Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

Використання методів машинного навчання до аналізу Big Data

Текстова частина до курсової роботи
за спеціальністю „Прикладна математика” 113

Керівник курсової роботи
к.т.н., доц. Жежерун О. П
(прізвище та ініціали)

_____
(підпис)

“____” _____ 2022 р.

Виконав студент Поздняков А.О.
(прізвище та ініціали)
“____” _____ 2022 р.

м. Київ – 2022 рік

Тема: Використання методів машинного навчання до аналізу Big Data

Календарний план виконання роботи:

№ п\п	Назва етапу дипломного проекту(роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	Жовтень 2021р.	
2.	Робота з літературою за темою машинного навчання та Big Data	Листопад - березень 2022р.	
3.	Аналіз Big Data та методів машинного навчання	Березень - квітень 2022р.	
4.	Кластеризація датасету та написання текстової частини	Квітень - травень 2022р.	
5.	Перегляд змісту роботи керівником		
6.	Створення презентації		
7.	Захист курсової роботи		

Студент Поздняков А.О. _____

Керівник Жежерун О.П. _____

“_____”_____ 2022 р.

Зміст

ВСТУП

Ми живемо у світі, що є перенасиченим різноманітними даними. Дані накопичуються всюди – смартфони та розумні гаджети відстежують наші переміщення та фізіологічні показники, веб-сайти накопичують відомості про наші дії в мережі Інтернет, системи розумних будинків збирають дані про спосіб життя їх власників, тощо.

В останні роки IT-компанії починають характеризувати себе новим терміном Data-driven, що означає прийняття більшості рішень всередині компанії на основі даних.

Очевидно, що такий напрямок вимагає засобів збору та обробки даних - цим займаються сфери Big Data та Data Science, зокрема машинне навчання (Machine Learning).

Ця робота ставить собі на меті аналіз інструментів Big Data та способів застосування алгоритмів Machine Learning задля отримання корисної інформації з величезних обсягів даних.

РОЗДІЛ 1: Big Data та Data Science

Термін Big Data, що з’явився наприкінці 2000-х років, в період найстрімкішого збільшення обсягу зберігаємих даних, є дуже обширним терміном, оскільки має на увазі абсолютно різні типи даних, які мають одну спільну рису - вони зазвичай є занадто великими, щоб їх можна було обробити традиційними засобами.

Найчастіше основними характеристиками Big Data називають “три V”:

- Volume (Об’єм) - загальна кількість даних
- Variety (Різноманітність) - можливість обробки різних типів даних
- Velocity (Швидкість) - наскільки швидко генеруються або обробляються дані

Саме ці характеристики відрізняють Big Data від звичайних наборів даних, оскільки вони так чи інакше впливають на всі основні аспекти роботи з великими даними, такі як зберігання та отримання нових даних, їх пошук та обробка тощо. Великими даними часто оперують сучасні IT-гіганти (такі як Google або Meta), фінансові, комерційні та навіть урядові компанії

Data Science - це розділ інформатики, що являє собою суміш статистики, Computer Science та штучного інтелекту. Великі об’єми даних містять безліч корисної інформації, яку в майбутньому можна використати. Саме для цього існує сфера Data Science - спеціалісти з аналізу та обробки даних видобувають корисні знання з величезних об’ємів даних за допомогою статистичних методів та моделей машинного навчання.

1.1 Типи даних в Data Science та Big Data

Оскільки однією з основних характеристик Big Data є її різноманітність (Velocity), стає очевидним використання різних типів даних.

Найпоширенішими з них є:

- структуровані дані
- дані природною мовою
- графічні дані

1.1.1 Структуровані дані

Структуровані дані є найпоширенішим типом даних. Виходячи з назви, їх особливість полягає в тому, що вони мають певну структуру, таку, що буде легко сприйматися як людиною, так і комп'ютером - дані зберігаються в фіксованому полі всередині запису. Загальними шаблонами таких даних можна назвати Panel Data або табличні дані. Яскравим прикладом є фінансові звіти, списки студентів, дані покупців деякого інтернет-магазину тощо. Зазвичай структуровані дані зберігаються в базах даних та файлах Excel.

1.1.2 Дані природною мовою

Природна мова - це будь-яка мова, яка виникла у людей природним шляхом використання. До природної мови відносять тексти, звичайне мовлення та навіть спів. Саме це і є даними природною мовою - все, що так чи інакше, базується на мовленні. Одним з основних напрямків Data Science є обробка і аналіз таких даних - NLP (Natural Language Processing). Метою NLP є розуміння моделями машинного навчання сенсу тексту. За допомогою цього моделі здатні класифікувати тексти, визначати їх тематику і навіть генерувати нові. Загалом, дані природною мовою відносять до неструктурованих, але під час аналізу спеціалісти з NLP часто перетворюють ці дані на структуровані, наприклад за

допомогою векторизації - початкові дані перетворюються на таблицю, кожен стовпчик якої відповідає конкретному унікальному слову з усіх, що містилися в початковому наборі, а значення 0 або 1 сигналізують про наявність цього слова в певному реченні, яке відповідає рядку, при цьому порядок слів у реченні не зберігається.

Початкові дані:

```
["Jake likes fish",
 "Andrew likes meat",
 "Sophie likes fish"]
```

Дані після застосування векторизації:

	Andrew	Jake	Sophie	fish	likes	meat
0	0	1	0	1	1	0
1	1	0	0	0	1	1
2	0	0	1	1	1	0

Рисунок 1.1 Приклад застосування векторизації

1.1.3 Графічні дані

До графічних даних відносяться зображення та відео-файли. Роботою з такими файлів займається ще одна область Data Science – Computer Vision. Задачі Computer Vision розбиваються на велику кількість напрямків, найпопулярнішими з яких є розпізнавання образів та тексту класифікація об’єктів, їх сегментація та виявлення. Зазвичай такі дані зберігаються у багатовимірних масивах даних, кількість елементів якого дорівнює кількості пікселів.

1.2 Системи зберігання даних в Big Data

Для великих об’ємів даних необхідні системи, що будуть забезпечувати користувача не тільки безпечним зберіганням, але й можливостями швидкого доступу до даних та запису нових. Ці умови є особливо важливими у великих IT-компаніях, коли доступ необхідний декільком спеціалістам одразу, що створює додаткову напругу на базу даних. Розглянемо найпоширеніші з цих систем.

1.2.1 Реляційні бази даних

Реляційні бази даних, безумовно, є найпопулярнішим типом баз даних. В їх основі лежить реляційна модель даних, тобто вони зберігають інформацію, що є взаємопов’язаною. Для маніпуляцій даними в цих базах даних використовується мова структурних запитів SQL (Structured Query Language). SQL дозволяє отримувати та додавати нову інформацію, створювати та видаляти таблиці тощо. В таблицях таких баз даних строки являють собою новий запис, що має унікальний ідентифікатор, а стовпці є певними атрибутами даних. Саме унікальні ідентифікатори часто є тим полем, за допомогою якого можна з’єднувати між собою таблиці, що мають певне відношення.
Основними перевагами реляційних баз даних є:

- Нормалізація даних - розбиття даних на декілька таблиць запобігає створенню дублікатних або пустих осередків даних
- Незалежність даних - будь-які зміни в структурі бази даних не викликають сильних змін в прикладних програмах

Крім того, називають наступні недоліки:

- Відносно низька швидкість доступу до даних та операції з’єднання
- Складність в масштабуванні бази даних

- Погана підтримка неструктурованих даних

Найпоширенішими системами управління реляційними базами даних є MySQL та PostgreSQL.

1.2.2 NoSQL бази даних

На початку 21 століття великі IT-компанії вирішували проблеми масштабування та паралельної обробки великих обсягів даних, що були наслідком поширення та розвитку мережі Інтернет. В результаті цього було створено ціле сімейство нових систем управління базами даних - NoSQL. Незважаючи на свою оманливу назву, термін NoSQL означає “не тільки SQL” (Not Only SQL). Основною відмінністю NoSQL баз даних від реляційних, як вже було зазначено, є майже необмежена можливість масштабування. Але, крім того, NoSQL бази даних також забезпечують обробку графових, потокових та неструктурованих форм даних.

За останній час було створено велику кількість NoSQl баз даних, але їх можна розподілити на декілька типів:

- Бази даних на основі пар “ключ-значення” - кожному значенню даних відповідає свій ключ, наприклад Year:2022, де ключем є Year, а значенням - 2022. Вагомою перевагою таких баз даних є саме можливість до масштабування, хоча вони й можуть бути складно реалізованими. Прикладом таких баз даних є Amazon DynamoDB, MemchacheDB та Berkeley DB.
- Стовпцеві бази даних - бази даних, в яких дані групуються не за строками, а за стовпцями - дані кожного стовпця зберігаються незалежно від інших стовпців. Важливими властивостями стовпцевих баз даних є гнучкість виконання складних запитів та висока швидкість, оскільки запити виконуються над окремими

стовпцями, а не над всією таблицею одночасно. Прикладами стовпцевих баз даних є Apache Cassandra та Apache HBase.

- Сховища документів - сховища документів зберігають повну інформацію про документ. Таке сховище має схожу структуру з базами даних на основі пар “ключ-значення” - воно також має ключ, але в значенні зберігаються структуровані документи. До сховищ документів належить найпопулярніша NoSQL база даних - MongoDB.
- Графові бази даних - бази даних, які можна зобразити у вигляді графу. Використання графового представлення можна ілюструвати за допомогою соціальних мереж. Вершинами графу позначаються користувачі, а ребрами - взаємини користувачів у рамках мережі. Прикладами графових баз даних є BlazeGraph, Neo4j, AllegroGraph, InfiniteGraph

1.2.3 Розподілені файлові системи

Розподілені файлові системи схожі на звичайні файлові системи, але їх особливість постає в тому, що вони працюють на декількох серверах одночасно. За своїм функціоналом розподілені файлові системи аналогічні до звичайних. Вони так само підтримують усі основні операції: зберігання, читання та видалення даних, і, що не менш важливо - реалізацію засобів безпеки файлів. Найпопулярнішою розподіленою файловою системою є HDFS (Hadoop Distributed File System).

Розглянемо основні переваги розподілених файлових систем на прикладі HDFS:

- Можливість зберігати файли, розмір яких перевищує розмір диску окремого серверу. В HDFS файли складаються з блоків даних, що розподілені між різними серверами.

- Окрім можливості зберігання величезних файлів, такий принцип зберігання даних забезпечує їх безпеку, оскільки кожен блок даних зберігається на декількох машинах.
- Можливість паралельного застосування операцій до одного файлу. Ця перевага, як і попередні, випливає з реплікації файлів на декілька серверів.
- Можливість легкого масштабування. Розподілені файлові системи дуже легко масштабуються як горизонтально, так і вертикально, оскільки користувач завжди може додати або більше серверів до кластеру, або більше ресурсів до вже існуючих (CPU, RAM, HDD).

РОЗДІЛ 2: Машинне навчання

Сфера машинного навчання, що є частиною Data Science, зробила стрибок у своєму розвитку за останні 20 років. Цей термін вперше було введено Артуром Самуелем, дослідником штучного інтелекту, ще в 1959 році, коли він створив програму для гри в шашки, що самостійно вчилася грати. Також до значних подій в історії машинного навчання відносять створення “Тесту Т’юрінгу”, що міг оцінювати рівень інтелекту комп’ютера, Аланом Т’юрінгом, створення персептрону - першої нейронної мережі Френком Розенблатом, перемогу комп’ютера над Гаррі Каспаровим, чемпіоном світу у грі в шахи. Але основним фактором, що сприяв розвитку машинного навчання, був і є технічний прогрес. Стрімке збільшення обчислювальної потужності комп’ютерів дозволило дослідникам отримувати результати підгонки моделей в десятки та сотні разів швидше, а самі моделі робити складнішими та більшими.

Загалом, в машинному навчанні виділяють два головних напрямки:

- навчання з вчителем (supervised learning)
- навчання без вчителя (unsupervised learning)

Розглянемо кожен з них детальніше.

2.1 Навчання з вчителем

Як підказує назва, в процесі навчання присутній “вчитель”, тобто моделі отримують на два датасети, перший - безпосередньо вихідні дані, які моделі треба передбачити, другий - вхідні дані, за допомогою яких модель вчиться визначати вихідні. “Вчителем” в такому процесі є вихідні дані, оскільки вони демонструють моделі, яким повинен бути результат за певних вхідних даних. Навчання з вчителем застосовують для задач регресії та класифікації. Задачі регресії мають на меті передбачення чисельної величини, наприклад передбачення цін на житловий будинок, кількості продуктів для закупівлі в

супермаркеті. Задача класифікації ж є задачею, в результаті якої потрібно віднести об’єкти до певного класу. Прикладом таких задач є класифікація зображень, відстежування шахрайських транзакцій тощо. Процес навчання з вчителем не сильно відрізняється серед моделей - найчастіше він є ітеративним. Перед навчанням для моделі обирається функція втрат, або функція помилки, наприклад середньоквадратична помилка (RMSE, root mean squared error) для задач регресії, або точність (accuracy) для задач класифікації. І, з кожною ітерацією модель навчається на новому наборі даних, обчислює для нього функцію втрат, та коригує свої коефіцієнти. Власне, весь процес навчання з вчителем зводиться до підгонки коефіцієнтів за допомогою мінімізації функції помилки.

2.2 Навчання без вчителя

Навчання без вчителя є протилежністю навчання з вчителем - модель отримує лише вхідні дані та сама намагається визначити вихідні. В цьому процесі “вчитель” повністю відсутній. Хоча навчання без вчителя є дуже обширним напрямком, в основному його використовують для задач кластеризації. Кластеризація - задача групування множини об’єктів на окремі підмножини (кластери) за певними ознаками. Яскравим прикладом використання кластеризації є групування користувачів задля отримання корисної інформації про користувачів з кожного кластеру. В цьому випадку відсутність “вчителя” є причиною відсутності функції помилки в процесі навчання. Навіть враховуючи такі метрики, як оцінка силуету (Silhouette score), якість кластеризації зазвичай оцінюється спеціалістом з аналізу даних, виходячи з його власного досвіду.

2.3 Алгоритми та моделі машинного навчання

2.3.1 Лінійна регресія

Лінійна регресія, безумовно, є найпопулярнішим алгоритмом машинного навчання. Вона задається наступним рівнянням:

$$Y = \alpha + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

Рисунок 2.1 Рівняння лінійної регресії

В цьому рівнянні Y відповідає вихідним даним, зміні (x_1, x_2, ..., x_n) відповідають вхідним даним, (w_1, w_2, ..., w_n) - коефіцієнти, які модель має підібрати, α − зміщення. Коефіцієнти моделі змінюються за допомогою ітеративного алгоритму градієнтного спуску. Цей алгоритм обчислює градієнт функції втрат та змінює коефіцієнти моделі у напрямку градієнту. Таким чином, через певну кільку кроків, алгоритм знаходить локальний або глобальний мінімум функції втрат та підганяє коефіцієнти моделі. Найчастіше для лінійної регресії використовують такі функції втрат як RMSE (root mean squared error) та MAE (mean absolute error).

Лінійна регресія також має аналітичне рішення:

$$w = (X^T X)^{-1} X^T Y$$

Рисунок 2.2 Аналітичне рішення лінійної регресії

В цьому рівнянні w - матриця коефіцієнтів, матриці X та Y - вхідні та вихідні дані відповідно.

Для алгоритму лінійної регресії найчастіше застосовуються два поширених методи регуляризації - L1 та L2. Ці методи штрафують модель за її зростаючу складність. Вони обидва знижають значення коефіцієнтів, але їх відмінністю є

той факт, що L1 може робити значення коефіцієнтів нульовими, тобто взагалі не враховувати деякі поля вхідного датасету, коли ж L2 не може робити коефіцієнти моделі нульовими. Моделі лінійної регресії, в яких застосовуються L1 та L2 регуляризації називають лассо регресією (Lasso Regression) та гребеневою (Ridge Regression) регресією відповідно.

При роботі з лінійною регресією необхідно охайно формувати вхідний датасет, щоб в нього не потрапили лінійно залежні або змінні, що утворюють лінійну комбінацію інших змінних. Мультиколінеарність негативно впливає на якість передбачень, які робить алгоритм.

Лінійна регресія використовується для передбачення чисельних даних, таких як ціна на певний товар, кількість продуктів, що потрібно закупити магазину тощо.

2.3.2 Логістична регресія

Іншим популярним алгоритмом машинного навчання є логістична регресія. Логістична регресія, на відміну від лінійної, застосовується тільки для задач класифікації. Загалом, вона є схожою на лінійну регресію, але до неї також застосовується логістична функція, з якої випливає назва самого алгоритму.

$$f(z) = \frac{1}{1+e^{-z}}$$

Рисунок 2.3 Рівняння логістичної регресії

Ця функція гарантує, що вихідне значення буде лежати на проміжку [0; 1], що інтерпретується, як ймовірність належності об’єкту до одного з бінарних класів. Якщо ця ймовірність більша за 0,5 - алгоритм відносить об’єкт до класу 1, якщо ж ця ймовірність менша за 0,5 - до класу 0.

Як і до лінійної регресії, до логістичної регресії можуть застосовуватися методи L1 та L2 регуляризації.

2.3.3 K-Means

K-Means є найпростішим й найпоширенішим алгоритмом кластеризації. Він є ітеративним і обов'язково сходиться. Кількість кластерів для алгоритму повинна заздалегідь бути задана користувачем. Зазвичай для цього використовується метод ліктя (Elbow Method) - K-Means запускається на декількох варіантах кількості кластерів, для кожного з цих варіантів обчислюється сума квадратів відстаней всередині кластерів. Оптимальним варіантом кількості кластерів є "лікоть" на графіку, що є різким переломом у фігурі.

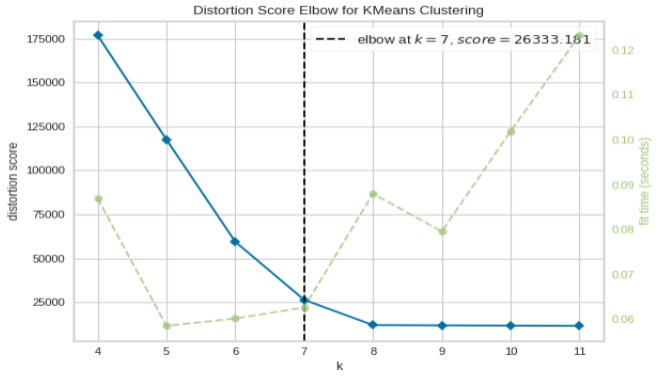

Рисунок 2.4 Візуалізація методу ліктя

Розглянемо цей алгоритм покроково:

1. Центри кластерів визначаються випадковим чином.
2. Для кожного об'єкту обчислюється відстань (зазвичай використовується Евклідова відстань) до всіх центрів кластерів.

3. Об'єкти присвоюються до кластерів, відстань до яких є найменшою.

4. Центри кластерів перераховуються відповідно до нових об’єктів, що було додано до кластерів.
5. Якщо внутрішньокластерні відстані не змінилися - алгоритм завершує роботу. В противному випадку алгоритм повертається до кроку №2.

Результатом роботи K-Means є номери кластерів, до яких алгоритм відніс кожен з об’єктів вхідного датасету. При чому ці номери можуть змінюватися при кожному запуску алгоритму, оскільки початкові центри кластерів визначаються випадковим чином.

РОЗДІЛ 3: Кластеризація датасету

Зробимо кластеризацію датасету, що містить в собі дані про мобільні телефони за 2021 рік - діапазон їх цін, характеристики та загальна популярність. Нашою метою буде розподілення телефонів на декілька цінових категорій, що будуть визначатися їх характеристиками - розміром екрану, обсягом пам’яті та батареї.

Для початку, перевіримо, чи корелюють характеристики мобільних телефонів з ціною. Як можемо побачити, обсяг пам’яті та розмір екрану мають лінійний взаємозв’язок з ціною.

	best_price	screen_size	memory_size	battery_size
best_price	1.000000	0.334798	0.672735	-0.085302
screen_size	0.334798	1.000000	0.337826	0.400950
memory_size	0.672735	0.337826	1.000000	0.016797
battery_size	-0.085302	0.400950	0.016797	1.000000

Рисунок 3.1 Кореляції змінних датасету

Для цієї задачі ми будемо використовувати алгоритм K-Means. Визначимо кількість кластерів за допомогою Elbow method. Для цього використаємо клас KElbowVisualizer бібліотеки yellowbrick.

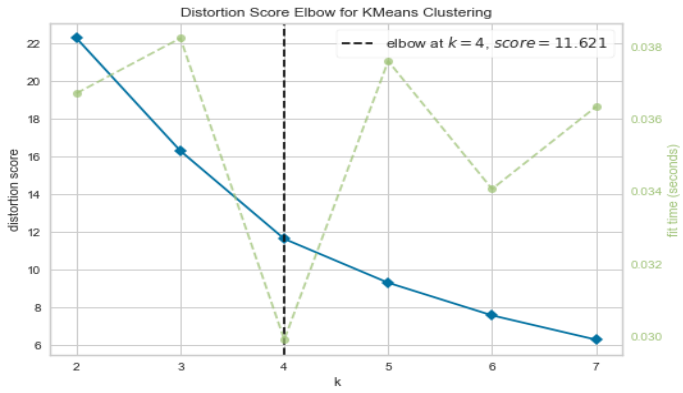

Рисунок 3.2 Візуалізація методу ліктя застосованого на датасеті

Як можна побачити на рисунку, Elbow method визначає кількість кластерів рівне 4. Виконаємо підгонку K-Means до нашого датасету для 4 кластерів.

```
Cluster's 0 average price (UAH): 6943.74 ; Cluster size : 267
Cluster's 1 average price (UAH): 10856.8 ; Cluster size : 460
Cluster's 2 average price (UAH): 6036.81 ; Cluster size : 21
Cluster's 3 average price (UAH): 30734.75 ; Cluster size : 32
```

Рисунок 3.3 Характеристики сформованих кластерів

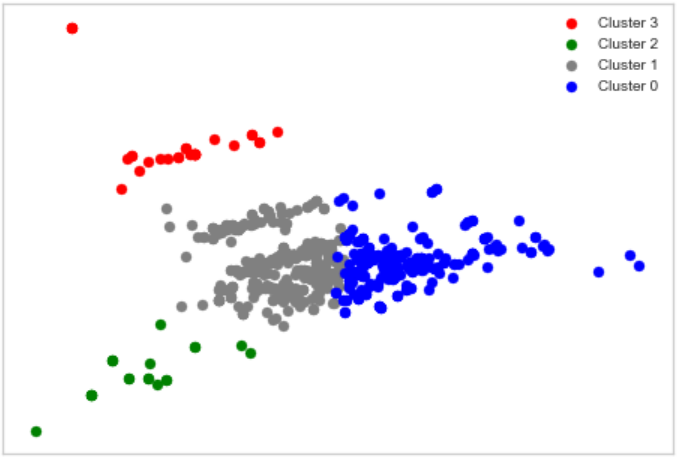

Рисунок 3.4 Візуалізація двомірної проекції датасету

В результаті кластери 1, 3 можна охарактеризувати як кластери з середньою та високою цінами відповідно. Але кластери 0 та 1 мають схожу низьку ціну, крім того, на двовимірній проекції нашого датасету видно, що ці кластери виглядають, як дві половини спільної області точок.

Виконаємо другу ітерацію кластеризації нашого датасету, але зменшимо кількість кластерів до 3.

```
Cluster's 0 average price (UAH): 7716.84 ; Cluster size : 307
Cluster's 1 average price (UAH): 10444.01 ; Cluster size : 441
Cluster's 2 average price (UAH): 30734.75 ; Cluster size : 32
```

Рисунок 3.5 Візуалізація двомірної проекції датасету

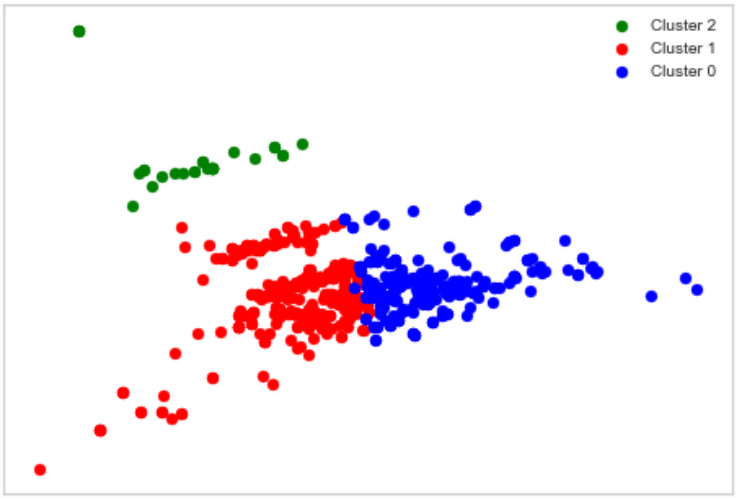

Рисунок 3.6 Візуалізація двомірної проекції датасету

На відміну від першої ітерації, ми отримали кластери, що гарно відрізняються один від іншого за середньою ціною. Метою цієї задачі було розподілення телефонів на декілька цінових категорій, що будуть визначатися їх характеристиками - отже, можемо охарактеризувати отримані кластери наступним чином:

- кластер 0 - бюджетні мобільні телефони
- кластер 1 - мобільні телефони середнього цінового діапазону
- кластер 2 - дорогі мобільні телефони

Висновок

Під час роботи було проведено обширний аналіз сфери Big Data та Data Science. Було детально розглянуто типи даних та засоби їх зберігання - SQL та NoSQL бази даних, розподілені файлові системи. Також було досліджено популярні методи машинного навчання, які використовуються для аналізу цих даних.

В останній частині роботи було проведено кластеризацію датасету з характеристиками та цінами мобільних телефонів, завдяки чому ми змогли об’єднати дані у декілька схожих між собою категорій.

Список використаних джерел

1. Joel Grus, *Data Science from Scratch: First Principles with Python*, 2015
2. Ankur A. Patel, *Hands-On Unsupervised Learning Using Python: How to Build Applied Machine Learning Solutions from Unlabeled Data*, 2019
3. Andrew Bruce, Peter Bruce, *Practical Statistics for Data Scientists: 50 Essential Concepts*, 2017
4. Wes McKinney, *Python for Data Analysis*, 2012
5. Davy Cielen, Arno Meysman, Mohamed Ali, *Introducing Data Science: Big Data, Machine Learning, and more, using Python tools*, 2016