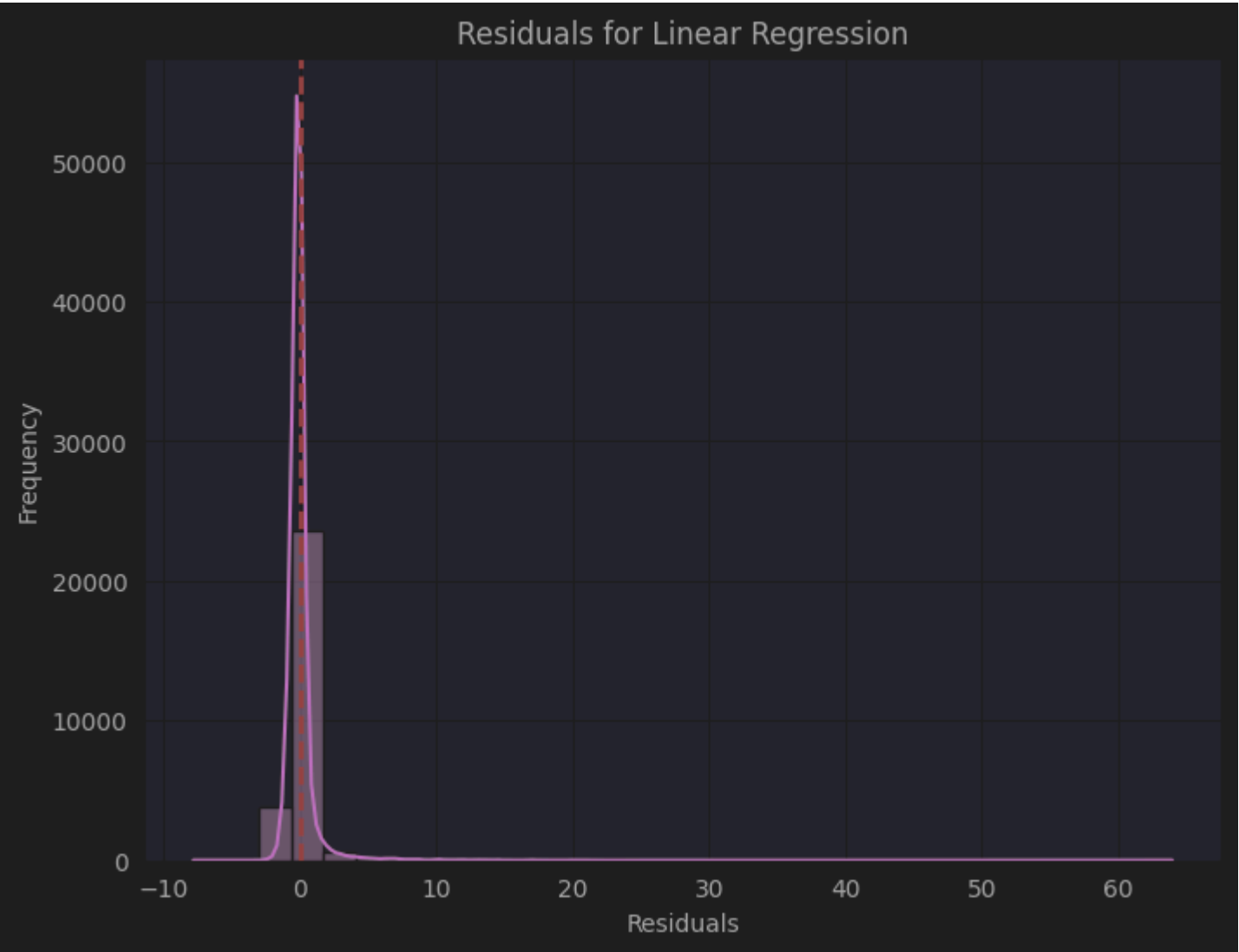
- $R^2 = 0.4063$ означає, що модель пояснює $\approx 40\%$ варіації у даних, що значно краще за лінійну регресію.
- Найменший MedAE (0.0846) показує, що у більшості випадків помилка дуже мала.

XGBoost Regressor

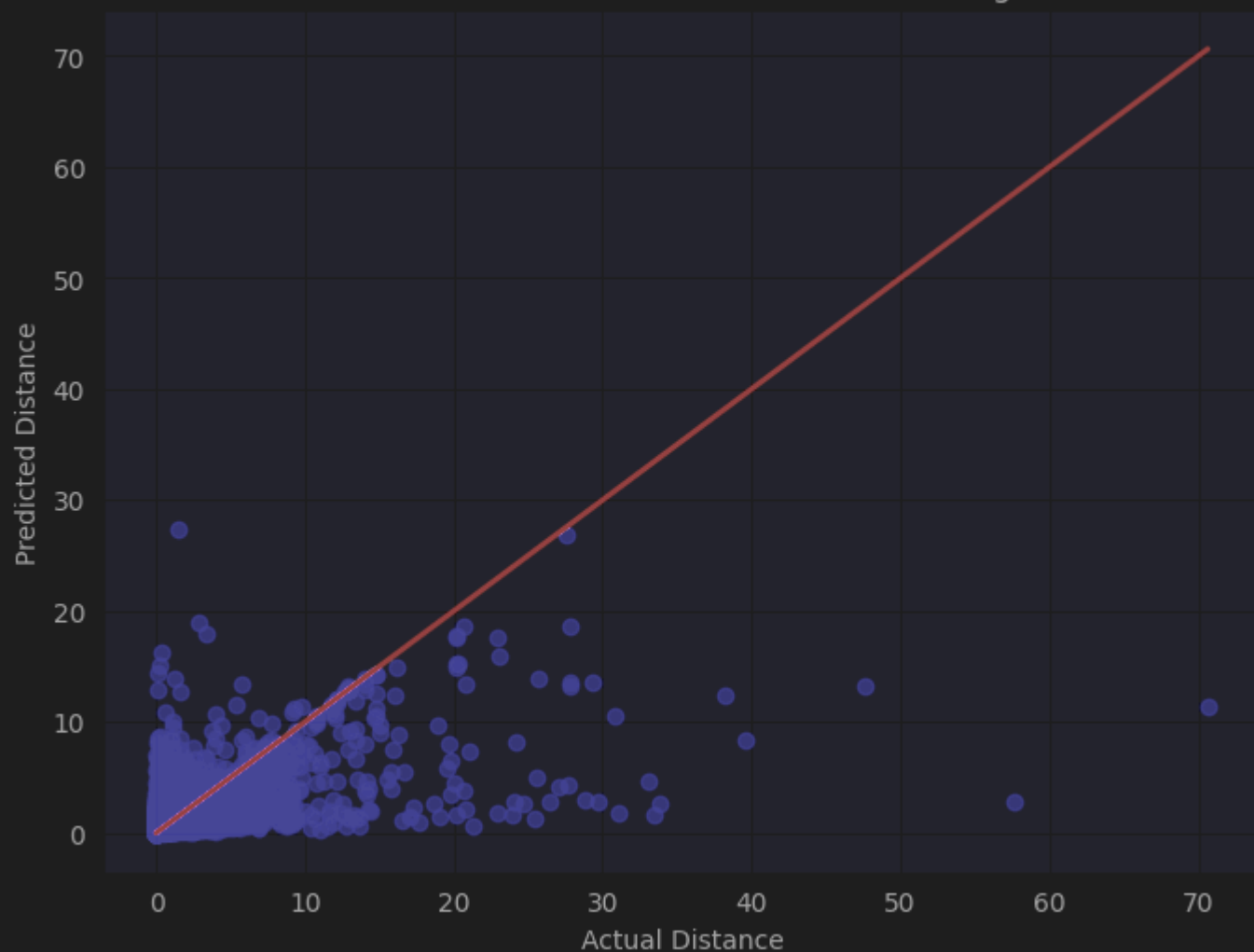
- Результати наближені до Random Forest, але трохи гірші.
- MAE (0.4090) та RMSE (1.3074) дещо вищі, що свідчить про більші середні помилки.
- $R^2 = 0.3701$, тобто модель пояснює $\approx 37\%$ варіації у відстанях, що все ще значно краще за лінійну регресію.
- MedAE (0.1410) вищий за Random Forest, що говорить про наявність більшого числа значних похибок.

Actual vs Predicted Distance (Linear Regression)

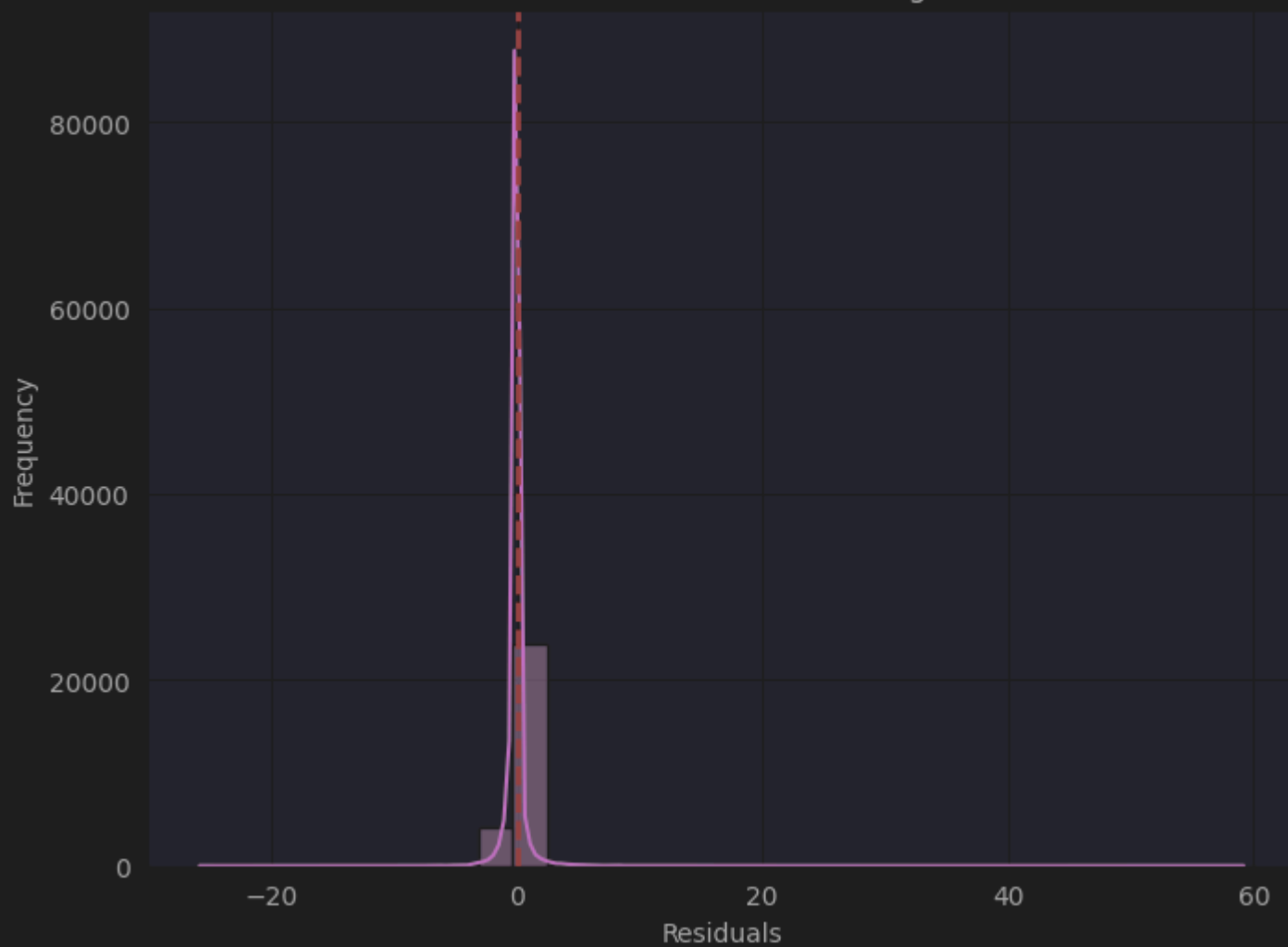




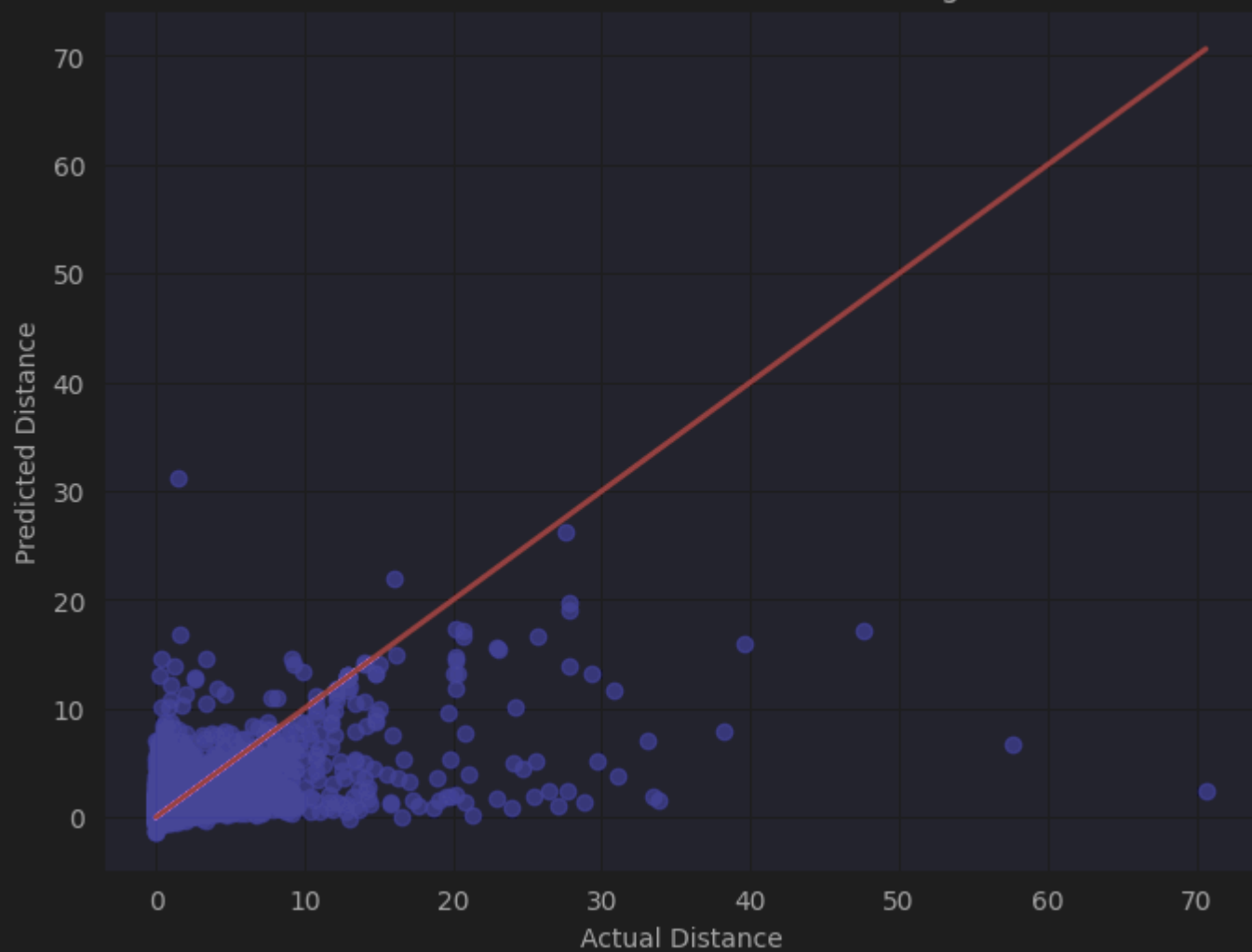
Actual vs Predicted Distance (Random Forest Regressor)

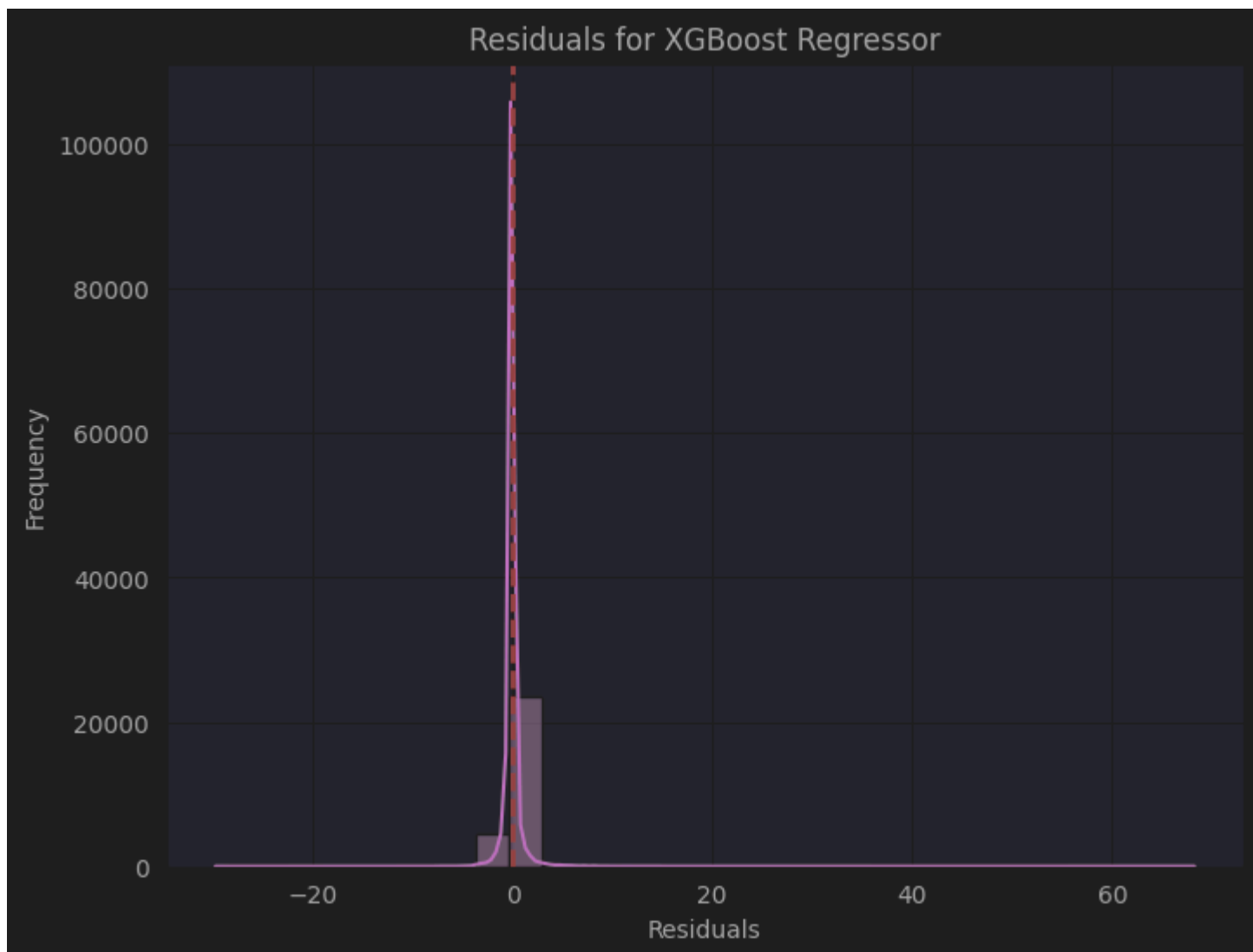


Residuals for Random Forest Regressor



Actual vs Predicted Distance (XGBoost Regressor)





2.5 Пошук асоціативних правил між числовими параметрами

1. Вибір ознак для аналізу

Першим кроком є вибір параметрів, які будуть використовуватися для пошуку асоціативних правил. В даному випадку, це набір числових ознак, таких як:

- Severity (середня важкість події),
- Distance(mi) (відстань у милях),
- Temperature(F) (температура в градусах Фаренгейта),
- Humidity(%) (вологість у відсотках),
- Precipitation(in) (опадів в дюймах),
- Wind_Speed(mph) (швидкість вітру в милях на годину),

- Visibility(mi) (видимість у милях).

Ці ознаки використовуються для того, щоб визначити потенційні асоціативні зв'язки між ними.

2. Дискретизація (перетворення числових даних на категоріальні)

Оскільки пошук асоціативних правил ефективно працює з категоріальними даними, потрібно дискретизувати числові значення. Для кожної з ознак (наприклад, температура, швидкість вітру тощо) застосовується метод бінаризації:

- Числові значення кожної ознаки розподіляються на дві категорії.
- Для цього використовуються функції, які розбивають значення ознаки на два інтервали. Якщо значення потрапляє в верхню частину діапазону, йому присвоюється категорія 1, якщо в нижню — 0.

Цей процес дозволяє отримати для кожної ознаки два значення: чи знаходиться параметр вище або нижче певного порогу. Це дає змогу використовувати бінарні значення для пошуку асоціативних правил, таких як Apriori.

3. Пошук frequent itemsets

Після бінаризації даних, наступним кроком є пошук часто зустрічаються наборів елементів (так званих itemsets). Це групи параметрів, що зустрічаються разом в одному записі (наприклад, подія з високою температурою та низькою швидкістю вітру).

Алгоритм Apriori застосовується для пошуку таких частих наборів елементів з певним мінімальним рівнем підтримки. Підтримка визначає, як часто певна комбінація ознак зустрічається в даних.

4. Генерація асоціативних правил

Після того, як знайдено часто зустрічаються набори елементів, наступним кроком є генерація асоціативних правил. Це дозволяє знайти залежності між різними ознаками. Наприклад, правило може виглядати так: "якщо температура висока, то ймовірність високої швидкості вітру також висока".

Асоціативні правила мають кілька важливих метрик:

- **Support** (підтримка) — скільки разів правило зустрічається в наборі даних.
- **Confidence** (довіра) — наскільки сильна залежність між параметрами в правій та лівій частинах правила.
- **Lift** (підйом) — показує, наскільки правило є сильнішим порівняно з випадковими залежностями.

5. Візуалізація асоціативних правил

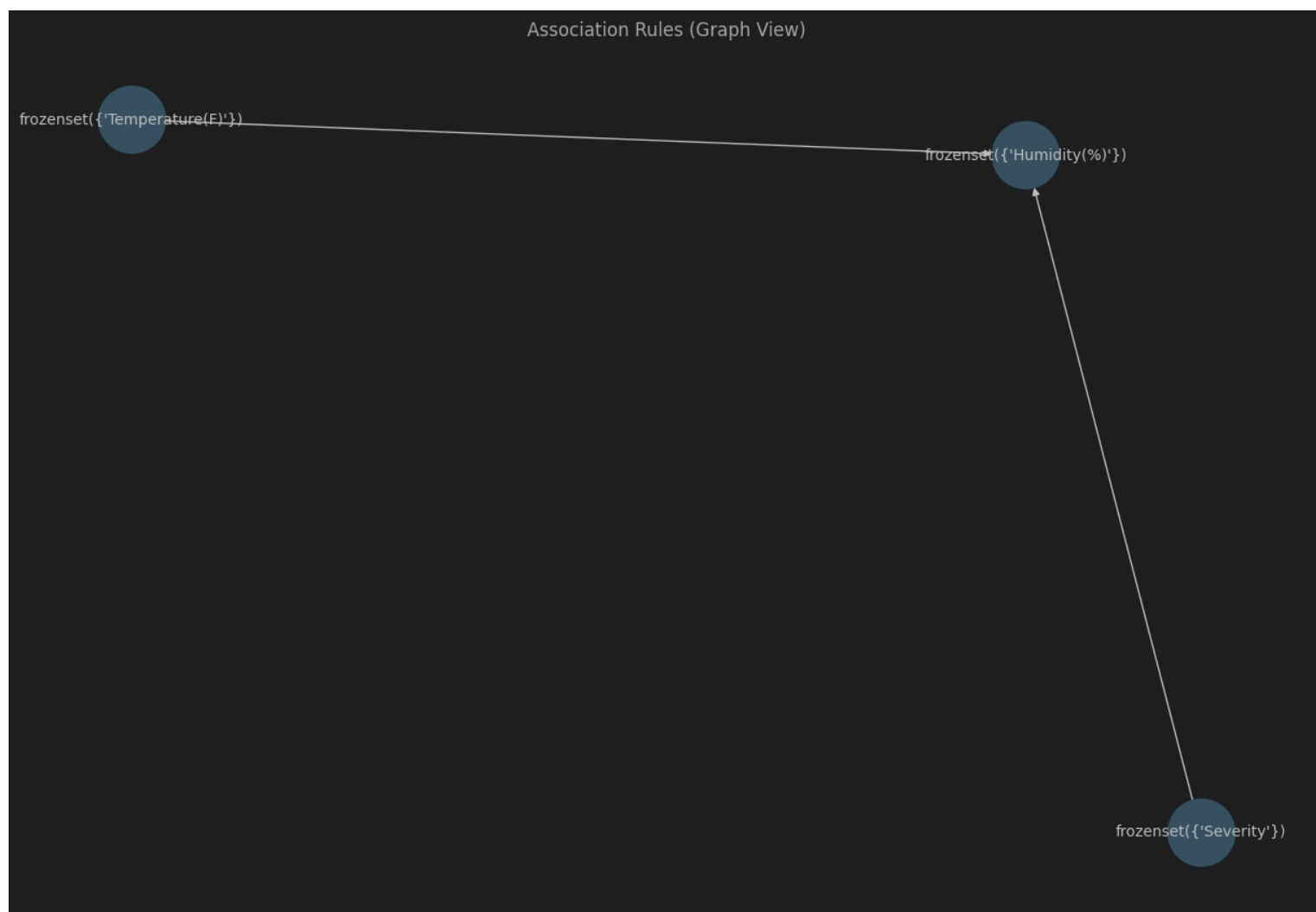
Щоб краще зрозуміти виявлені асоціативні правила, їх можна представити у вигляді графа. У цьому графі:

- **Вершини (вузли)** — це ознаки (наприклад, температура, швидкість вітру).
- **Ребра (зв'язки)** — це виявлені асоціативні правила між ознаками.
- **Вага ребер** — це довіра (confidence) до конкретного правила.

Це дозволяє наочно побачити, які ознаки часто співвідносяться між собою, та визначити ключові патерни в даних.

♦ Discovered Association Rules:

	antecedents	consequents	support	confidence	lift
0	(Severity)	(Humidity(%))	0.07587387	0.77837174	1.17160011
1	(Temperature(F))	(Humidity(%))	0.04659465	0.40508111	0.60972548



Перше правило, що виявлено, вказує на сильний зв'язок між важкістю події та вологістю. Це може означати, що під час більш важких подій (наприклад, серйозних аварій) спостерігається вища ймовірність високої вологості.

Друге правило, що виявлене, вказує на те, що між температурою і вологістю є деякий зв'язок. $Lift < 1$ вказує, на зворотню асоціацію. Якщо значення температури лежить в другій половині відсортованого масиву температур, то значення вологості повітря більш ймовірно буде лежати в першій половині відсортованого масиву значень вологості повітря. Іншими словами, збільшення температури ймовірно провокує зниження вологості та навпаки.

2.6 Пошук асоціативних правил між параметрами POI (Points Of Interest)

Здійснюється аналіз взаємозв'язків між різними типами об'єктів інтересу (POI), які можуть впливати на дорожні умови або події. Ідея полягає в тому, щоб за допомогою асоціативних правил виявити, які типи POI часто зустрічаються разом у певних ситуаціях.

Цей процес включає кілька етапів:

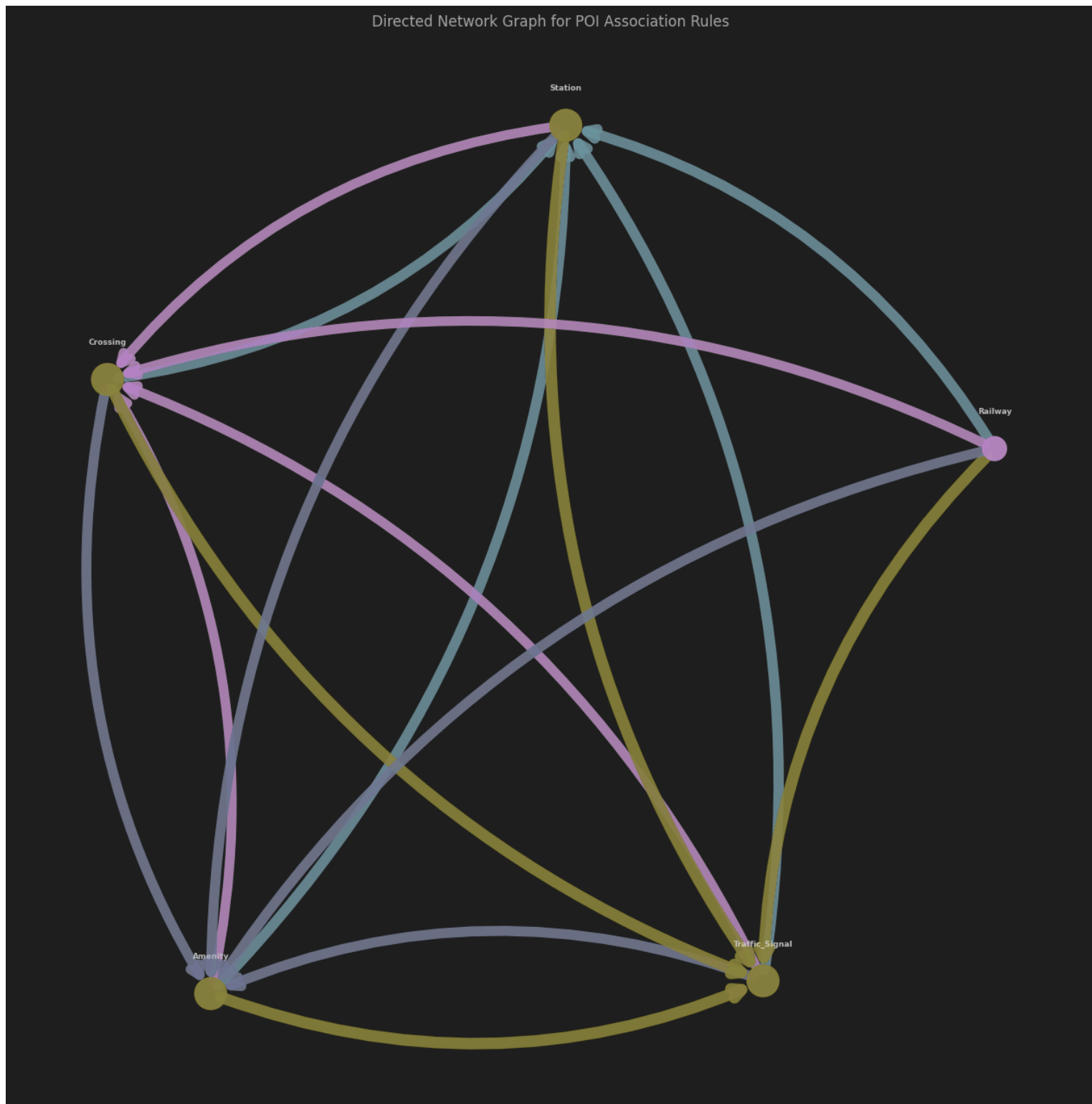
1. Підготовка даних: Вибираються параметри POI, що характеризують різні елементи інфраструктури, такі як "Світлофор", "Знак Стоп", "Перехрестя" тощо. Ці параметри мають бінарні значення, де кожен параметр вказує на наявність чи відсутність певного об'єкта інтересу.
2. Аналіз частоти комбінацій: Шукаються часті комбінації POI, що трапляються разом у даних. Це дозволяє виявити ті групи POI, які часто зустрічаються в одних і тих самих ситуаціях.
3. Виявлення асоціативних правил: Створюються правила, що вказують на наявність певних POI за умовою присутності інших. Наприклад, якщо є "Railway", то ймовірність наявності "Station" може бути високою.
4. Оцінка надійності правил: Для кожного знайденого правила оцінюється його надійність (confidence), що дає змогу визначити, які з правил є найбільш релевантними та значущими.
5. Візуалізація результатів: Для зручності інтерпретації асоціативних правил будуються граф, де вузли представляють різні POI, а ребра — зв'язки між ними. Чим товстіші ребра, тим сильніше асоціативне правило. Додатково, ребра розфарбовані в кольори видимого спектру відповідно до ваги ребра.

Цей підхід дозволяє виявити закономірності в розташуванні та взаємодії різних типів POI, що можуть бути корисні для прогнозування дорожніх ситуацій,

планування інфраструктури чи аналізу дорожнього руху.

◆ Discovered Association Rules for POIs:

	antecedents	consequents	support	confidence	lift
0	(Amenity, Crossing)	(Traffic_Signal)	0.05342506	0.84141067	2.72821617
1	(Amenity, Railway)	(Station)	0.02115314	0.85268237	4.85510321
2	(Amenity, Railway)	(Traffic_Signal)	0.02131510	0.85921090	2.78593220
3	(Station, Railway)	(Traffic_Signal)	0.02278681	0.81204517	2.63300056
4	(Amenity, Crossing, Railway)	(Station)	0.01542827	0.83594048	4.75977625
5	(Amenity, Traffic_Signal, Railway)	(Crossing)	0.01709715	0.80211430	3.09176388
6	(Amenity, Crossing, Railway)	(Traffic_Signal)	0.01709715	0.92636398	3.00367145
7	(Amenity, Crossing, Station)	(Traffic_Signal)	0.02918063	0.88039091	2.85460691
8	(Amenity, Traffic_Signal, Railway)	(Station)	0.01839985	0.86323092	4.91516578
9	(Traffic_Signal, Station, Railway)	(Amenity)	0.01839985	0.80747837	6.48741899
10	(Amenity, Station, Railway)	(Traffic_Signal)	0.01839985	0.86984021	2.82039701
11	(Crossing, Station, Railway)	(Traffic_Signal)	0.01759006	0.89630427	2.90620489
12	(Amenity, Traffic_Signal, Crossing, Railway)	(Station)	0.01473819	0.86202636	4.90830710
13	(Amenity, Traffic_Signal, Station, Railway)	(Crossing)	0.01473819	0.80099502	3.08744960
14	(Traffic_Signal, Crossing, Station, Railway)	(Amenity)	0.01473819	0.83787030	6.73159292
15	(Amenity, Crossing, Station, Railway)	(Traffic_Signal)	0.01473819	0.95527157	3.09740229



Ключові метрики:

- **Support** — показує, наскільки часто комбінація атрибутів з'являється в даних. Більший support означає, що правило є більш поширеним.
- **Confidence** — ймовірність, що при наявності деяких атрибутів з'явиться інший атрибут. Висока confidence вказує на сильний зв'язок.
- **Lift** — показує, наскільки ймовірніше з'являється один атрибут при

наявності іншого, порівняно з випадковим розподілом. Lift більше 1 свідчить про позитивний зв'язок.

Найбільш значимі правила:

1. (Amenity, Crossing) → (Traffic_Signal)

Support: 0.0534, Confidence: 0.8414, Lift: 2.728

- Це правило показує, що коли є "Amenity", і "Crossing", то з великою ймовірністю (84%) буде присутній "Traffic_Signal". Lift 2.73 вказує на сильний позитивний зв'язок між цими атрибутами. Правило має найбільший support (5,3%).

2. (Amenity, Railway) → (Station)

Support: 0.0212, Confidence: 0.8527, Lift: 4.8551

- Це правило демонструє, що наявність "Amenity" і "Railway" вказує на наявність "Station" з дуже високою ймовірністю (85%). Lift 4.86 свідчить про дуже сильний зв'язок.

3. (Amenity, Crossing, Railway) → (Station)

Support: 0.0154, Confidence: 0.8359, Lift: 4.7598

- Це правило має високу ймовірність (83%) наявності "Station" при одночасній наявності "Amenity", "Crossing" та "Railway", і lift вказує на сильний зв'язок (4.76).

4. (Amenity, Traffic_Signal, Railway) → (Station)

Support: 0.0184, Confidence: 0.8632, Lift: 4.9152

- Цей зв'язок між "Amenity", "Traffic_Signal" і "Railway" з наявністю "Station" є дуже сильним, з високою ймовірністю (86%) та lift 4.92, що вказує на дуже потужний зв'язок між атрибутами.

5. (Traffic_Signal, Crossing, Station, Railway) → (Amenity)

Support: 0.0147, Confidence: 0.8379, Lift: 6.7316

- Це правило має надзвичайно високий lift (6.73), що вказує на дуже сильний зв'язок між "Traffic_Signal", "Crossing", "Station" і "Railway" з наявністю "Amenity". Це одне з найбільш значущих правил за lift.

Висновки

В результаті виконання лабораторної роботи були розроблені та застосовані кілька моделей для аналізу та прогнозування параметрів дорожніх робіт на основі машинного навчання. Робота дозволила створити ефективні методи прогнозування серйозності, тривалості та дистанції дорожніх робіт, а також виявити асоціативні правила між різними параметрами, що характеризують ці роботи.

Основні етапи роботи:

- Для прогнозування серйозності дорожніх робіт були розроблені класифікаційні моделі, зокрема логістична регресія, Random Forest Classifier та XGB Classifier. Оцінка ефективності моделей показала добрі результати з високими значеннями метрик точності.
- Для кластеризації даних був використаний алгоритм K-Means, що дозволив виявити залежності між параметрами, такими як температура, швидкість вітру та вологість повітря, що впливають на різні типи дорожніх робіт.
- Для прогнозування тривалості та дистанції дорожніх робіт були застосовані регресійні моделі, зокрема Linear Regression, Random Forest Regressor та XGB Regressor. Моделі показали високу ефективність, що дозволяє точно прогнозувати ці параметри на основі вхідних даних.
- Алгоритм Apriori був використаний для пошуку асоціативних правил між числовими та категоріальними параметрами, що дозволяє аналізувати залежності між різними аспектами дорожніх робіт та умовами навколишнього середовища.

Перспективи розвитку цієї роботи включають подальше вдосконалення моделей, розширення набору параметрів для аналізу, а також інтеграцію з реальними даними для більш точної оцінки ефективності. Крім того, можна розглянути інтеграцію додаткових алгоритмів для класифікації та регресії, що

дозволить забезпечити більш універсальний підхід до аналізу дорожніх робіт.

Розроблений додаток може бути використаний для оптимізації планування та управління дорожніми роботами, а також для аналізу зовнішніх факторів, які впливають на їх ефективність.

Джерела

1. Алгоритм Apriori [Електронний ресурс] - Режим доступу до ресурсу:
<https://medium.com/nuances-of-programming/%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B9-%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1-%D1%80%D0%B5%D1%88%D0%B8%D1%82%D1%8C-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC-apriori-%D1%81-%D0%BD%D1%83%D0%BB%D1%8F-302ec6e7688c>
2. Han, Jiawei; Kamber, Micheline; Pei, Jian (2012). Mining Frequent Patterns, Associations, and Correlations: Basic Concepts and Methods, p 243-278.
3. Hahsler, Michael (2005). "Introduction to arules – A computational environment for mining association rules and frequent item sets" [Електронний ресурс] - Режим доступу до ресурсу:
<https://web.archive.org/web/20190430193743/https://mran.revolutionanalytics.com/web/packages/arules/vignettes/arules.pdf>
4. Wong, Pak (1999). "Visualizing Association Rules for Text Mining" [Електронний ресурс] - Режим доступу до ресурсу:
<https://neuro.bstu.by/ai/Data-mining/Stock-market/InfoVis1999Association.pdf>
5. Mlxtend [Електронний ресурс] - Режим доступу до ресурсу:
<https://rasbt.github.io/mlxtend/>
6. Principal Component Analysis (PCA) [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/principal-component-analysis-pca/>
7. Elbow Method in K-Means Clustering [Електронний ресурс] - Режим доступу до ресурсу: <https://builtin.com/data-science/elbow-method>
8. A Tutorial on Principal Component Analysis [Електронний ресурс] - Режим доступу до ресурсу:
<https://user.eng.umd.edu/~jzsimon/biol708L/ref/ShlensPCATutorial.pdf>
9. Silhouette Score [Електронний ресурс] - Режим доступу до ресурсу:
<https://how.dev/answers/what-is-silhouette-score>

10. Elbow Method for optimal value of k in KMeans [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/>
11. MacQueen (1967). Some methods for classification and analysis of multivariate observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press: 281—297.
12. Silhouette Algorithm to determine the optimal value of k [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.geeksforgeeks.org/silhouette-algorithm-to-determine-the-optimal-value-of-k/>
13. Sklearn library [Електронний ресурс] - Режим доступу до ресурсу:
<https://scikit-learn.org/stable/>
14. XGBoost Regression In Depth [Електронний ресурс] - Режим доступу до ресурсу:
<https://medium.com/@fraidoonomarzai99/xgboost-regression-in-depth-cb2b3f623281>
15. XGBoost Classification In Depth [Електронний ресурс] - Режим доступу до ресурсу:
<https://medium.com/@fraidoonomarzai99/xgboost-classification-in-depth-979f11ef4bf9>
16. Confidence Intervals for Regression Parameters [Електронний ресурс] - Режим доступу до ресурсу: <https://online.stat.psu.edu/stat415/lesson/7/7.5>
17. Сеньо П. С. (2007). Теорія ймовірностей та математична статистика (вид. 2-ге, перероб. і доп.). Київ: Знання. с. 446.
18. Карташов М. В. Імовірність, процеси, статистика. — Київ : ВПЦ Київський університет, 2007. — 504 с.