Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДУ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ВЕЛИКИХ ОБСЯГІВ ДАНИХ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-19-2
Пригодій А.І.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Кобилін О.А.
посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Пригодій Анні Ігорівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та реалізація методу машинного навчання для класифікації великих обсягів даних

затверджена наказом по університету від «_23_» жовтня 2020 року № 1428Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 листопада 2020 р.

3. Вихідні дані до роботи *набір тестових даних, байєсівський класифікатор, теоретичні відомості про баєсівський класифікатор, середовище розробки Power BI*

4. Перелік питань, що потрібно опрацювати в роботі

1. Дослідження методів класифікації великих обсягів даних у машинному навчанні та їх аналіз
2. Огляд сучасних сервісів для машинного навчання
3. Розробка методу для класифікації великих обсягів даних
4. Обробка та аналіз отриманих результатів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) *Актуальність проблеми класифікації великих обсягів даних, постановка задачі, тестові дані*

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	23.10.2020	
2	Аналіз завдання, підбір літератури	24.10.20-25.10.20	
3	Аналіз літератури з досліджуваної проблеми	26.10.20-27.10.20	
4	Аналіз технічних засобів	28.10.20-29.10.20	
5	Розробка методу	30.10.20-05.11.20	
6	Програмна реалізація	06.11.20-16.11.20	
7	Оформлення пояснювальної записки	17.11.20-25.11.20	
8	Перевірка на плагіат	26.11.20	
9	Рецензування	27.11.20	
10	Підготовка презентації та доповіді	28.11.20	
11	Занесення роботи в електронний архів	29.11.20	
12	Попередній захист атестаційної роботи	30.11.20	

Дата видачі завдання 23 жовтня 2020 р.

Студент _____
(підпис)

Керівник роботи ___ ___доц. Кобилін О.А._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка атестаційної роботи: 72 с., 19 рис., 40 джерел.

МАШИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ, ВЕЛИКІ ДАНІ, R.

Об’єктом дослідження даного дослідження є методи машинного навчання.

Метою даного дослідження є дослідження та реалізація методу, що базується на Байєсівському класифікаторі для великих обсягів даних у формі часових рядів.

Робота присвячена дослідженню класифікації великих обсягів даних. Розглянуто останні тенденції у машинному навчанні, описано основні методики навчання систем. Проведено огляд сучасних сервісів для машинного навчання. Також оглянуто та проаналізовано сучасні методи машинного навчання для класифікації даних.

В рамках дослідження було реалізовано метод класифікації для великих обсягів даних основі байєсівського класифікатора.

MACHINE LEARNING, CLASSIFICATION, BIG DATA, R.

The object of study of this study are the methods of machine learning.

The purpose of this study is to study and implement a method based on the Bayesian classifier for large amounts of data in the form of time series.

The work is devoted to the study of the classification of large amounts of data. The latest trends in machine learning are considered, the main methods of learning systems are described. A review of modern services for machine learning. Modern machine learning methods for data classification are also reviewed and analyzed.

The study implemented a classification method for large amounts of data based on the Bayesian classifier.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

МН – машинне навчання

ВД – великі данні

ЧР – часові ряди

ОС – операційна система

CSV – comma-separated language

XML – extensible markup language

ВСТУП

На сьогоднішній день, великі дані є одним з двигунів розвитку інформаційних технологій. Це пов’язано з тим, що по всім користувачам інтернету стало накопичуватися величезна кількість інформації.

Термін Великі дані викликає безліч розбіжностей, багато хто припускає, що це тільки обсяг накопиченої інформації, але також не потрібно забувати і про технічну сторону, даний напрямок включає в себе технології зберігання, обчислення, а також сервісні послуги [1].

Сфера використання технологій Великих даних величезна. Наприклад, за допомогою Великих даних можна дізнатися про переваги клієнтів, про ефективність маркетингових кампаній або провести аналіз ризиків. Але найбільш популярне їх використання помічено в торгівлі, охороні здоров’я, телекомунікації, в фінансових компаніях, а також в державному управлінні.

При використанні даної технології в роздрібних магазинах можна накопичити безліч інформації про клієнтів, системі управління запасами, постачання товарної продукції. За допомогою отриманої інформації можна прогнозувати попит або поставки товару, а також оптимізувати витрати.

У фінансових компаніях великі дані надають можливість проаналізувати кредитоспроможність позичальника, тобто на основі виявленого обороту грошових коштів підібрати вигідні та оптимальні умови кредитування, запропонувати додаткові відповідні йому банківські послуги. Застосування такого підходу дозволить значно скоротити час розгляду заявок.

Оператори стільникового зв’язку також, як і фінансові організації, мають величезні бази даних, що дозволяє їм проводити детальний аналіз накопиченої інформації. Крім використання Великі дані з метою надання якісних послуг технологію можна застосувати для виявлення та запобігання шахрайству.

Всі вище перераховані застосування технології великих даних потребують класифікації. Це робить дані більш структурованими та полегшує подальшу роботу з ними.

Таким чином, задача класифікації великих даних є актуальним завданням машинного навчання. Тому ставиться завдання розробки методу класифікації великих даних на основі байєсівського класифікатора.

1 ОГЛЯД МЕТОДІВ ТА ОСОБЛИВОСТІ ОБРОБКИ ВЕЛИКИХ ДАНИХ

1.1 Теоретичні відомості про великі дані

Великі дані – це структуровані і неструктуровані дані величезних обсягів і різноманітності, а також методи їх обробки, які дозволяють розподіллено аналізувати інформацію [2].

Щоб масив інформації позначити приставкою «великі» він повинен мати такі ознаки, що описані на рисунок 1.1.

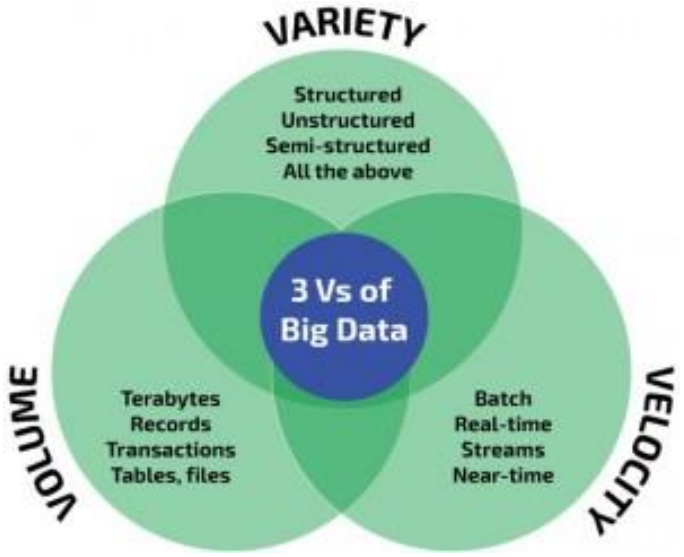


Рисунок 1.1 – Ознаки великих даних

Правило VVV [3]:

– обсяг (Volume) – дані вимірюються по фізичній величині і займаному простору на цифровому носії. До «біг» відносять масиви понад 150 Гб на добу;

– швидкість, оновлення (Velocity) – інформація регулярно оновлюється і для обробки в реальному часі необхідні інтелектуальні технології великих даних;

– різноманітність (Variety) – інформація в масивах може мати неоднорідні формати, бути структурованою частково, повністю і

накопичуватися безсистемно. Наприклад, соціальні мережі використовують великі дані у вигляді текстів, відео, аудіо, фінансових транзакцій, картинок і іншого.

У сучасних системах розглядаються два додаткових фактори:

– мінливість (Variability) – потоки даних можуть мати піки і спади, сезонності, періодичність. Сплески неструктурованою інформації складні в управлінні, вимагає потужних технологій обробки;

– значення даних (Value) – інформація може мати різну складність для сприйняття і обробки, що ускладнює роботу інтелектуальним системам. Наприклад, масив повідомлень з соцмереж – це один рівень даних, а транзакційні операції – інший. Завдання машин визначити ступінь важливості інформації, що надходить, щоб швидко структурувати.

Принцип роботи технології великих даних заснований на максимальному інформуванні користувача про який-небудь предмет або явище. Завдання такого ознайомлення з даними – допомогти зважити всі «за» і «проти», щоб прийняти вірне рішення [4]. В інтелектуальних машинах на основі масиву інформації будується модель майбутнього, а далі імітуються різні варіанти і відслідковуються результати.

До джерел великих даних відносять:

– інтернет-блоги, соцмережі, сайти, ЗМІ та різні форуми;

– корпоративну інформацію – архіви, транзакції, бази даних;

– показання зчитувальних пристроїв – метеорологічні прилади, датчики стільникового зв’язку та інші.

Самі по собі алгоритми великих даних виникли при впровадженні перших високопродуктивних серверів, що володіють достатніми ресурсами для оперативної обробки інформації та придатних для комп’ютерних обчислень і для подальшого аналізу [5].

Сам термін Big Data вперше був озвучений в 2008 році на сторінках спецвипуску журналу Nature в статті головного редактора Кліффорда Лінча.

Цей номер видання був присвячений вибухового зростання глобальних обсягів даних (рис. 1.2) і їх ролі в науці.

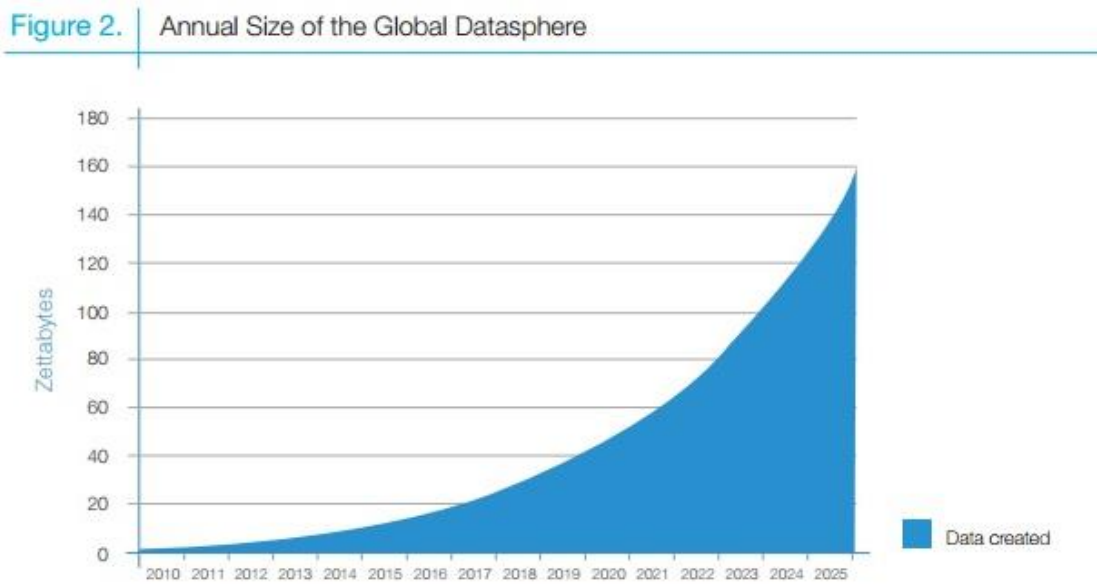


Рисунок 1.2 – Прогнозований графік зростання обсягів даних в світі до 2025 року за даними компанії Seagate

1.2 Принципи роботи з великих даних

Принципи роботи з масивами даних включають три основні чинники:

– можливість розширення системи. Під нею розуміють зазвичай горизонтальну масштабованість носіїв інформації [6]. Тобто зросли обсяги вхідних даних – збільшилися потужність і кількість серверів для їх зберігання;

– стійкість до відмови. Підвищувати кількість цифрових носіїв, інтелектуальних машин пропорційно обсягам даних можна до нескінченності. Але це не означає, що частина машин не буде виходити з ладу, застарівати. Тому одним із чинників стабільної роботи з великими даними є відмовостійкість серверів;

– локалізація. Окремі масиви інформації зберігаються і обробляються в межах одного виділеного сервера, щоб економити час, ресурси, витрати на передачу даних.

Всі сучасні засоби роботи з великими даними слідують цим трьом принципам. Для того, щоб їх дотримуватися – необхідно придумувати якісь методи, способи і парадигми розробки засобів розробки даних [7].

Для збору і обробки Великих Даних використовуються різні технології, основні з яких:

– масова паралельна обробка (MPP);

– MapReduce – обчислювальна парадигма, запропонована компанією Google;

– обробка складних подій – онлайн-обробка інформації з різних джерел, що залежить від часу;

– Hadoop – проект фонду Apache Software Foundation, який реалізує парадигму MapReduce;

– RDBMS;

– Cassandra – альтернатива для Hadoop HDFS, база даних, виконана як NoSQL;

– Hive – файлове сховище, створене компанією Facebook;

– NoSQL.

1.3 Принцип роботи MapReduce

MapReduce – це модель розподіленої обробки даних, запропонована компанією Google для обробки великих обсягів даних на комп’ютерних кластерах (рис. 1.3).

MapReduce передбачає, що дані організовані у вигляді деяких записів. Обробка даних відбувається в 3 стадії:

– Стадія Map. На цій стадії дані проходять попередню обробку за допомогою функції map(), яку визначає користувач. Робота цієї стадії полягає в попередній обробці і фільтрації даних. Робота дуже схожа на операцію map в функціональних мовах програмування призначена для користувача функція застосовується до кожної вхідного відрізку.

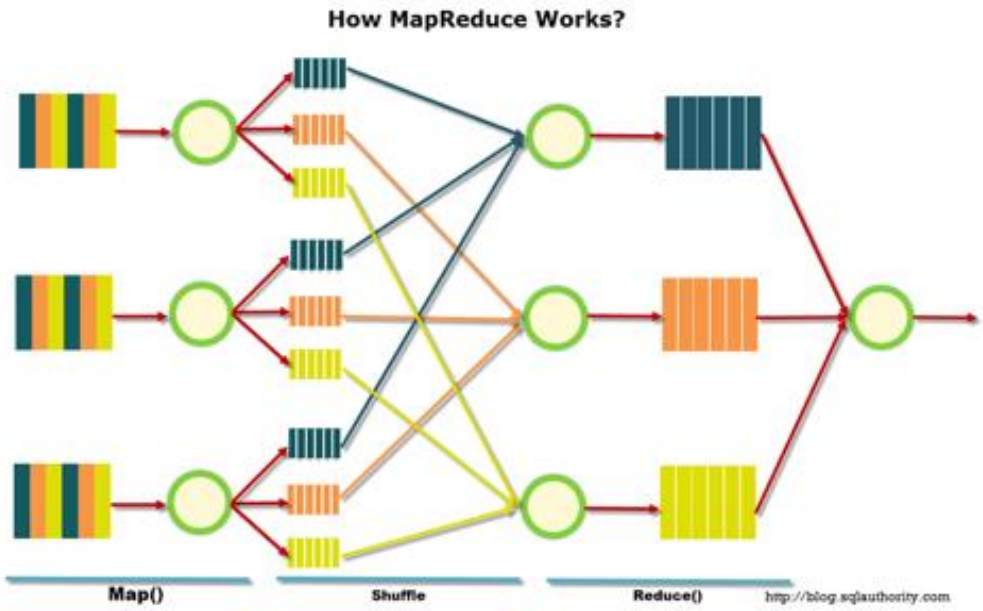


Рисунок 1.3 – Принцип роботи MapReduce

Функція map() застосована до одного вхідного запису і видає множину пар ключ-значення. Множина може видати тільки один запис, може не видати нічого, а може видати кілька пар ключ-значення [8]. Що буде знаходиться в ключі і в значенні – вирішувати користувачу, але ключ дуже важлива річ, так як дані з одним ключем в майбутньому потраплять в один екземпляр функції reduce;

– Стадія Shuffle. Проходить непомітно для користувача. У цій стадії висновок функції map «розбирається по кошиках» – кожен кошик відповідає одному ключу виведення стадії map. Надалі ці кошики будуть служити входом для reduce;

– Стадія Reduce. Кожен «кошик» із значеннями, сформована на стадії shuffle, потрапляє на вхід функції reduce().

Функція reduce задається користувачем і обчислює фінальний результат для окремої «кошика». Множина всіх значень, повернутих функцією reduce(), є фінальним результатом MapReduce-завдання.

1.4 Методики аналізу та обробки великих даних

Під аналізом великих даних розуміється як аналіз масивів даних в рамках можливостей персонального комп'ютера, так і в рамках можливостей систем керування базами даних, при цьому як в першому, так і в другому випадку при формуванні статистики і візуалізації виникають певні труднощі, які полягають в необхідності забезпечення скоординованої роботи комп'ютерних програм на десятках, сотнях або навіть тисячах серверів [9].

До основних способів аналізу великих масивів інформації відносять такі:

– глибинний аналіз, класифікація даних. Ці методики прийшли з технологій роботи зі звичайною структурованою інформацією в невеликих масивах. Однак в нових умовах використовуються вдосконалені математичні алгоритми, засновані на досягненнях в цифровій сфері;

– краудсорсінг. В основі цієї технології можливість отримувати і обробляти потоки в мільярди байт з багатьох джерел. Кінцеве число «постачальників» не обмежується нічим. Хіба тільки потужністю системи;

– спліт-тестування. З масиву вибираються кілька елементів, які порівнюються між собою по черзі «до» і «після» зміни. A\B тести допомагають визначити, які чинники мають найбільший вплив на елементи. Наприклад, за допомогою спліт-тестування можна провести величезну кількість ітерацій поступово наближаючись до достовірного результату;

– прогнозування. Аналітики намагаються заздалегідь задати системі ті чи інші параметри і в подальшій перевіряти поведінку об’єкта на основі надходження великих масивів інформації;

– машинне навчання. Штучний інтелект в перспективі здатний поглинати і обробляти великі обсяги несистематизованих даних, згодом використовуючи їх для самостійного навчання;

– аналіз мережевої активності. Методики big data використовуються для дослідження соцмереж, взаємовідносин між власниками аккаунтів, груп, спільнотами. На основі цього створюються цільові аудиторії за інтересами, геолокації, віком і іншим метрик.

1.5 Нечітка кластеризація

В області аналізу даних нечітке моделювання часто дозволяє отримувати більш адекватні результати в порівнянні з результатами, які ґрунтуються на використанні традиційних аналітичних моделей і алгоритмів.

Нечітка множина є сукупністю елементів довільної природи, щодо яких не можна з повною певністю стверджувати – чи належить той чи інший елемент розглянутої сукупності даній множині чи ні [10].

Взаємозв’язок між кластерним аналізом і теорією нечітких множин засновано на тій обставині, що при вирішенні задач структуризації складних систем більшість об’єктів виявляються розмитими за своєю природою. Ця розмитість полягає в тому, що перехід від приналежності до неналежності елементів до даних класах швидше поступовий, ніж стрибкоподібною.

Вимога знаходження однозначної кластеризації елементів досліджуваної проблемної області є досить грубим і жорстким, особливо при вирішенні погано або слабо структурованих задач системного аналізу. Методи нечіткої кластеризації послаблюють цю вимогу. Ослаблення вимоги здійснюється за рахунок введення в розгляд нечітких кластерів і відповідних

їм функцій приналежності, які приймають значення з інтервалу [0, 1]. На рисунку 1.4 зображена нечітка кластеризація з різними ступеням приналежності спостережень [11].

Для будь-якої міри схожості величина приналежності спостереження кластеру залежить від схожості об’єкта і прототипу цього кластера. В разі якщо мірою подібності є відстань, то величина приналежності об’єкта обернено пропорційна його відстані до центроїда кластера. Сума приналежності спостереження кластерам в будь-який момент часу повинна бути дорівнює 1.

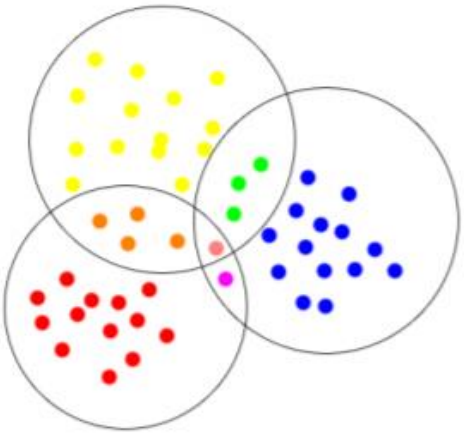


Рисунок 1.4 – Нечітка кластеризація

Таким чином, в загальному випадку завданням нечіткої кластеризації є знаходження нечіткого розбиття або нечіткого покриття безлічі елементів досліджуваної сукупності, які утворюють структуру нечітких кластерів, присутніх в розглянутих даних. Це завдання зводиться до знаходження ступенів належності елементів шуканим нечітким кластерам, які в сукупності і визначають нечітке розбиття або нечітке покриття вихідної множини розглянутих елементів [12].

Основні ідеї алгоритму для вирішення завдання нечіткої кластеризації були запропоновані в 1973 році. Надалі алгоритм був розвинений Джеймсом Бездеком і отримав назву нечітких *c*-середніх (FCM).

Поряд з традиційним імовірнісним підходом до нечіткої кластеризації, коли кожен об’єкт з певною ймовірністю належить до кожного з кластерів, існує ймовірнісна підхід до кластерного аналізу. Можливісна кластеризація також розглядає нечіткі кластери і відповідні їм функції приналежності, які беруть значення з інтервалу [0, 1]. Різниця полягає в тому, що ймовірнісна кластеризація має на увазі наявність суворого обмеження, що сума приналежності об’єкта до всіх кластерів дорівнює 1, а можливісна кластерний аналіз не має на увазі подібного обмеження.

Перевага можливісного кластерного аналізу над імовірнісним полягає в тому, що об’єкти, які мають низький рівень подібності з будь-яким з кластерів, будуть мати значення приналежності близьке нулю для всіх кластерів, в той час як імовірнісний нечіткий кластерний аналіз буде явно віддавати перевагу одному або декільком кластерам (хоча всі вони повинні бути досить погані).

Часто результати можливісної кластеризації перетворюють у імовірнісний вигляд за допомогою нормалізації, попередньо позбувшись від outlier-об’єктів. Зазвичай, результати отримані таким чином, краще результатів, отриманих при прямому використанні імовірнісної кластеризації (проте, це також залежить від вихідних даних, від конкретних алгоритмів нечіткої кластеризації і ініціалізації їх параметрів) [13].

Алгоритм FCM має ітеративний характер послідовного поліпшення деякого нечіткого розбиття, яке задається користувачем або формується автоматично за деяким евристичним правилом. На кожній з ітерацій рекурентно перераховуються значення функцій приналежності об’єктів нечітким кластерам і їх типові представники (центроїди).

Алгоритм закінчить роботу в разі, коли станеться виконання заданого апріорі деякого кінцевого числа ітерацій, або коли мінімальна абсолютна різниця між значеннями функцій приналежності (або центроїдами кластерів)

на двох послідовних ітераціях не стане менше деякого апріорі заданого значення. Формально алгоритм FCM визначається в формі ітеративного виконання наступній послідовності кроків:

– задається кількість шуканих нечітких кластерів m, максимальне кількість ітерацій алгоритму *n*, параметр збіжності алгоритму *ε*. Є поточною нечіткого розбиття на першій ітерації алгоритму для матриці даних *D* задається деякий вихідне нечітке розбиття на *m* непустих нечітких кластерів, які описуються сукупністю функцій приналежності;

– для вихідного поточного нечіткого розбиття розраховуються центри нечітких кластерів і значення цільової функції. Кількість виконаних ітерацій встановлюється в 1;

– формується нове нечітке розбиття вихідної множини об’єктів кластеризації на m непустих нечітких кластерів, характеризуються сукупністю функцій приналежності;

– для нового нечіткого розбиття розраховуються центри нечітких кластерів і значення цільової функції;

– якщо кількість виконаних ітерацій перевищує вказану кількість *n* або ж значення модуля різниці між значеннями функцій приналежності (або центроїдами кластерів) на двох останніх ітераціях менше значення параметра збіжності алгоритму *ε*, то в якості шуканого результату нечіткої кластеризації приймається останнім нечітке розбиття і виконання алгоритму припиняється.

Алгоритм FCM за своїм характером відноситься до наближених алгоритмів пошуку екстремуму для цільової функції при наявності обмежень. В результаті виконання даного алгоритму визначається локально-оптимальне нечітке розбиття, яке описується сукупністю функцій приналежності [14].

1.6 Постановка задачі дослідження

Таким чином, задача класифікації великих даних є актуальним завданням машинного навчання. Тому ставиться завдання розробки методу класифікації великих даних на основі байєсівського класифікатора.

Об’єктом дослідження даного дослідження є методи машинного навчання.

Метою даного дослідження є дослідження та реалізація методу, що базується на Байєсівському класифікаторі для великих обсягів даних у формі часових рядів.

Для цього необхідно вирішити такі завдання:

– провести аналіз існуючих методів класифікації великих обсягів даних;

– розробити власний метод для класифікації великих обсягів даних;

– використати сучасний сервіс Power BI для реалізації та візуалізації роботи розробленого методу.

2 МЕТОД МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ВЕЛИКИХ ОБСЯГІВ ДАНИХ

2.1 Лінійний класифікатор

Лінійні класифікатори класифікують дані по мітках на основі лінійної комбінації вхідних функцій. Отже, ці класифікатори поділяють дані за допомогою лінії, площини або гіперплощини (площині більш ніж в 2 вимірах). Їх можна використовувати тільки для класифікації даних, які можна розділити лінійно. Їх можна модифікувати для класифікації нелінійно розділених даних. Розглянемо два алгоритми лінійної класифікації: Логістичну регресію та Метод опорних векторів [15].

2.1.1 Модель бінарної логістичної регресії

У математичній статистиці логістична регресія є широко використовуваною статистичною моделлю, яка використовує логістичну функцію для моделювання залежності вихідної змінної від набору вхідних в разі, коли перша є бінарної [16].

Це різновид множинної регресії, загальне призначення якої полягає в аналізі зв’язку між декількома незалежними змінними і залежною змінною. Регресія в загальному вигляді застосовується, коли вхідні і вихідна змінні безперервні. А логістична регресія кращим чином підходить, коли вихідна змінна приймає тільки два значення.

Значення факторів в моделях бінарного вибору повинні бути виміряні в кількісній шкалі. Також в моделі бінарного вибору можна включати в якості факторів категоріальні змінні [17]. Для моделювання ймовірності дихотомічної залежної змінної підбирають спеціальну монотонно зростаючу функцію, яка може приймати значення в межах від 0 до 1.

Є спеціальна функції в моделях бінарного вибору зазвичай використовують:

– логістичну функцію;

– функцію стандартного нормального розподілу.

За допомогою методу бінарної логістичної регресії (рис. 2.1) можна досліджувати залежність дихотомічних змінних (бінарних, що мають лише два можливих значення) від незалежних змінних, що мають будь-який вид шкали.

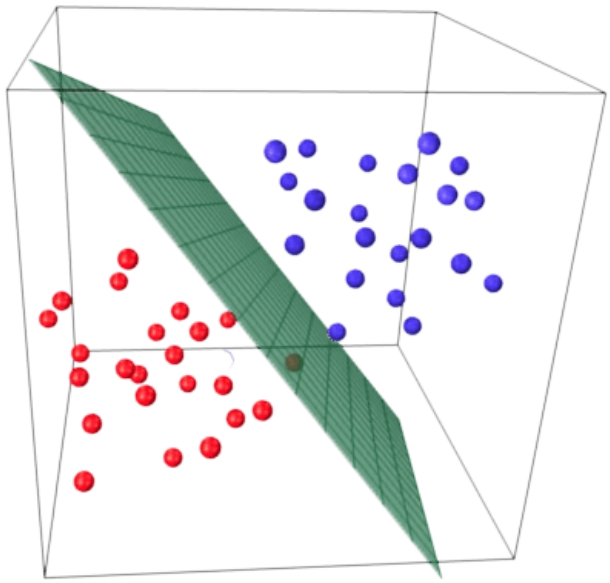


Рисунок 2.1 – Приклад роботи бінарної логістичної регресії

Як правило, у випадку з дихотомічними змінними мова йде про деяке подію, яка може відбутися або не відбутися; бінарна логістична регресія в такому випадку розраховує ймовірність настання події в залежності від значень незалежних змінних [18].

Імовірність настання події для деякого випадку розраховується за формулою

$$p = \frac{1}{1+e^{-z}}, \tag{2.1}$$

де $z= b_1*X_1+b_2*X_2+...+b_n*X_n + a$, X_1 – значення незалежних змінних;

b_1 – коефіцієнти, розрахунок яких є задачею бінарної логістичної регресії;

a – деяка константа.

Якщо для p вийде значення менше 0,5, то можна припустити, що подія не настане; в іншому випадку передбачається настання події.

За допомогою логістичної регресії прогнозується ймовірність відгуку для залежної змінної від включених в модель незалежних змінних. На основі прогнозних значень ймовірності можна зробити класифікацію всіх спостережень на дві групи. Окремим аналізом при побудові моделі логістичної регресії є аналіз ROC-кривих (Receiver Operator Characteristic). ROC-аналіз дозволяє вибрати оптимальне значення порогового значення ймовірності для класифікації. ROC-крива – крива, яка використовується для представлення результатів бінарної класифікації та оцінки ефективності класифікації [19].

2.1.2 Метод опорних векторів

Метод опорних векторів (рис. 2.2) – це набір алгоритмів, що використовуються для задач класифікації та регресійного аналізу [20].

Нехай x_i лежить на замикані межі, тобто $|w'x + w_0| = 1$. Межу, ширину роздільної полоси, потрібно зробити як можна більшою. Враховуючи, що замикання межі задовольняє умові $|w'x + w_0| = 1$, тоді відстань від x_i до $g(x)$=0 дорівнює

$$\frac{|w'x + w_0|}{\|w\|} = \frac{1}{\|w\|}. \qquad (2.2)$$

Таким чином, ширина роздільної полоси дорівнює $\frac{2}{\|w\|}$.

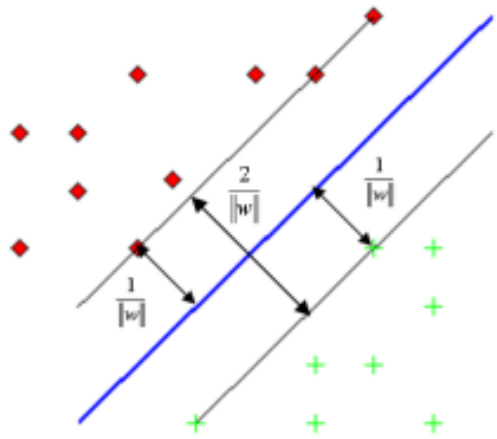


Рисунок 2.2 – Побудова опорних векторів

Для того, щоб виключити точки з розділяючої полоси, випишемо умову приналежності к одному з класів

$$\begin{cases} w'x_i + w_0 \geq 1, \text{ належить } x_i \text{ першому класу,} \\ w'x_i + w_0 \leq 1, \text{ належить } x_i \text{ другому класу.} \end{cases} \quad (2.3)$$

Введемо індексну функцію

$$\begin{cases} u_i = 1, \text{ якщо } x_i \text{ належить першому класу,} \\ u_i = -1, \text{ якщо } x_i \text{ належить другому класу.} \end{cases} \quad (2.4)$$

Таким чином, задача вибору розділяючої функції, породжуючої коридор найбільшої ширини можно записати у вигляді

$$J(w) = \frac{1}{2}\|w\|^2 \to \min, \quad (2.5)$$

при умові $u_i(w'x_i+w_0)\geq 1$ для усіх i.

Так цільова функція є квадратичною функцією, то у даній задачі існує одне рішення.

Згідно теоремі Куна-Таккера ця умова еквівалентна наступній задачі

$$L(a) = \sum_{i=0}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j u_i u_j x_i^T x_j \to \max, \tag{2.6}$$

при умові, що $a \geq 0 \; \forall i \; \sum_{i=1}^{n} a_i u_i = 0$, де $a = \{a_i, ..., a_n\}$ нові змінні.

Перепишемо *L(a)* в матричному вигляді

$$L(a) = \sum_{i=1}^{n} a_i - \frac{1}{2} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}^T H \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}, \tag{2.7}$$

де коефіцієнти матриці H рахуються наступним чином:

$$H_{i,j} = u_i u_j x_i^T x_j. \tag{2.8}$$

Задача $L(a) \to max$ вирішується методами квадратичного програмування.

Після находження оптимальних $a=\{a_1, a_2 ..., a_n\}$ для кожного *i* виконується одна з двох умов:

– $a_i = 0$ (і відповідає неопорному вектору);

– $a_i \neq 0$ та $u_i(w' x_i + w_0 - 1) = 0$ (і відповідає опорному вектору).

Тоді може бути знайдено w з відношення $w = \sum_{i=1}^{n} a_i u_i x_i$ та значення w_0 можна знайти, враховуючи, що для будь-якого $a_i > 0$ та $a_i[u_i(w' x_i + w_0) - 1] = 0$

$$w_0 = \frac{1}{u_i} - w' x_i. \tag{2.9}$$

Після чого отримаємо дискримінанту функцію

$$g(x) = (\sum \{a_i u_i x_i \,|\, x_i \in S\})^T x + w_0. \quad (2.10)$$

Сумування проводиться не по векторам, а тільки по множині S, яке представляє собою множину опорних векторів $S = \{x_i \,|\, a_i \neq 0\}$.

Найбільш масштабні проблеми, які були вирішені за допомогою методу опорних об'єктів (і його модифікованих реалізацій), виділяють відображення рекламних банерів на сайтах, розпізнавання статі на основі фотографії і сплайсинг людської ДНК [21].

2.2 Байєсівський класифікатор

Байєсівський класифікатори представляють собою сімейство простої ймовірної класифікації. Байєсівський класифікатор (рис. 2.3) заснований на застосуванні Теореми Байєса з строгими (наївними) пропозиціями про незалежність параметрів, де для заданого набору параметрів підбирається найвідповідніший клас. Найбільш часто використовувана область наївного байєсівського класифікатора відноситься до задачі класифікації текстів [22].

Передбачається, що алгоритм класифікації працює на деякій множині документів D = $\{b_i\}$. Вся множина документів розбивається на непересічні підмножини класів

$$C = \{c_i\}, \bigcup_i b_i = D, c_i \bigcap_i c_j = \varnothing (i \neq j). \quad (2.11)$$

Завданням класифікації є визначення класу, до якого належить даний документ. Кожному елементу b ставиться у відповідність набір ознак $b=\{w_i\}$.

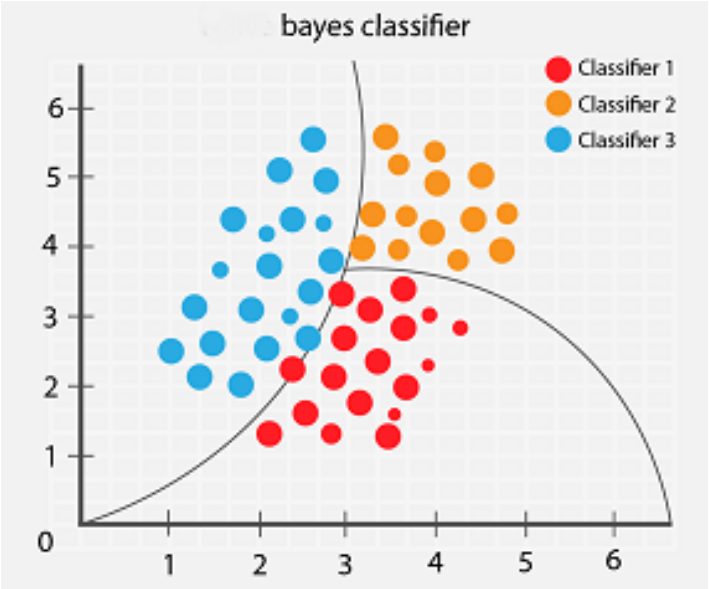


Рисунок 2.3 – Приклад роботи байєсівського класифікатора

Далі застосовується алгоритм класифікації для виділення документів найбільш відповідних заданому класу [23].

Для того, щоб застосувати теорему Байєса для класифікації документів, робляться такі припущення:

$$P(c_j) = \frac{n(c_j)}{\sum_j n(c_j)}, \tag{2.12}$$

де $n(c_j)$ – кількість термів в класі c_j.

Передбачається, що всі терми (слова, словосполучення) незалежні, відповідно

$$P(w_i \left| c_j \right.) = \frac{n(w_i, c_j)}{n(c_j)}, \tag{2.13}$$

де $\{w_i\}$ – набір термінів в документі b;

$n(w_i,c_j)$ – кількість термів w_i у класі c_j.

Для визначення відповідної категорії документів для заданого документа, потрібно отримати відповідну множину словоформ. По множині словоформ будується структура з неповторюваних слів і їх лічильників – (w_i,n_i).

Визначення підходящої категорії починається з кореня дерева множин статистики. Через M позначимо кількість множин статистики в даному вузлі дерева. Категорії позначимо через $c_j\,(j=0,...,M-1)$. Для кожного слова w_i в кожній множині статистики знаходимо це слово і відповідний лічильник $n(w_i,C_j)$ (тут j (j=0,1,…,M-1) – номер категорії (множини статистики)). Через $n(c_j)$ позначимо число документів у j-ї категорії [24].

Крім того, нехай

$$N_j(w_i)=\frac{n(w_i,c_j)}{n(c_j)}, \tag{2.14}$$

нормований лічильник слова w_i в j-ї категорії.

Тоді ймовірність відповідності унікального (тобто кожне слово зустрічається тільки один раз) слова w_i j-ї категорії буде дорівнювати

$$P(c_j\left|w_i\right.)=\frac{N_j(w_i)}{S(w_i)}n(w_i,c_j), \tag{2.15}$$

де

$$S(w_i)=\sum_{j=0}^{M-1}N_j(w_i). \tag{2.16}$$

Якщо $P(c_j\left|w_i\right.)=0,$ то потрібно взяти це число рівним малому значенню, наприклад, рівним 0,0001.

Тоді ймовірність того, що документ відповідає категорії c_j (j =0,…,M-1) буде дорівнювати

$$P(c_j \left| \{w_i\}) = P(c_j) \prod_i P(c_j | w_i), \right. \quad (2.17)$$

де добуток береться по всім словам досліджуваної множини словоформ і

$$P(c_j) = \frac{n(c_j)}{\sum_{j=0}^{M-1} n(c_j)}. \quad (2.18)$$

$P(c_j)$ – апріорна ймовірність зустрічі категорії c_j. Зауважимо, що якщо документ містить велику кількість слів, які не зустрічаються в категорії c_j (j= 0, .., M-1), то значення $P(c_j \left| \{w_i\}) \right.$ може вийти за межі визначення змінної. Тому, значення $P(c_j \left| \{w_i\}) \right.$ потрібно контролювати, і, якщо воно виходить за межі значення змінної, то обмежувати заданим малим числом, наприклад, просто брати рівним 0.

Після першого кроку визначаємо k-штук категорій з найбільшим значенням $P(c_j \left| \{w_i\}) \right.$. Зберігаємо їх назву і значення $P(c_j \left| \{w_i\}) \right.$. Відповідно до назви кожного з цих множин, заходимо в відповідний пункт і проводимо обробку. Якщо інформація відсутня, то цей пункт пропускається. Після того, як будуть опрацьовані всі дочірні (за певними на першому кроці) категорії, з них і збережених на попередньому кроці, вибирається k-штук категорій з найбільшим значенням $P(c_j \left| \{w_i\}) \right.$. Зберігаємо їх назва і значення $P(c_j \left| \{w_i\}) \right.$, після чого, переходимо до наступного кроку, але тільки по тим категоріям, які не були збережені на попередньому кроці. Процес продовжуємо до тих пір, поки не буде категорій, за якими можна здійснювати перевірку. Результатом

виконання програми буде *k*-штук категорій, найбільш ймовірних для досліджуваного документа, з точки зору критерію Байєса [25].

2.3 Дерево рішень

Дерево рішень – ефективний інструмент інтелектуального аналізу даних і прогнозної аналітики. Він допомагає у вирішенні завдань щодо класифікації та регресії [26].

Дерево рішень (рис. 2.4) є ієрархічну деревоподібну структуру, що складається з правила виду «Якщо ..., то ...». За рахунок навчальної множини правила генеруються автоматично в процесі навчання.

Правила генеруються за рахунок узагальнення безлічі окремих спостережень (навчальних прикладів), що описують предметну область. Тому їх називають індуктивними правилами, а сам процес навчання індукцією дерев рішень.

У навчальній множині для прикладів має бути задано цільове значення, так як дерева рішень – моделі, створювані на основі навчання з учителем. За типом змінної виділяють два типи дерев:

– дерево класифікації – коли цільова змінна дискретна;
– дерево регресії – коли цільова змінна безперервна.

Дерево рішень – метод представлення вирішальних правил в певній ієрархії, що включає в себе елементи двох типів – вузлів (node) і листя (leaf). Вузли включають в себе вирішальні правила і проводять перевірку прикладів на відповідність обраного атрибута навчальної множини.

Простий випадок: приклади потрапляють в вузол, проходять перевірку і розбиваються на два підмножини:

– перше – ті, які задовольняють встановлене правило;
– друге – ті, які не задовольняють встановлене правило.

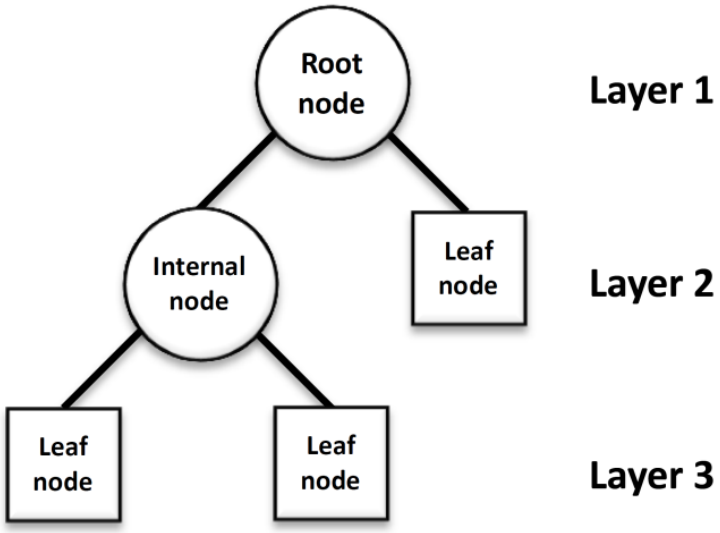


Рисунок 2.4 – Приклад дерева рішення

Далі до кожної підмножини знову застосовується правило, процедура повторюється. Це триває, поки не буде досягнута умова зупинки алгоритму. Останній вузол, коли не провадиться перевірка і розбиття, стає листом [27].

Лист визначає рішення для кожного, що потратив в нього прикладу. Для дерева класифікації – це клас, що асоціюється з вузлом, а для дерева регресії – відповідний листу модальний інтервал цільової змінної. У листі міститься не правило, а підмножина об’єктів, які відповідають всім правилам гілки, яка закінчується цим листом.

Приклад потрапляє в лист, якщо відповідає всім правилам на шляху до нього. До кожного листа є тільки один шлях. Таким чином, приклад може потрапити тільки в один лист, що забезпечує єдність розв’язку.

Основне завдання при побудові дерева рішень – послідовно і рекурсивно розбити навчальну множину на підмножини з застосуванням вирішальних правил в вузлах. Цей процес продовжують до того, поки всі вузли в кінці гілок не стануть листами [28].

Вузол стає листом в двох випадках:

– природним чином – коли він містить єдиний об’єкт або об’єкт тільки одного класу;

– після досягнення заданого умови зупинки алгоритм – наприклад, мінімально допустиму кількість прикладів в вузлі або максимальна глибина дерева.

В основі побудови лежать «жадібні» алгоритми, що допускають локально-оптимальні рішення на кожному кроці (розбиття в вузлах), які призводять до оптимального підсумкового рішення. Тобто при виборі одного атрибута і проведенні розбиття по ньому на підмножини, алгоритм не може повернутися назад і вибрати інший атрибут, навіть якщо це дасть краще підсумкове розбиття. Отже, на етапі побудови дерева рішень можна точно стверджувати, що вдасться домогтися оптимального розбиття.

Популярні алгоритми, які використовуються для навчання дерев рішень, будуються на основі принципу «розділяй і володарюй». Задають загальну множину S, що містить:

– n прикладів, для кожного з яких задана мітка класу C_i (i = 1,2,..,k);

– m атрибутів A_j (j = 1,2,...,m), які визначають належність об’єкта до того чи іншого класу.

Тоді можливо три випадки:

– приклади множин S мають однакову мітку C_i, отже, все навчальні приклади відносяться до одного класу. В такому випадку навчання не має сенсу, тому що всі приклади в моделі будуть одного класу, який і «навчиться» розпізнавати модель. Саме дерево буде схоже на один великий аркуш, асоційований з класом C_i. Тоді його використання не матиме сенсу, тому що все нові об’єкти будуть ставитися до одного класу;

– множина S – порожня множина без прикладів. Для нього сформується лист, клас якого вибереться з іншої множини. Наприклад, найпоширеніший з батьківської множини клас;

– множина S складається з навчальних прикладів всіх класів C_k. В такому випадку безліч розбивається на підмножини відповідно до класів. Для цього вибирають один з атрибутів A_j безлічі S, що складається з двох і більше унікальних значень: (a_1, a_2, ..., a_p), де p – число унікальних значень ознаки.

Множину S розбивають на p підмножин (S_1, S_2, ..., S_p), що складаються із прикладів з відповідним значенням атрибута. Процес розбиття триває, але вже з наступним атрибутом. Він буде повторюватися, поки всі приклади в результуючих підмножини не опиняться одного класу.

Третя застосовується в більшості алгоритмів, використовуваних для побудови дерев рішень. Ця методика формує дерево зверху вниз, тобто від кореневого вузла до листя.

Алгоритм навчання може працювати до отримання «чистих» підмножин з прикладами одного класу. В такому випадку висока ймовірність отримати дерево, в якому для кожного прикладу буде створено окремий лист. Таке дерево не вийде застосовувати на практиці через перенавчання. Кожному наприклад буде відповідати свій унікальний шлях в дереві. Вийде набір правил, актуальний тільки для даного прикладу [29].

Перенавчання в разі дерева рішень має схожі з нейронними мережами наслідки. Воно буде точно розпізнавати приклади з навчання, але не зможе працювати з новими даними. Ще один мінус – структура перенавчання дерева складна і погано піддається інтерпретації.

Фахівці вирішили примусово зупиняти будівництво дерева, щоб воно не ставало «перенавчання». Для цього використовують кілька підходів:

– рання зупинка. Алгоритм зупиняється після досягнення заданого значення критерію (наприклад, процентної частки правильно розпізнаних прикладів). Перевага методу – скорочення витрат часу на навчання. Головний недолік – рання зупинка негативно позначається на точності дерева. Через це багато фахівців радять віддавати перевагу відсікання гілок;

– обмеження глибини дерева. Алгоритм зупиняється після досягнення встановленого числа розбиття в гілках. Цей підхід також негативно позначається на точності дерева;

– завдання мінімально допустимого числа прикладів у вузлу. Встановлюється обмеження на створення вузлів з числом прикладом менше

заданого. В такому разі не будуть створюватися тривіальні розбиття і незначні правила.

Цими підходами користуються рідко, тому що вони не гарантують кращого результату. Найчастіше, вони працюють тільки в якихось певних випадках. Рекомендацій щодо використання будь-якого методу немає, тому аналітикам доводиться набирати практичний досвід шляхом проб і помилок.

Без обмеження «зростання» дерево рішень стане занадто великим і складним, що унеможливить подальшу інтерпретацію. А якщо робити вирішальні правила для створення вузлів, в які будуть потрапляти по 2-3 приклади, вони не втратять практичної цінності.

Сьогодні існує багато алгоритмів навчання: ID3, CART, C4.5, C5.0, NewId, ITrule, CHAID, CN2 і інші. Найпопулярнішими вважаються:

– ID3 (Iterative Dichotomizer 3). Алгоритм дозволяє працювати тільки з дискретної цільової змінної. Дерева рішень, побудовані на основі ID3, виходять кваліфікуючими. Число нащадків в вузлу необмежено. Алгоритм не працює з пропущеними даними;

– C4.5. «Просунута» версія ID3, доповнена можливістю роботи з пропущеними значеннями атрибутів;

– CART (Classification and Regression Tree). Алгоритм вирішує завдання класифікації і регресії, так як дозволяє використовувати дискретну та неперервну цільові змінні. CART будує дерева, в кожному вузлі, що має тільки два нащадки.

2.4 Способи оцінки якості регресії

Якістю моделі регресії називається адекватність побудованої моделі вихідним (спостережуваним) даними. Для оцінки якості моделі регресії використовуються спеціальні показники:

– парний лінійний коефіцієнт кореляції, який розраховується за формулою:

$$r_{yx} = \frac{\overline{xy} - \bar{x} * \bar{y}}{G(x) * G(y)} = \frac{Cov(x, y)}{G(x) * G(y)}, \tag{2.19}$$

де $G(x)$ – середньоквадратичне відхилення незалежної змінної;

$G(y)$ – середньоквадратичне відхилення залежної змінної.

Також парний лінійний коефіцієнт кореляції можна розрахувати через МНК-оцінку коефіцієнта моделі регресії за формулою:

$$r_{yx} = \tilde{\beta} * \frac{G(x)}{G(y)}. \tag{2.20}$$

Парний лінійний коефіцієнт кореляції характеризує ступінь тісноти зв'язку між досліджуваними змінними. Він розраховується тільки для кількісних змінних. Чим ближче модуль значення коефіцієнта кореляції до одиниці, тим більш тісним є зв'язок між досліджуваними змінними. Даний коефіцієнт змінюється в межах [30].

Якщо значення коефіцієнта кореляції знаходиться в межах від нуля до одиниці, то зв'язок між змінними пряма, тобто зі збільшенням незалежної змінної збільшується і залежна змінна, і навпаки. Якщо коефіцієнт кореляції знаходиться в межах від мінус одиниці до нуля, то зв'язок між змінними зворотній, тобто зі збільшенням незалежної змінної зменшується залежна змінна, і навпаки. Якщо коефіцієнт кореляції дорівнює нулю, то зв'язок між змінними відсутній. Якщо коефіцієнт кореляції дорівнює одиниці або мінус одиниці, то зв'язок між змінними існує функціональний зв'язок, тобто зміни незалежної і залежної змінних повністю відповідають один одному.

Коефіцієнт детермінації розраховується як квадрат парного лінійного коефіцієнта кореляції і позначається як r_{yx2}. Даний коефіцієнт характеризує в

процентному відношенні варіацію залежної змінної, пояснений варіацією незалежної змінної, в загальному обсязі варіації.

Якість лінійної моделі множинної регресії характеризується за допомогою показників, побудованих на основі теореми про розкладання дисперсії.

Теорема. Загальна дисперсія залежної змінної може бути розкладена на пояснений і непоясненим побудованої моделлю регресії дисперсії [31].

В даний час існує дві основні концепції в боротьбі за підвищення якості прогнозних регресійних моделей:

– виявлення за наступним виключенням з аналізу єдиною аномальною незв’язок (виявлення з подальшою ліквідацією декількох аномальних незв’язок на основі поетапного усунення по одному викиду);

– знаходження із наступним виключенням більшої кількості незв’язок, які не завжди є аномальними і їх спільне відкидання призводить до мінімальних змін параметрів вихідного регресійного рівняння.

Першу концепцію реалізують методи Ектона, Бекмана, а так само Прескотта-Лунда. Ці методи призначені для виявлення з подальшим видаленням єдиного аномального вимірювання при нормальному законі розподілу випадкових величин нев’язок і їх кількості $n \geq 30$.

Друга концепція – перший метод принципово нової концепції – метод Кука. Трохи згодом з’явилися методи Белслі-Ку-Велша і Аткінсона. Суть методу Кука полягає в знаходженні при відкиданні прирівнюють вимірювань, які стабілізують параметри, нового регресійного рівняння по відношенню до вихідного.

3 ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДУ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ВЕЛИКИХ ДАНИХ

3.1 Програмне забезпечення Power BI

Power BI – комплексне програмне забезпечення бізнес-аналізу (BI) компанії Microsoft [32].

Microsoft Power BI має безліч вбудованих конекторів до різних сервісів і баз даних, з допомогою яких є можливість завантажувати в програму потрібний набір даних з різних джерел, зв’язати їх між собою і будувати консолідовані звіти і діаграми. У Power BI можна імпортувати дані з найбільш відомих баз даних і сервісів, використовуючи різні формати файлів. Доступні коннектори розподілені на чотири групи:

– група «Файл» (Excel, CSV, XML, Текст, JSON, Папка);

– група «База даних» (SQL Server, Access, SQL Server Analysis Service, Oracle, IBM DB2, MySQL, PostgreSQL, Sybase, Teradata, SAP HANA);

– група «Azure» (База даних Microsoft Azure SQL, Microsoft Azure Marketplace, Microsoft Azure HDInsight, Місце BLOB-об’єктів, Таблично централізованому сховищі вільно Microsoft Azure, Azure HDInsight Spark, Microsoft Azure DocumentDB, Місце озера даних Microsoft Azure);

– група «Інше» (Інтернет, Список SharePoint, Канал OData, Файл Hadoop, Active Directory, Microsoft ExChage;, Dynamics CRM online, Facebook, Google Analytics, Об’єкти Salesfore, Звіти Salesforce, ODBC, R-скрипт, appFigures, GitHub, MailChimp, Marketo, QuickBook Online, Smartsheets, SQL Sentry, Stripe, SweetIQ, Twilio, Zendesk, Spark, Порожній запит).

Power BI – це аналітична середовище (комплекс програм і онлайн сервісів), яка дає можливість:

– збір інформації абсолютно з будь-яких джерел даних. Це можуть бути різні сервіси, бази даних, файли, Google Docs, Яндекс-Диск, Excel, csv,

папки, документи, дані з Інтернет, API і різні інші коннектори, які щомісяця розробляє і додає в програму команда Power BI;

– обробка отриманих даних, їх приведення до єдиного вигляду і стандарту. Об’єднання і зв’язок всіх розрізнених таблиць в єдину модель даних (інформаційний колодязь);

– розробка і моделювання власних формул, метрик, показників і KPI для контролю і аналізу необхідних параметрів управління даними;

– інтерактивна візуалізація всіх метрик, KPI, таблиць в графічному вигляді. Що покращує і багаторазово прискорює процес відстеження, порівняння та аналізу операційної інформації в бізнес управлінні;

– представлення всіх звітів і дашборда через Інтернет за допомогою Online служби Power BI Service або через мобільний додаток Power BI Mobile;

– надання роздільних прав доступу для співробітників;

– використання серверних потужностей хмари Microsoft для автоматичної обробки будь-якої кількості даних;

– автоматичне оновлення всієї інформації (в моделі даних звітів), розміщеної в хмарі Power BI, що дозволяє отримувати актуальні дані в звітах Power BI в режимі онлайн «прямо тут і зараз»;

– автоматичне сповіщення системою потрібних співробітників при досягненні критичних значень в заданих KPI.

І все повністю автоматизовано, автоматично обновлюється і доступно online для аналізу з будь-яких пристроїв (ПК, планшети, смартфони) в інтерактивному режимі з наданням індивідуальних доступів для перегляду різними користувачами (рис. 3.1).

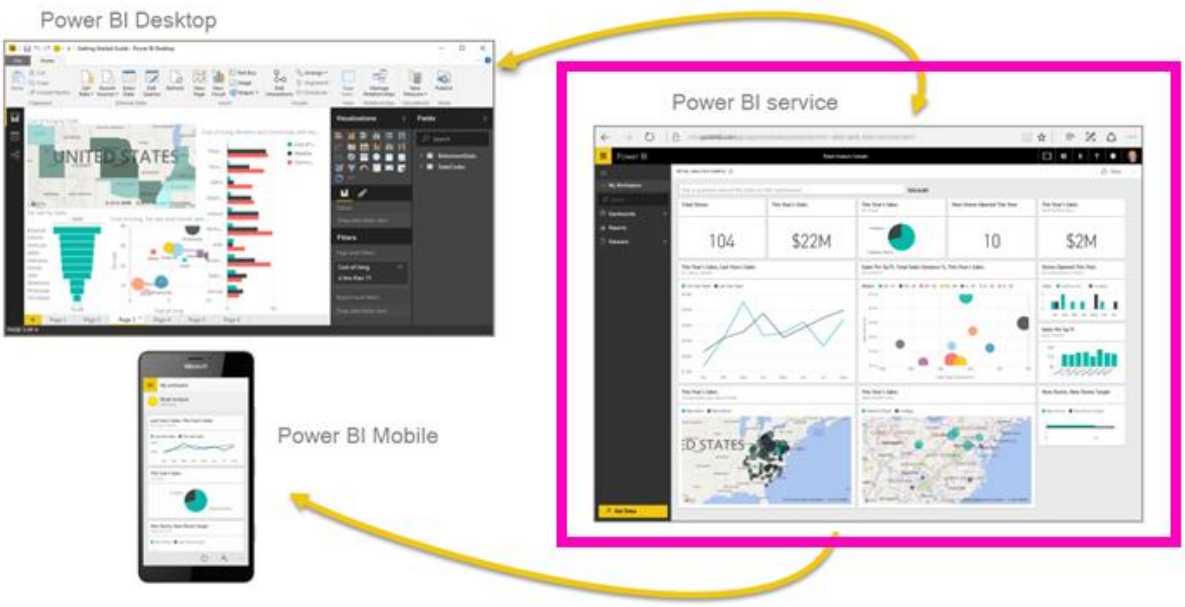


Рисунок 3.1 – Схема взаємодії продуктів Power BI

Power BI містить інструменти для самостійного створення інтерактивних звітів (dashboard):

– Power BI Desktop (рис. 3.2) – це додаток для завантаження, аналізу та візуалізації даних на dashboard-панелях, яке також дозволяє публікувати готові dashboard-звіти на порталі Power BI;

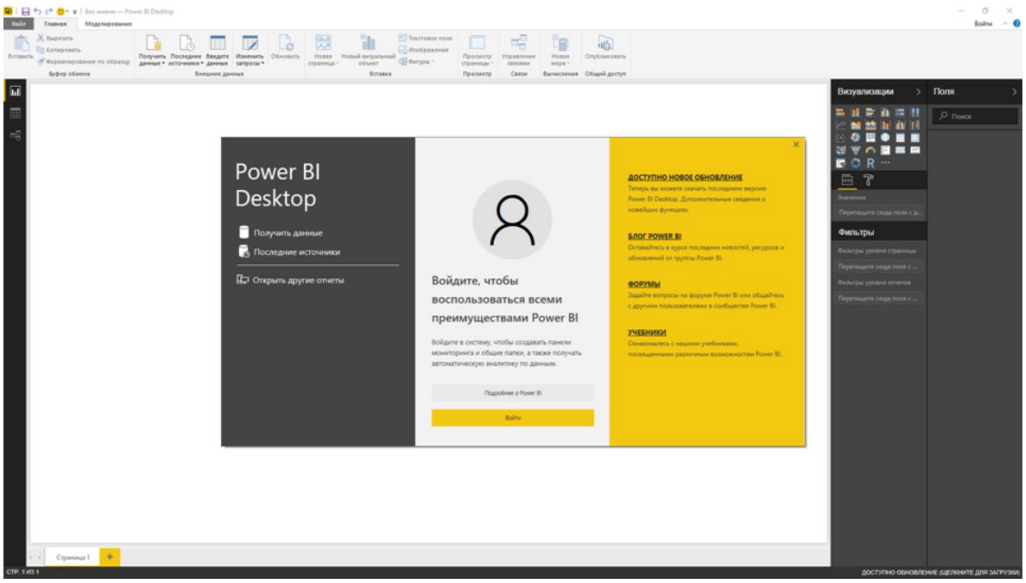


Рисунок 3.2 – Power BI Desktop

– портал Power BI (рис. 3.3) – це хмарний портал Power BI, який використовується для публікації dashboard-звітів та спільної роботи зі звітами Power BI;

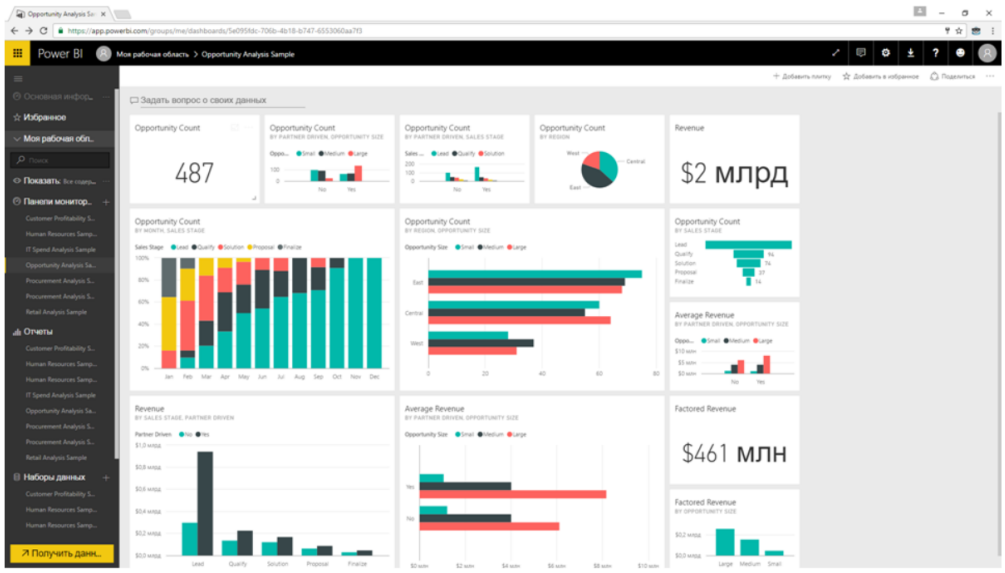


Рисунок 3.3 – Портал Power BI

– Power BI Premium – це розширені засоби бізнес-аналітики Power BI, які дозволяють розгорнути служби Power BI на серверах всередині організації.

Цей підхід дозволять перевести хмарні компоненти Power BI в локальні (тобто розгорнути їх на серверах Усередині організації). В першу чергу це дозволяє замінити хмарний портал Power BI, який використовується для публікації звітів, на локальний портал Power BI Report Server. Рішення Power BI Premium передбачає окрему модель ліцензування, при якій немає необхідності купувати ліцензії для кожного користувача [33];

– Power BI Mobile дозволяє відкривати і переглядати будь-які звіти Power BI, які опубліковані на хмарному порталі Power BI або на локальному порталі Power BI Report Server.

Додаток Power BI Mobile працює на будь-яких пристроях з операційною системою Windows 10, iOS і Android. Це дозволяє отримати доступ до звітів Power BI за допомогою мобільних пристроїв;

– Power BI Embedded – це інтеграційні засоби Power BI, які призначені для розробників стороннього програмного забезпечення. За допомогою Power BI Embedded можна вбудувати компоненти бізнес – аналітики Power BI безпосередньо в стороннє додаток, використовуючи спеціальний набір API;

– Сервер Power BI Report Server – це локальний портал для публікації звітів Power BI, який входить до складу рішення рівня Power BI Premium і є альтернативою хмарного порталу Power BI. Портал Power BI Report Server розгортається на серверах всередині організації і доступний за обраним корпоративним доменом;

– Додатки Power BI Insights apps – це готові рішення бізнес – аналітики, реалізовані на платформі Microsoft Power BI [34].

3.2 Аналіз даних в середовищі Power BI

Вбудовані функції Power BI дозволяють впоратися з завданням аналізу даних. Є можливість розробляти коротку аналітику і ділитися їй в звітах і на інформаційних панелях з різними робочими групами в організації [35].

Можливості розширеної аналітики Power BI дозволяють виявляти категорії і тенденції, переглядати зміни в даних з плином часу і багато іншого. За допомогою цих відомостей можна створювати прогнозні моделі даних і допомагати організації в прийнятті більш зважених бізнес-рішень, ефективних планів і прогнозів.

Функція кластеризації Power BI (рис. 3.4) дозволяє швидко знаходити групи схожих точок даних в підмножині даних. Вона аналізує набір даних для виявлення подібностей і відмінностей в значеннях атрибутів, а потім розділяє дані, що мають схожість, в підмножина даних. Ці підмножини даних називаються кластерами.

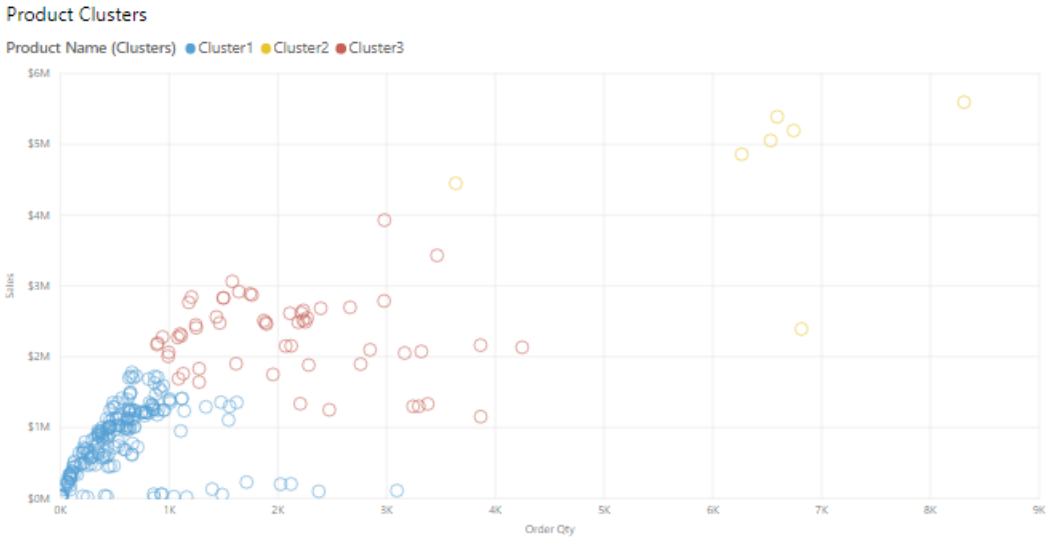


Рисунок 3.4 – Приклад кластеризації, застосованої до точкової діаграмі

Аналіз часових рядів включає в себе аналіз ряду даних за часом для виявлення значущих відомостей і тенденцій, а потім робить прогнози. Результатом аналізу часових рядів є дані, які найкраще підходять для прогнозування дій.

Аналіз часових рядів часто включає в себе використання візуальних елементів, таких як діаграми Ганта, планування проектів та набори даних про переміщення запасів. У Power BI можна використовувати візуальні елементи для перегляду зміни даних з плином часу, що, в свою чергу, дозволяє робити спостереження, наприклад про те, чи є які-небудь важливі події, що порушують роботу з даними.

Щоб виконати аналіз часових рядів в Power BI (рис. 3.5), необхідно використовувати тип візуалізації, який підходить для відображення тенденцій і змін з плином часу, таких як графік, діаграма з областями або точкова діаграма. Також є можливість імпортувати власний візуальний елемент часових рядів в Power BI Desktop з Microsoft AppSource.

На додаток до діапазону настроюються візуальні елементи часових рядів в Microsoft AppSource є візуальний елемент анімації Ось відображення, яка

працює як динамічний зріз і є способом відображення тенденцій і закономірностей в даних без будь-якої взаємодії з користувачем

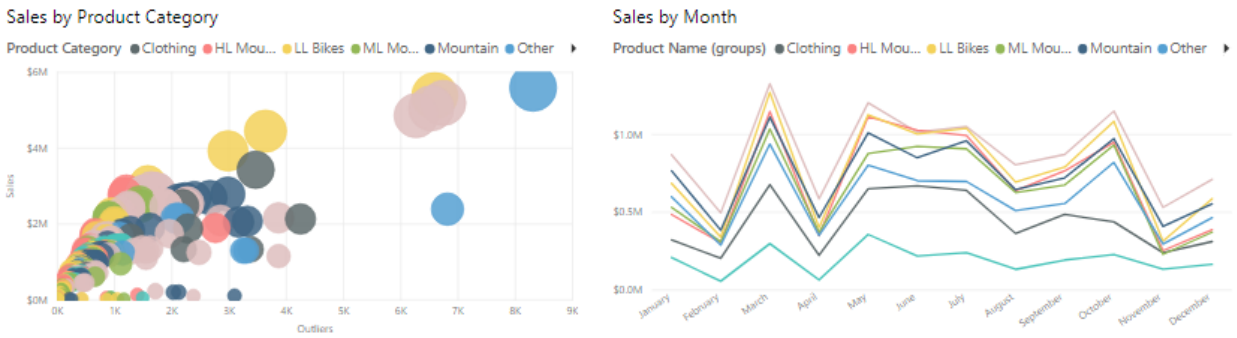


Рисунок 3.5 – Приклад аналізу часового ряду

На додаток до готових візуалізацій, які є в складі Power BI Desktop, Microsoft AppSource пропонує велику бібліотеку настроюються візуальних елементів, які можна імпортувати в Power BI Desktop. Ці настроюються візуальні елементи дозволяють розширити можливості вибору, коли мова заходить про використання розширеної аналітики. Може бути настроюється візуальний елемент, що дозволяє вирішити бізнес-завдання, яку не вдається вирішити за допомогою стандартних візуальних елементів, або візуальний елемент, який представляє дані так, як це не можуть зробити стандартні візуальні елементи.

З допомогою засобів Power BI є, також, можливість виявлення викидів. Викид (рис. 3.6) – це тип аномалій в даних на основі середнього або накопичених результатів. Потрібно визначити викиди, щоб ізолювати точки даних, які значно відрізняються від інших точок даних, а потім виконати аналіз причин відмінностей. Результати цього аналізу можуть істотно вплинути на прийняття бізнес-рішень.

Power BI дозволяє визначати викиди в даних, але спочатку необхідно визначити логіку, яка формує викид. Є можливість використовувати порогові точки, такі як обчислення, на основі яких можна вважати, що стався викид.

Процес визначення викидів включає сегментування даних за двома групами. Одна група – це дані викидів, а інша група – ні. Обчислювані стовпці можна використовувати для виявлення викидів, проте результати будуть статичними до тих пір, поки не будуть оновлені дані. Найбільш ефективним способом виявлення викидів є використання візуалізації або формули DAX, оскільки ці методи гарантують, що результати будуть динамічними [36].

Після визначення викидів в даних можна використовувати зрізи або фільтри для виділення цих викидів. Крім того, можна додати умовні позначення в візуальні елементи, щоб викиди можна було з легкістю ідентифікувати на тлі інших даних. Потім можна деталізувати дані викидів для більш докладного аналізу.

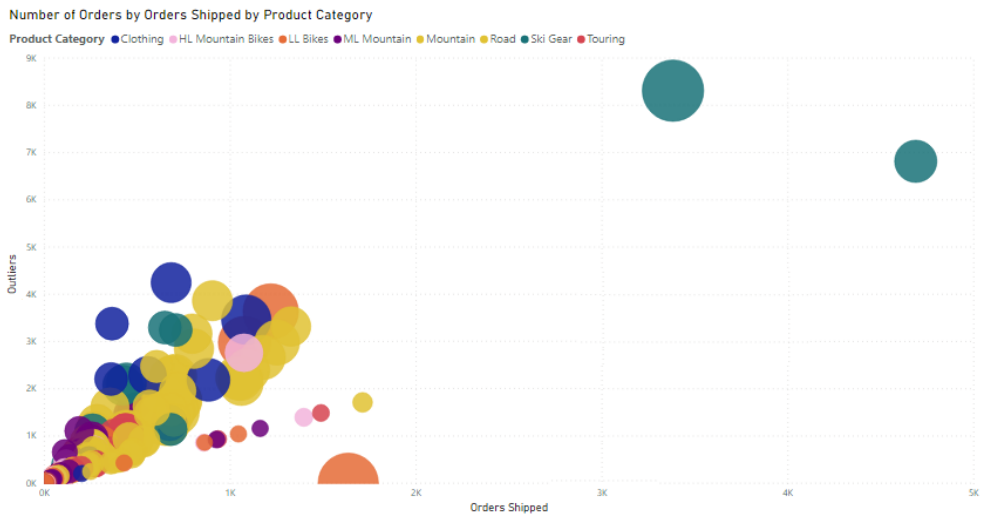


Рисунок 3.6 – Приклад точкової діаграма для заповнення викидів

3.3 Використання мови R у Power BI

Система PowerBI не дозволяє проводити глибокий статистичний аналіз даних. Вона дозволяє тільки візуалізувати дані, будувати різні діаграми, дивитися їх розрізи, провалюватися в дані всередині дивлячись їх додаткові

деталі. Однак, часто буває, що необхідно провести більш глибокий і більш детальний аналіз. Для цього в цю систему інтегрований пакет R пакет [37].

Мова R – це потужна мова програмування, що використовується для статистики, дослідження даних і аналітики даних. Пакет R – це і пакет, і програмне середовище, яка володіє дуже широкими можливостями. Усередині цього пакета можна проводити статистичний аналіз даних, можна досліджувати часові ряди, можна розв'язувати рівняння різного виду, диференціальні рівняння, лінійні, нелінійні рівняння, також системи рівнянь, вирішувати завдання оптимізації і проводити багато методи Data Mining. Також можна вирішувати завдання класифікації за допомогою нейронних мереж і інші можливості. Є можливість використовувати R в редакторі Power Query для Power BI Desktop, щоб:

– підготувати моделі даних;

– створювати звіти;

– очищувати дані, розширено формувати дані та аналітику наборів даних, включаючи заповнення відсутніх даних, прогнози, кластеризацію.

Служба Power BI підтримує перегляд і взаємодію з візуальними елементами, створеними за допомогою сценаріїв R. Візуальні елементи, створені за допомогою сценаріїв R, зазвичай називають візуальними елементами R, можуть представляти розширене формування даних та аналітику, таку як прогнозування, з використанням багатої аналітики і можливостей візуалізації R.

Візуальні елементи R, які створюються в Power BI Desktop, а потім публікуються в службі Power BI, здебільшого ведкть себе як будь-які інші візуальні елементи в службі Power BI; є можливість взаємодіяти, фільтрувати, нарізати і закріплювати їх на панелі інструментів або ділитися ними з іншими.

Візуальні елементи R створюються з сценаріїв R, які потенційно можуть містити код із загрозами безпеці або конфіденційності. Ці ризики в основному існують на етапі розробки, коли автор сценарію запускає сценарій на своєму власному комп'ютері.

Служба Power BI застосовує технологію пісочниці для захисту користувачів і служби від загроз безпеки. Цей підхід з пісочницею накладає деякі обмеження на сценарії R, що виконуються в службі Power BI, такі як доступ до Інтернету або доступ до інших ресурсів, які не потрібні для створення візуального елемента R.

Пакети R – це колекції функцій, даних і скомпільованої коду R, які об’єднані в чітко визначений формат. Коли R встановлений, він поставляється зі стандартним набором пакетів, а інші пакети доступні для завантаження і установки. Після установки пакет R повинен бути завантажений в сеанс, який буде використовуватися. Основне джерело безкоштовних пакетів R є CRAN, то Всебічне R Archive Network.

Power BI Desktop може використовувати будь-який тип пакетів R без обмежень. Є можливість встановити пакети R для використання в Power BI Desktop самостійно.

Для того, щоб запустити власний скрипт написаний на мові R в Power BI необхідно обрати Отримати дані, далі обрати Інше та Сценарій R, а потім обрати Підключитися (рис. 3.7).

Після цього з’явиться вікно, в яке необхідно скопіювати або написати власний сценарій, написаний на мові R (рис. 3.8). Далі для того, щоб запустити цей сценарій необхідно натиснути на ОК. Після успішного виконання сценарію можливо обрати отриманні дані для додати їх у модель Power BI.

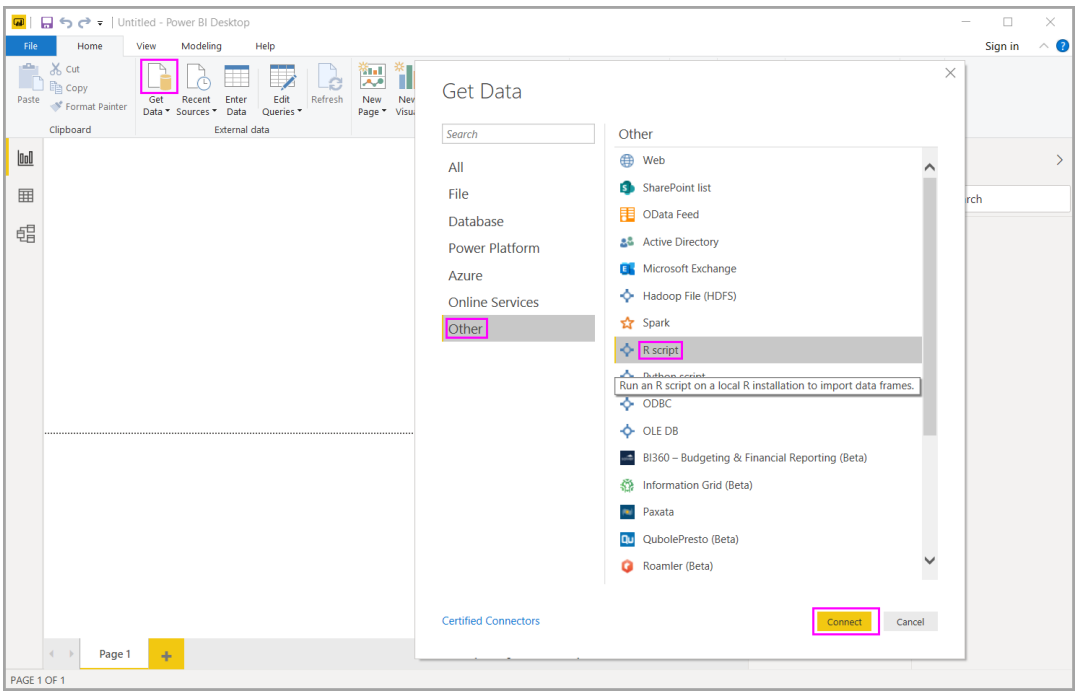


Рисунок 3.7 – Вікно для вибору джерел отримання даних

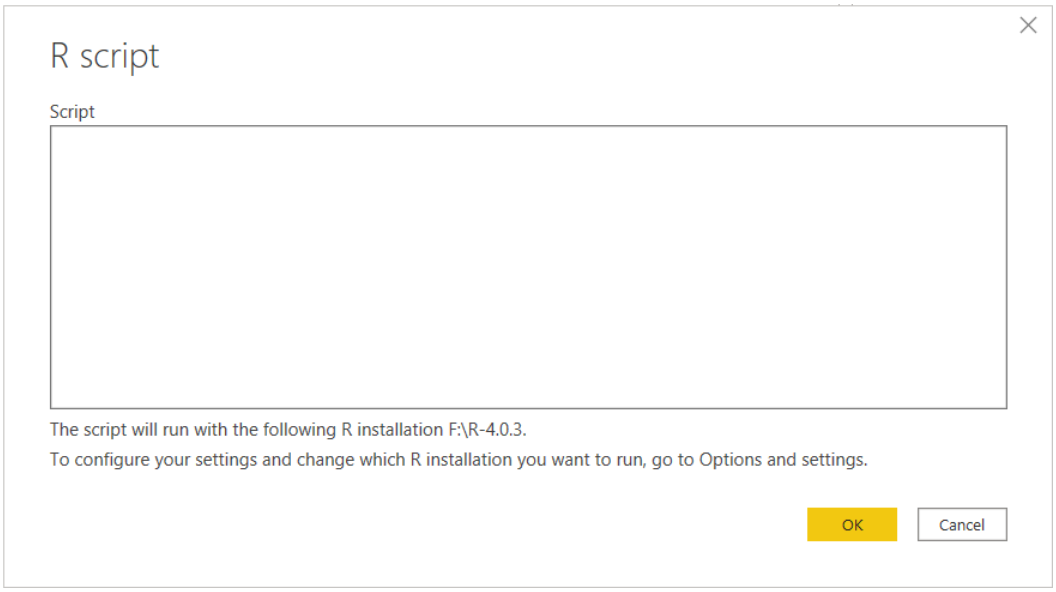


Рисунок 3.8 – Вікно для написання сценаріїв на мові R

3.4 Програмне забезпечення Tableau

Tableau Desktop (рис. 3.9) – це система аналітики (BI), яка допомагає бізнесу розкривати зміст даних, прискорюючи пошук необхідних показників. Інтернет-сервіс об’єднує підготовку візуальних даних і аналітичні інструменти для забезпечення наскрізного аналітичного процесу [38].

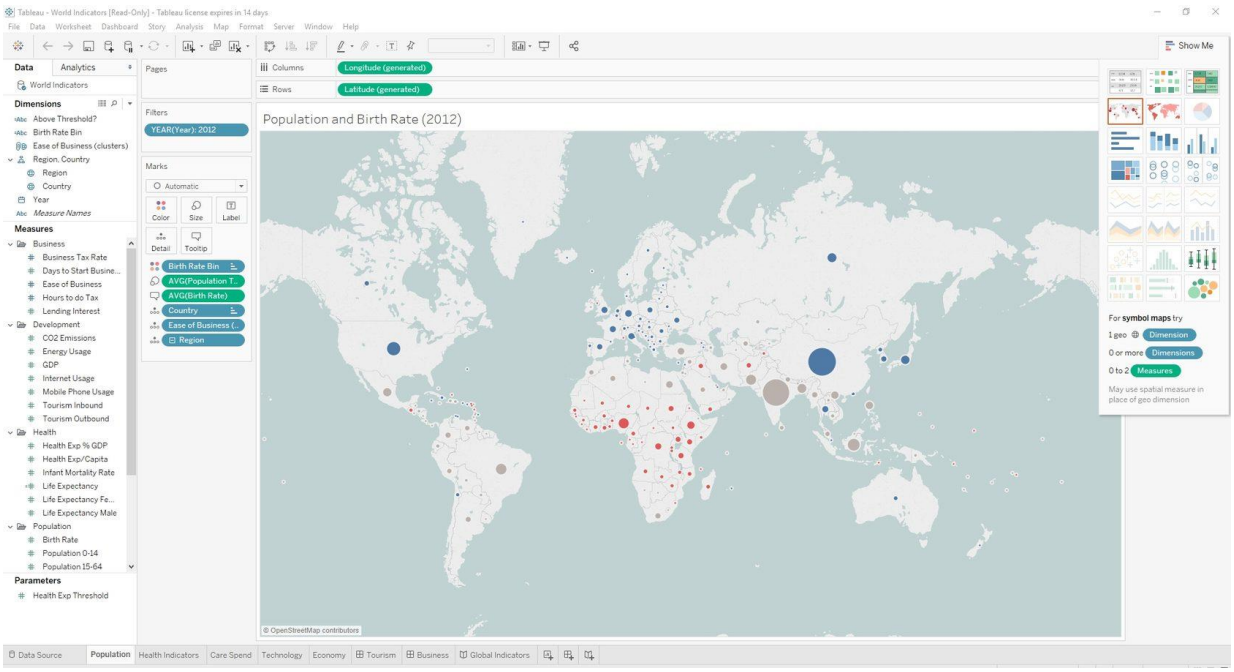


Рисунок 3.9 – Інтерфейс програми і зовнішній вигляд інформаційної панелі

Програмний продукт Tableau Desktop від компанії Tableau Software дозволяє представити сирі дані в легкому для читання форматі для бізнес-аналітики (BI), на підставі якого легше приймати зважене рішення. Інтерактивний, візуальний аналіз дозволяє визначити неочевидні залежності.

Програмне забезпечення Tableau Desktop (рис. 3.10) надає наступні можливості: побудова складних обчислень з існуючих даних, перетягування опорних ліній і прогнозів, виявлення тенденцій, перегляд статистичних зведень.

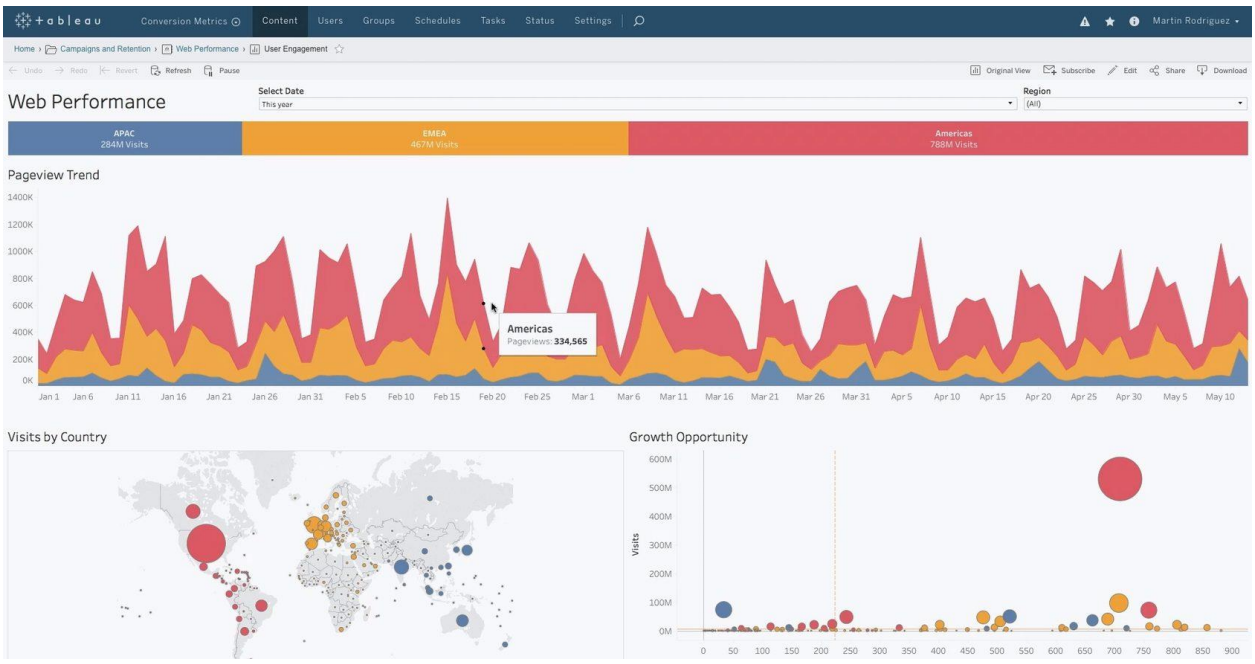


Рисунок 3.10 – Приклад дашборда

Гнучка настройка відображення даних дозволяє сформувати власне уявлення з аналізом тенденцій, регресій і кореляцій.

Програма Tableau Desktop надає можливість гнучкого доступу до інформації, в залежності від корпоративної архітектури та екосистеми даних, підключаючись до даних, що зберігаються на сервері або в хмарі із здійсненням оперативних запитів або готових панелей інформації. Програмний продукт можна розгорнути на Windows, Linux або macOS.

Tableau Online пропонує такі переваги:

– легкий запуск – програма готова до роботи через кілька хвилин;

– не вимагає установки – не потрібно налаштовувати сервери, оновлювати програмне забезпечення або масштабувати ємність обладнання;

– працює на всіх пристроях – немає вимог до обладнання або софту;

– зручне спільне використання – аналітична система, яка повністю розміщена в хмарі;

– легко додавати користувачів і ділитися результатами;

– працює в будь-якому місці – підключення до будь-яких даних, спільне створення і редагування;

– система безпеки – є можливість налаштовувати доступ і дозволу до документів.

Tableau Server – програмне рішення, яке дає можливість помістити аналітику на web-браузері і зробити її доступною будь-якому користувачеві. Рішення розгортається за хвилини, просто підтримується і робить спільну аналітичну роботу швидкої і легкої.

Tableau Server гарантує безпечний доступ до інформації. У програмі створена гнучка система імен користувачів і паролів. Окремі звіти можуть бути опубліковані з вкладеним паролем, дозволяючи користувачам мережі отримати доступ до цих звітів без додаткової авторизації. В якості альтернативи, Tableau Server може бути об’єднаний з Microsoft Active Directory, щоб використовувати імена облікових записів організації [39].

Tableau Server – це повне рішення. Для роботи з ним не потрібно інше програмне забезпечення. Єдині вимоги до інфраструктури - Microsoft Windows, щонайменше один користувач Tableau Desktop Professional і веб-браузер.

Tableau Server (рис. 3.11) пропонує такі переваги:

– високу ступінь захищеності – створюється довірена середа;

– зручність у роботі – просто розгортати, масштабувати і інтегрувати;

– безпечно підключення до всіх джерел даних – програма працює з базами даних і хмарними сервісами через власний коннектор і API;

– підключається до популярних корпоративних систем – Cloudera Hadoop, Oracle, AWS Redshift, cubes, Teradata, Microsoft SQL Server і іншим;

– централізоване управління метаданими і правилами безпеки;

– легко інтегрується з існуючими протоколами безпеки будь-яких таблиць, наприклад, Active Directory, Kerberos, OAuth;

– гнучке розгортання – встановлюється локально на Windows або Linux, розгортається в хмарі за допомогою AWS, Azure або Google Cloud Platform.

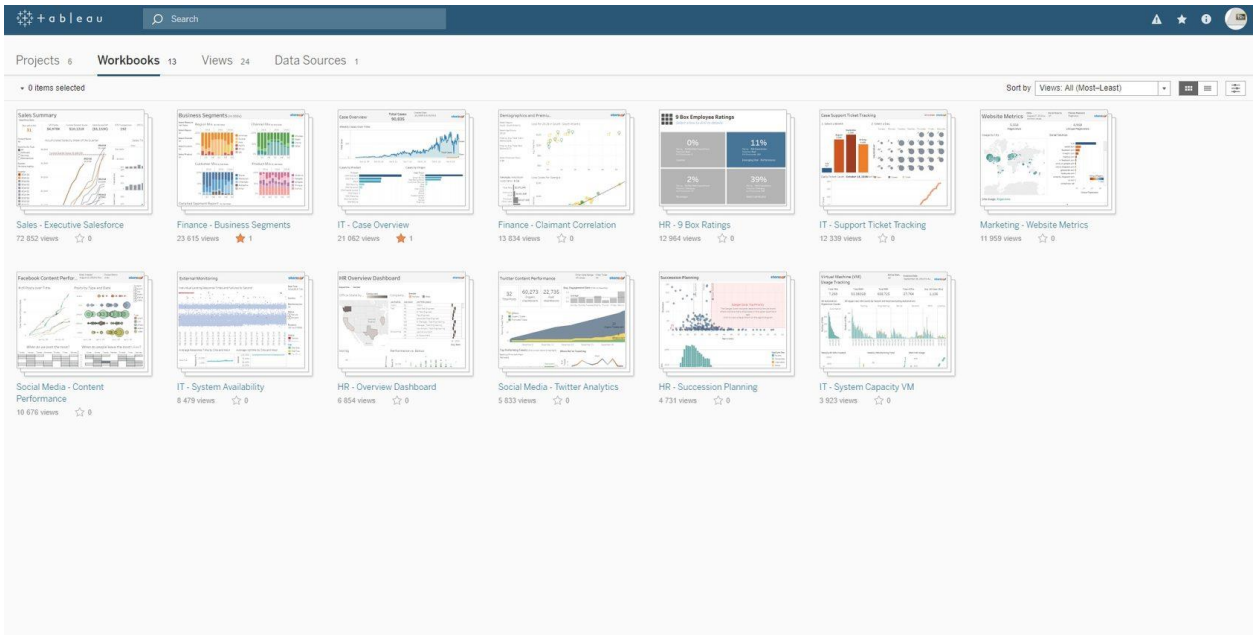


Рисунок 3.11 – Інтерфейс Tableau Server

Tableau надає рішення для всіх видів галузей, відділів і середовищ передачі даних. Нижче наведені деякі унікальні функції, які дозволяють Tableau обробляти різні сценарії:

– швидкість аналізу – оскільки він не вимагає високого рівня знань в області програмування, будь-який користувач, який має доступ до даних, може почати використовувати його для отримання цінності з даних;

– Self-Reliant – Tableau не вимагає складної настройки програмного забезпечення. Версія для ПК, яка використовується більшістю користувачів, легко встановлюється і містить всі функції, необхідні для запуску і завершення аналізу даних;

– візуальне виявлення – користувач досліджує і аналізує дані за допомогою візуальних інструментів, таких як кольори, лінії тренду, діаграми і графіки;

– змішування різноманітних наборів даних (рис. 3.12). Tableau дозволяє змішувати різні реляційні, напівструктуровані і необроблені джерела даних в режимі реального часу без дорогих витрат на попередню інтеграцію. Користувачам не потрібно знати деталі того, як зберігаються дані;

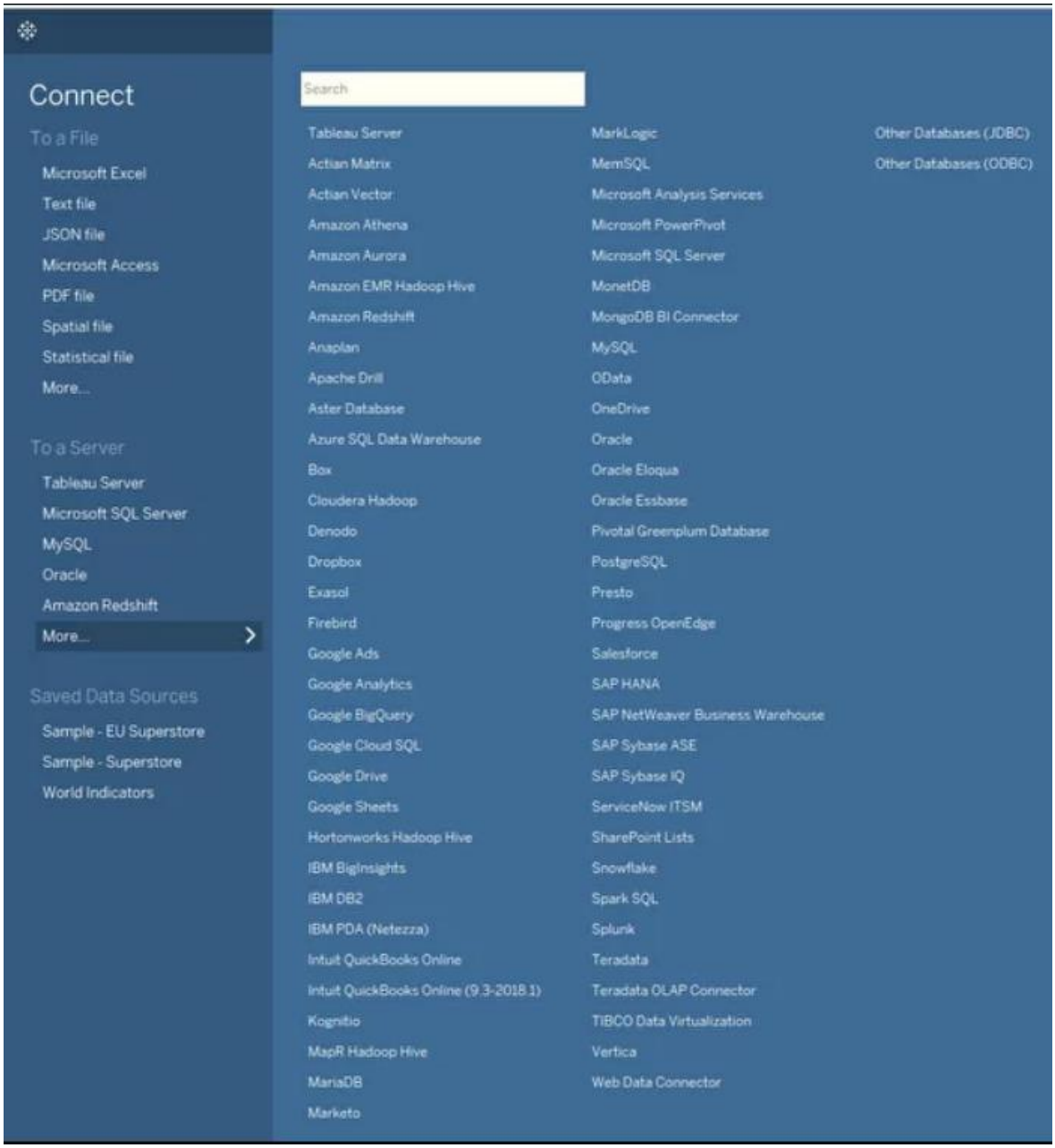


Рисунок 3.12 – Список можливих джерел Tableau

– Agnostic Architecture – Tableau працює у всіх видах пристроїв, де передаються дані. Отже, користувачеві не потрібно турбуватися про конкретні

вимоги до обладнання або програмного забезпечення для використання Tableau;

– спільна робота в реальному часі – Tableau може фільтрувати, сортувати і обговорювати дані на льоту, а також вбудувати живу панель моніторингу в такі портали, як сайт SharePoint або Salesforce. Є можливість зберегти своє уявлення даних і дозволити колегам підписуватися на інтерактивні інформаційні панелі, щоб вони могли бачити найостанніші дані, оновивши свій веб-браузер;

– централізовані дані – сервер Tableau забезпечує централізоване розташування для управління всіма опублікованими джерелами даних організації. Є можливість видаляти, змінювати дозволу, додавати теги і управляти розкладами в одному зручному місці. Легко запланувати поновлення екстрактів і управляти ними на сервері даних. Адміністратори можуть централізовано визначати розклад для витягів на сервері як для інкрементного, так і для повного оновлення.

3.5 Студія машинного навчання Microsoft Azure

Студія машинного навчання Microsoft Azure – це інструмент для спільної роботи, що підтримує функцію перетягування об’єктів і призначений для створення, тестування і розгортання рішень для прогнозного аналізу даних. Студія машинного навчання публікує моделі як веб-служби, які потім можна використовувати в зручних для користувача додатках і засобах бізнес-аналітики (наприклад, в Excel) [40].

Студія машинного навчання – це місце, де перетинаються обробка і аналіз даних, прогнозна аналітика, хмарні ресурси і дані клієнта.

Для розробки моделі прогнозної аналітики зазвичай використовуються дані з одного або декількох джерел. Для отримання набору результатів ці дані перетворюються і аналізуються за допомогою різних операцій і статистичних

функцій. Така розробка моделі – це ітеративний процес. Шляхом зміни різних функцій і їх параметрів виконується зведення результатів, поки не буде отримана підготовлена ефективна модель.

Студія машинного навчання Azure надає інтерактивне візуальне робочий простір, що спрощує створення, тестування і виконання ітерацій моделі прогнозної аналітики. Для виконання ітерацій в макеті моделі слід відредагувати експеримент, при необхідності зберегти копію і виконати його знову. Навчальний експеримент є можливість перетворити в прогнозний, а потім опублікувати його як веб-службу, щоб модель стала доступна іншим користувачам (рис. 3.13).

Експеримент складається з наборів даних, що надають дані для модулів аналітики, які з’єднуються один з одним для створення моделі прогнозної аналітики. Зокрема, правильний експеримент має такі характеристики:

– у експерименту принаймні один набір даних і один модуль;

– набори даних можуть бути пов’язані тільки з модулями;

– модулі можуть бути пов’язані або з наборами даних, або з іншими модулями;

– всі вхідні порти для модулів повинні мати зв’язок з потоком даних;

– всі необхідні параметри для кожного модуля повинні бути встановлені.

Експеримент можна створити з нуля або на основі існуючого експерименту в якості шаблону.

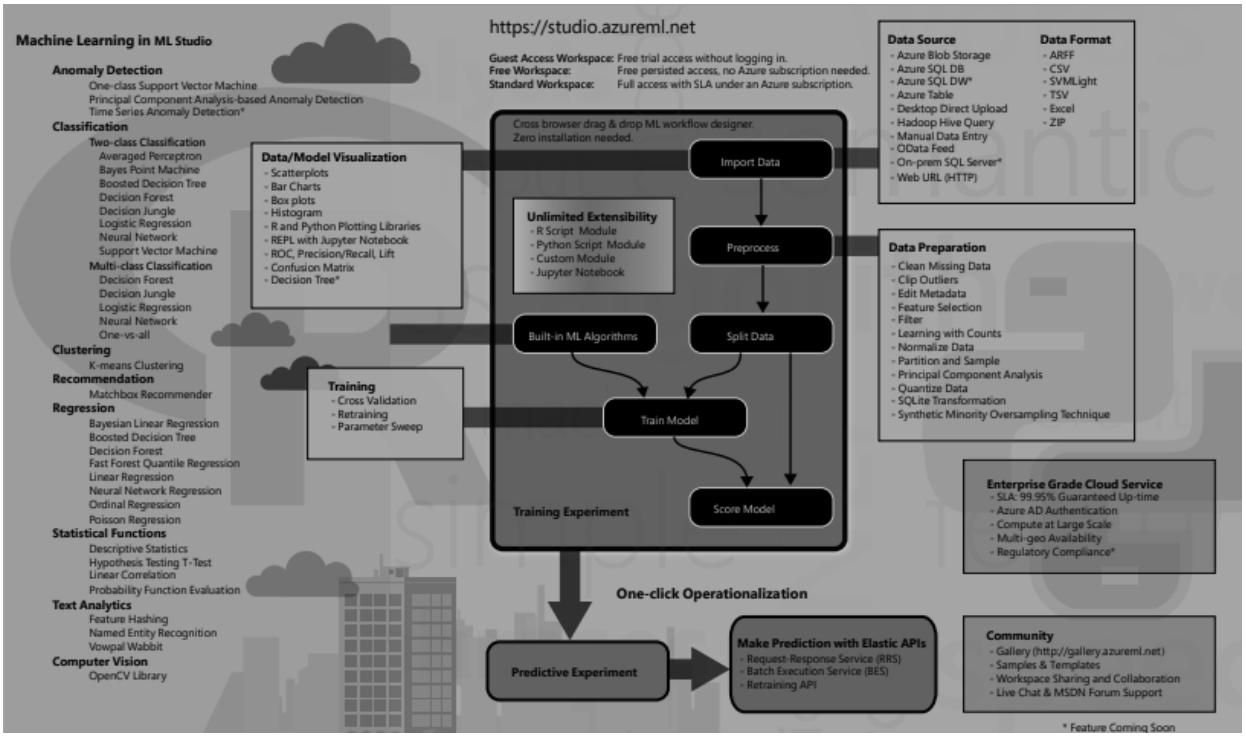


Рисунок 3.13 – Можливості Студії машинного навчання Microsoft Azure

Модуль – це алгоритм, який можна виконувати з даними. В Студії машинного навчання є ряд модулів, починаючи з функцій введення даних для процесів навчання, оцінки та перевірки. Нижче наведені приклади включених модулів:

– перетворення в ARFF – перетворює упорядкований набір даних .NET в формат ARFF;

– обчислення елементарної статистики – виконує елементарні статистичні обчислення, такі як отримання математичного очікування, середньоквадратичного відхилення і ін.;

– лінійна регресія – створює інтерактивний лінійно-регресійний аналіз на основі градієнтного спуску;

– модель оцінки – виконує оцінку навченої моделі класифікації або регресії.

Процес побудови алгоритму машинного навчання (рис 3.14) складається з таких частин:

– визначення мети. Всі алгоритми машинного навчання марні без явно-визначеної мети проведення експерименту;

– збір даних. Під час цього етапу формується вибірка даних, необхідна для подальшого навчання моделі;

– підготовка даних. На цьому етапі проводиться підготовка даних шляхом формування характеристик, видалення викидів і поділу вибірки на навчальну і тестову;

– розробка моделі. У процесі розробки моделі проводитися вибір одного або декількох моделей даних і відповідних алгоритмів навчання, які на думку розробника повинні будуть дати необхідний результат. Часто цей процес поєднаний з паралельним дослідженням ефективності декількох моделей і візуальним аналізом даних з метою відшукання будь-яких закономірностей;

– навчання моделі. Під час навчання алгоритм навчання здійснює пошук прихованих закономірностей у вибірці даних з метою відшукання способу передбачення. Сам процес пошуку визначається обраною моделлю і алгоритмом навчання;

– оцінка моделі. Після того як модель навчена необхідно досліджувати її прогностичні характеристики. Найчастіше для цього її проганяють на тестовій вибірці і оцінюють отриманий рівень помилки. Залежно від цього і вимог до точності модель може бути як прийнята в якості підсумкової, так і вироблено повторне навчання після додавання нових вхідних характеристик або навіть зміни алгоритму навчання;

– використання моделі. У разі успішного тестування навченої моделі настає стадія її використання. І це той випадок, коли Azure ML стає незамінний, даючи всі необхідні інструменти для публікації, моніторингу та монетизації алгоритмів.

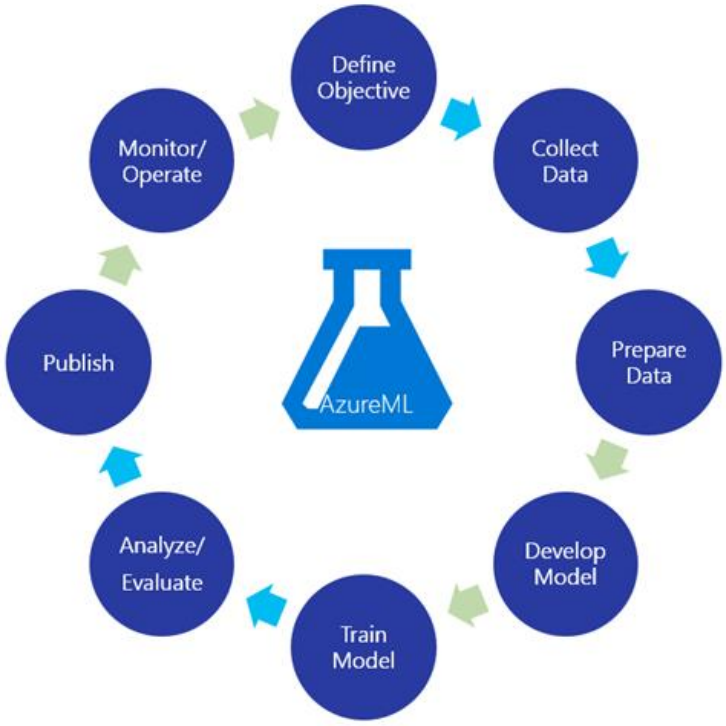


Рисунок 3.14 – Процес побудови алгоритму машинного навчання в Microsoft Azure

3.6 Платформа Kaggle

Kaggle (рис. 3.15) – платформа для змагань з аналітики та передбачуваного моделювання, в рамках якого статисти та добувачі даних конкурують у створені найкращої моделі для прогнозування та опису даних, запропонованих компаніями або користувачами. Цей краудсорсинговий підхід ґрунтується на тому, що є безліч стратегій, які можуть бути застосовані до будь-якого завдання з передбачуваного моделювання, і наперед не відомо, яка методика або аналітичний підхід буде найбільш ефективним.

Головні компоненти Kaggle:

– набори даних: безліч наборів даних різних типів і розмірів, які можна безкоштовно завантажити. Тут можна знайти цікаві дані для вивчення або тестування своїх навичок моделювання;

– змагання по машинному навчання: колись були серцем Kaggle, такі тести на моделювання – кращий спосіб вивчити нові види машинного навчання і відточити здібності за допомогою цікавих проблем, заснованих на реальних даних;

– вивчення: серія навчальних матеріалів по вивченню даних, що охоплюють SQL і глибоке навчання, які подаються в Jupyter Notebooks;

– обговорення: місце, де можна задати свої питання і отримати поради від тисяч експертів з аналітичними даними в співтоваристві Kaggle;

– ядра: онлайн-середовище для програмування, яка працює на серверах Kaggle. У ній можна писати Python / R-скрипти і працювати в Jupyter Notebooks. Такі ядра абсолютно і ідеальні для тестування: не потрібно налаштовувати середу у себе на комп'ютері. Їх можна використовувати для аналізу будь-якого набору даних, змагань по машинному навчання або виконання завдань з розділу «Навчання».

Рисунок 3.15 – Принцип змагань на Kaggle

3.7 Розгляд предметної області

В роботі розглядатимуться таблиці, що містять дані про затримки та скасування рейсів авіакомпаній. Дана предметна область була обрана тому, що обробка подібних даних є доволі актуальною задачею, бо кількість авіаперевезень з кожним роком стає все більшим та оброблювати їх стандартними методами не завжди є можливим.

Також важливим фактором є те, що нові дані про затримки та скасування рейсів з’являються кожної хвилини. Таким чином є потреба мати можливість оброблювати їх в реальному часі та в великій кількості. Класифікація в даному випадку є досить доречним рішень.

Які питання може вирішити класифікація даних про затримки та скасування рейсів? Насамперед:

– отримання відомостей про кількості затримок та скасувань в залежності від дня тижня, часу, погодних умов та інше;

– отримання відомостей про інтервал часу, на який затримується літак на виліт чи посадку в залежності від дня тижня, часу, погодних умов та інше;

– узгодження між собою рейсів, в уникнення ланцюгів запізнень;

– узгодження роботи трансферів, вивантаження та завантаження багажу;

– для ефективного розподілу пасажирів в середині аеропорту (на стойках паспортного контролю та пунктах отримання багажу).

Класифікація використовується в великій кількості областей діяльності людини. В авіаперевезеннях класифікація причин та часу затримок полегшує аеропортам роботу над ефективністю та безперебійністю свого функціонування. Це може значно скоротити час проведення пасажирів в аеропортах, збільшення пропускної кількісті літаків та допомогти в більш ефективному розподілу витрат.

В роботі розглядається аналіз і обробка інформації про рейси. Для цього було розроблено метод на основі байєсівського класифікатора.

3.8 Опис даних, що використовувались для класифікації

Дані були взяті з платформи Kaggle. Перша таблиця містить в собі близько 11 мільйонів записів про затримки та відміну рейсів у період з січня по червень 2020 року, поля, що містить таблиця наведено в таблиці 3.1.

Таблиця 3.1 – Похідні дані для дослідження

№	Назва стовпчика	Опис
1.	year	рік
2.	quarter	квартал
3	month	місяць року
4.	day_of_month	день місяця
5.	day_of_week	день тижня
6.	fl_date	повна дата польоту
7.	mkt_unique_carrier	код авіакомпанії
8.	mkt_carrier_fl_num	номер рейсу
9.	tail_num	номер хвоста літака
10.	origin	скорочення аеропорту, з якого вилітає літак
11.	origin_city_name	місто вильоту
12.	origin_state_abr	скорочення штату рейсу, що вилітає
13.	origin_state_nm	назва штату рейсу вильоту
14.	dest	скорочення аеропорту прибуття рейсу
15.	dest_city_name	місто прибуття рейсу
16.	dest_state_abr	скорочення штату прибуття рейсу
17.	dest_state_nm	назва штату прильоту рейсу
18.	crs_dep_time	запланований час відправлення
19.	dep_time	фактичний час відправлення
20.	dep_delay	затримка відправлення (різниця між фактичним часом відправлення та запланованим у хвилинах)
21.	dep_delay_new	затримка вильоту з ігноруванням ранніх виїздів
22.	dep_del15	затримка виїзду більше 15 хвилин

Продовження таблиці 3.1

№	Назва стовпчика	Опис
23.	dep_time_blk	запланований час відправлення в погодинному блоці
24.	taxi_out	час між таксі літака від воріт і зльоту
25.	wheels_off	час зльоту літака
26.	wheels_on	час посадки літака
27.	taxi_in	час між таксі літака до воріт і посадкою
28.	crs_arr_time	запланований час прибуття
29.	arr_time	фактичний час прибуття
30.	arr_delay	затримка прибуття (різниця між фактичним часом прибуття та запланованим часом прибуття в хвилинах)
31.	arr_delay_new	затримка прибуття, ігноруючи ранні прибуття
32.	arr_del15	затримка прибуття більше 15 хвилин
33.	arr_delay_group	затримка прибуття за кількістю кроків по 15 хвилин
34.	arr_time_blk	запланований час прибуття в погодинному блоці
35.	cancelled	0: рейс не скасовано 1: рейс скасовано
36.	cancellation_code	причина скасування (a: перевізник, b: погода, c: національна авіаційна система, d: безпека)
37.	crs_elapsed_time	загальний запланований час польоту
38.	actual_elapsed_time	фактичний загальний час польоту
39.	air_time	фактичний загальний час польоту літака в повітрі
40.	distance	відстань між аеропортами вильоту и прибуття

Кінець таблиці 3.1

№	Назва стовпчика	Опис
41.	distance_group	відстань між аеропортами вильоту та прибуття із збільшення окружності до 250 миль
42.	carrier_delay	затримка оператора
43.	weather_delay	затримка погоди
44.	nas_delay	затримка національної авіаційної системи
45.	security_delay	затримка безпеки
46.	late_aircraft_delay	пізня затримка літака

Друга таблиця містить в собі близько 5 мільйонів записів про відміну рейсів у 2015 року, поля, що містить таблиця наведено в таблиці 3.2.

Таблиця 3.2 – Похідні дані для дослідження

№	Назва стовпчика	Опис
1.	year	Рік
2.	month	Місяць
3.	day	День
4.	day_of_week	день тижня
5.	airline	Авіакомпанія
6.	flight_number	номер рейсу
7.	tail_number	номер хвоста рейсу
8.	origin_airport	аеропорт вильоту рейсу
9.	destination_airport	аеропорт прибуття рейсу
10.	scheduled_departure	запланований виліт
11.	departure_time	фактичний час відправлення
12.	departure_delay	затримка відправлення

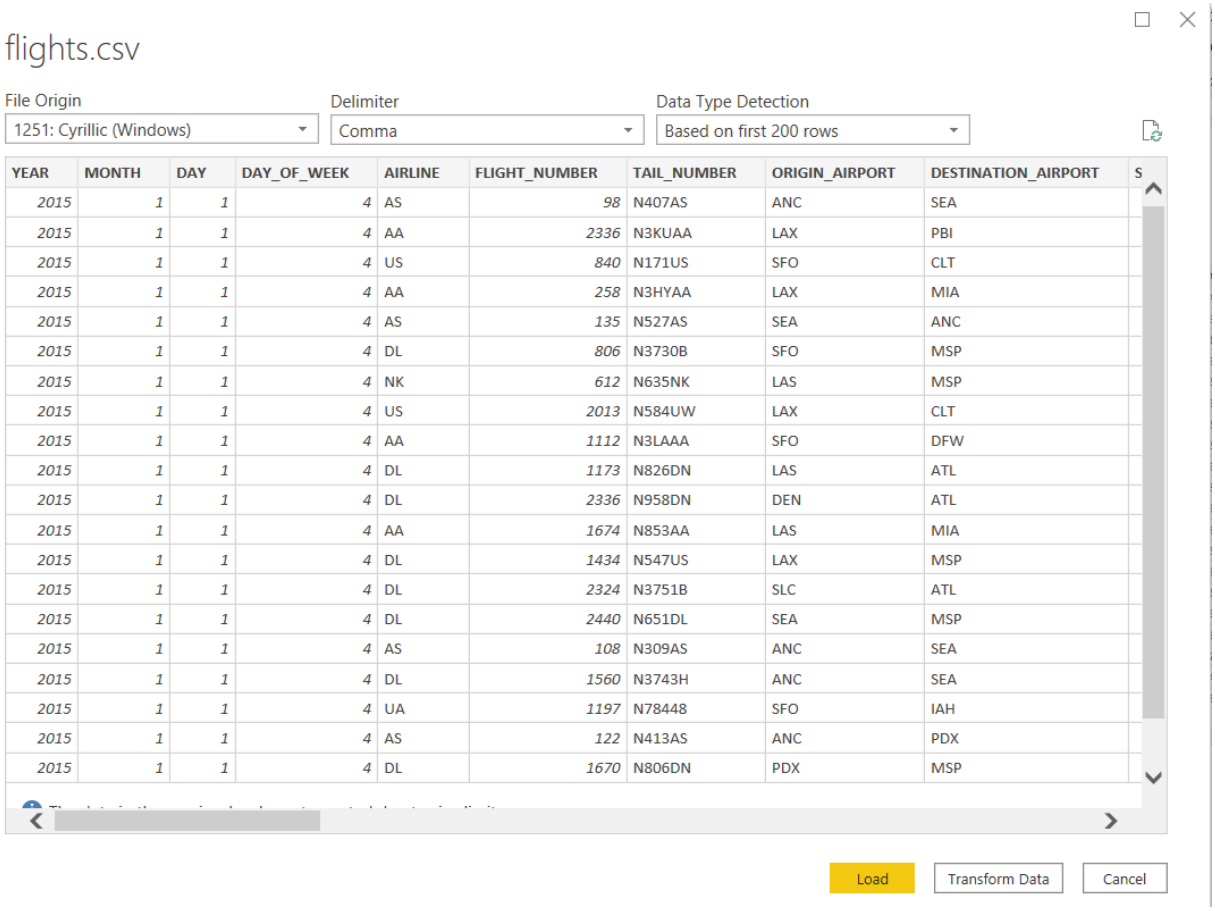
Продовження таблиці 3.2

№	Назва стовпчика	Опис
13.	taxi_out	час між таксі літака від воріт і зльоту
14.	wheels_off	час зльоту літака
15.	scheduled_time	запланований час
16.	elapsed_time	фатично витрачений час
17.	air_time	час польоту
18.	distance	відстань
19.	wheels_on	час посадки літака
20.	taxi_in	час між таксі літака до воріт і посадкою
21.	scheduled_arrival	запланований час прибуття
22.	arrival_time	фактичний час прибуття
23.	arrival_delay	затримка прибуття
24.	diverted	перенаправлення рейсу
25.	cancelled	0: рейс не скасовано 1: рейс скасовано
26.	cancellation_reason	причина відміни рейсу
27.	air_system_delay	затримка повітряної системи
28.	security_delay	затримка безпеки
29.	airline_delay	затримка авіакомпанії
30.	late_aircraft_delay	пізня затримка літака
31.	weather_delay	затримка погоди

3.9 Реалізація методу на основі байєсівського класифікатора на мові R

У рамках дослідження було реалізовано власний метод, на основі байєсівського класифікатора. Для реалізації та візуалізації було обрано середовище Power BI Desktop. Це обумовлено тим, що Power BI є сучасним та потужним середовищем для аналізу даних.

На початку було створено новий простір та завантажено таблиці з даними. Для цього натиснули на значок «Get data», далі натиснули More та обрали отримати дані з CSV-файлу. Обрали потрібний нам файл та після підключення Power BI надав можливість попередньо переглянути завантажені дані (рис. 3.16) та додати їх до створеного простору.



YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	S
2015	1	1	4	AS	98	N407AS	ANC	SEA	
2015	1	1	4	AA	2336	N3KUAA	LAX	PBI	
2015	1	1	4	US	840	N171US	SFO	CLT	
2015	1	1	4	AA	258	N3HYAA	LAX	MIA	
2015	1	1	4	AS	135	N527AS	SEA	ANC	
2015	1	1	4	DL	806	N3730B	SFO	MSP	
2015	1	1	4	NK	612	N635NK	LAS	MSP	
2015	1	1	4	US	2013	N584UW	LAX	CLT	
2015	1	1	4	AA	1112	N3LAAA	SFO	DFW	
2015	1	1	4	DL	1173	N826DN	LAS	ATL	
2015	1	1	4	DL	2336	N958DN	DEN	ATL	
2015	1	1	4	AA	1674	N853AA	LAS	MIA	
2015	1	1	4	DL	1434	N547US	LAX	MSP	
2015	1	1	4	DL	2324	N3751B	SLC	ATL	
2015	1	1	4	DL	2440	N651DL	SEA	MSP	
2015	1	1	4	AS	108	N309AS	ANC	SEA	
2015	1	1	4	DL	1560	N3743H	ANC	SEA	
2015	1	1	4	UA	1197	N78448	SFO	IAH	
2015	1	1	4	AS	122	N413AS	ANC	PDX	
2015	1	1	4	DL	1670	N806DN	PDX	MSP	

Рисунок 3.16 – Завантаження тестових даних

Перед нами представлене чисте полотно нового простору з вже підключеними таблицями даних (рис. 3.17). Центральне частина вікна є областю для візуалізації проаналізованих даних. Для цього можна використовувати як вже готові рішення так і підключати власні розробки. Для цього є панель Visualizations. Нами було обрано варіант з підключенням власних розробок, а саме R script visual.

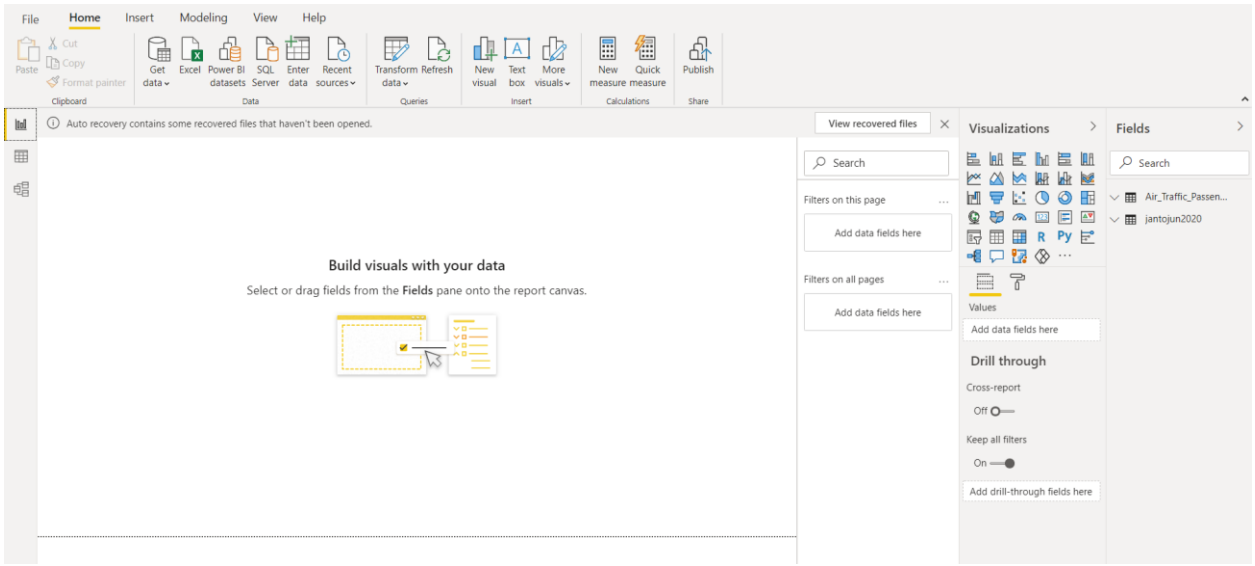


Рисунок 3.17 – Стартове вікно нового простору

Після вибору R script visual з’являється панель R script editor (рис. 3.18). Куди необхідно вставити файл з прописаним скриптом для класифікації даних.

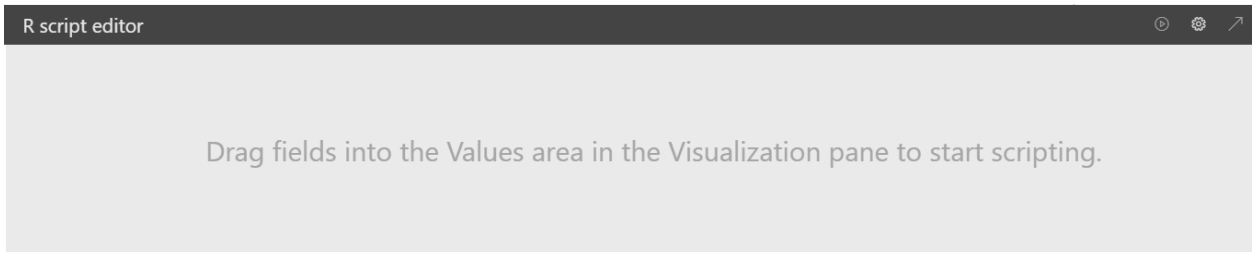


Рисунок 3.18 – Панель R script editor

Далі вставили файл с прописаним скриптом розробленого методу (рис. 3.19) в Power BI.

Рисунок 3.19 – Скрипт розробленого методу

Признаками для класифікації було обрано запланований час прибуття та фактичний час прибуття. Кількість класів було обрано 5:

– перший клас – запланований час та фактичний час співпадають;

– другий клас – запланований час та фактичний відрізняються до 6 голин;

– третій клас – запланований час та фактичний відрізняються до 2 годин;

– четвертий клас – запланований час та фактичний відрізняються до 30 хвилин.

– п’ятий клас – запланований час та фактичний відрізняються більше, ніж на 6 годин.

Після запуску скрипта було отримано графік, що ілюструє розподіл даних на класи методом, розробленим на основі байєсівського класифікатора (рис. 3.20).

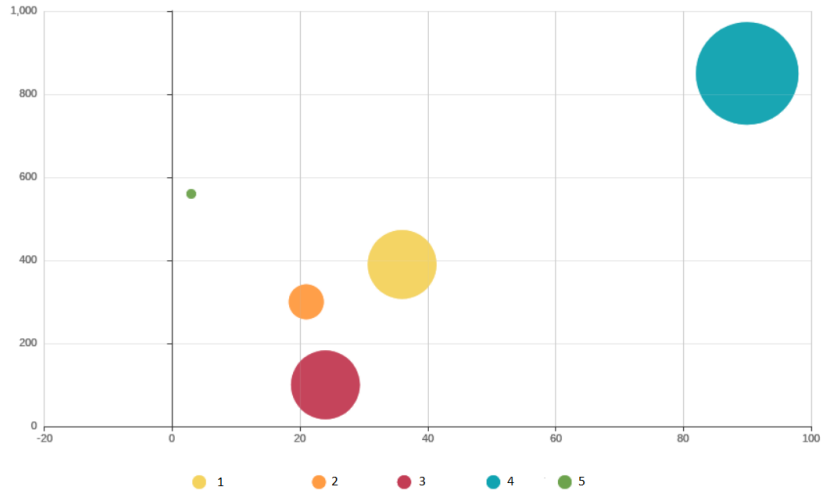


Рисунок 3.20 – Класифікація методом на основі байєсівського класифікатора для тестових даних з першої таблиці

Далі перевірили роботу алгоритму для даних другої таблиці. Для цього створили новий елемент візуалізації та так само запустили скрипт, написаний на мові R. Отримані результати зображені на рисунку 3.21.

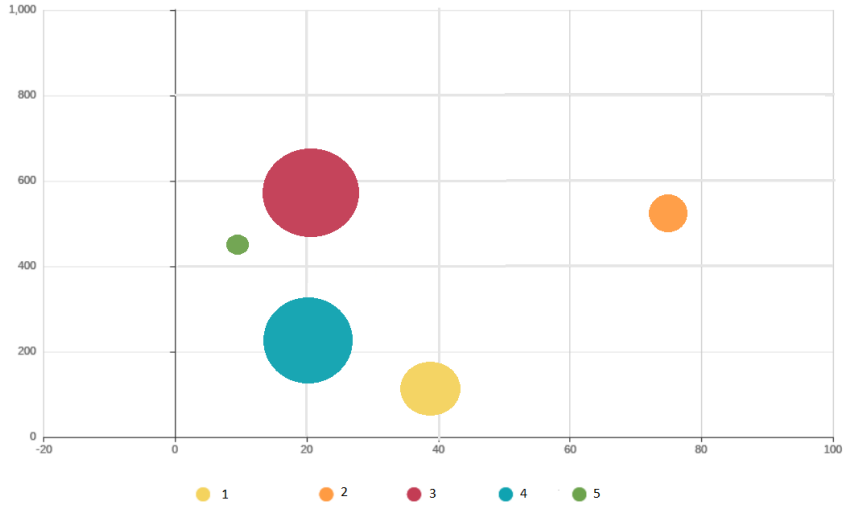


Рисунок 3.21 – Класифікація методом на основі байєсівського класифікатора для тестових даних з другої таблиці

Таким чином було отримано класи, розміри яких залежать від кількості даних, що входить в той чи інший клас.

ВИСНОВКИ

У рамках дослідження було проведено огляд відомих методів машинного навчання для класифікації великих обсягів даних, аналіз розглянутих методів дозволяє більш точно підібрати необхідний метод і тим самим значно підвищити швидкість та ефективність класифікації, що у свою чергу, значно підвищує ефективність подальшої роботи з даними.

Також було оглянуто перспективні сервіси для аналізу даних, такі як Power BI, Microsoft Azure та Tableau. Використання яких зменшує час на розгортання середовища. Також, використання подібних сервісів підходить для тестування розроблених методів, бо вони вже мають велику кількість вбудованих рішень, з яких є можливість вибрати під будь-яку задачу.

Для експериментальних досліджень було обрано байєсівський класифікатор. На основі нього було розроблено власний метод для більш ефективної класифікації великих обсягів даних. Для цього було використано середовище Power BI та його можливість підтримки та візуалізації власних розробок. Метод був написаний на мові R та засобами Power BI результати роботи методу були відображенні у вигляді бульбашкової діаграми. Розроблений метод вирішує задачу розподілу великого обсягу даних на класи. В рамках обраних нами даних, було вирішено розділити дані на п’ять класів.

Розроблений метод був протестований на двох наборах великих даних та показав позитивні результати класифікації. Він може використовуватися для класифікації числових статистичних даних для швидкого розподілу даних по класах, отриманих в режимі реального часу. Перевагою його використання також є швидка робота та відсутність необхідності перерозподілу даних по класам після надходження нових даних.

Результати даного дослідження було апробовано на 24-му Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Zadeh, L. A. (2015). Fuzzy logic—a personal perspective. Fuzzy sets and systems, 281, 4-20.

2. Шумейко, А. А., & Сотник, С. Л. (2012). Интеллектуальный анализ данных (введение в Data Mining). *Днепропетровск: Белая ЕА*, *212*.

3. Вискребенцева С.О., Кобилін О.А. (2019) Методи сегментації зображень. Матеріали XXIII міжнародного молодіжного форуму. Радіоелектроніка та молодь у XXI столітті, 19-20.

4. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)* (pp. 665-670). IEEE.

5. Работягов, А. В., Ляшенко, В. В., & Кобылин, О. А. (2016). Сегментация сложных изображений цитологических препаратов.

6. Lyashenko, V., Mohammad, A., & Kobylin, O. (2015). Experiments with Fusion of Images with Use of Wavelet Transformation in Problems of the Text Information Analysis.

7. Деркач, О. І. (2016). Аналітична обробка текстової інформації за допомогою засобів кластеризації. *Young*, *34*(7).

8. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, *5*(57).

9. Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons.

10. Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.

11. Perret, B., Chierchia, G., Cousty, J., Guimarães, S. J. F., Kenmochi, Y., & Najman, L. (2019). Higra: Hierarchical graph analysis. *SoftwareX*, *10*, 100335.

12. Steinley, D. (2006). K means clustering: a half century synthesis. *British Journal of Mathematical and Statistical Psychology*, *59*(1), 1-34.

13. Ackermann, M. R. (2009). *Algorithms for the Bregman k-Median problem* (Doctoral dissertation, University of Paderborn).

14. Khachumov, M. V. (2012). Distances, metrics and cluster analysis. *Scientific and Technical Information Processing*, *39*(6), 310-316.

15. Huang, Z., & Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, *7*(4), 446-452.

16. Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons.

17. Zhang, J., Zhao, Z., Xue, Y., Chen, Z., Ma, X., & Zhou, Q. (2017). Time series analysis. *Handbook of Medical Statistics*, *269*.

18. Крашений, І. Е., Попов, А. О., Рамірез, Х., Горріз, Х. М., Крашеный, И. Э., Попов, А. А., ... & Горриз, Х. М. (2016). Використання методів кластеризації в системах нечіткого виводу для діагностики хвороби Альцгеймера на основі ПЕТ-зображень.

19. Штовба, С. Д. (2006). Побудова функцій належності нечітких множин за кластеризацією експериментальних даних. *Інформаційні технології та комп'ютерна інженерія*, (2), 92-95.

20. Xu, J., Han, J., Xiong, K., & Nie, F. (2016, July). Robust and Sparse Fuzzy K-Means Clustering. In *IJCAI* (pp. 2224-2230).

21. Gorshkov, Y., Kolodyazhniy, V., & Bodyanskiy, Y. (2009, June). New recursive learning algorithms for fuzzy Kohonen clustering network. In *Proc. 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems* (pp. 58-61).

22. Bodyanskiy, Y. V., Tyshchenko, O. K., & Mashtalir, S. V. (2019, June). Fuzzy Clustering High-Dimensional Data Using Information Weighting. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 385-395). Springer, Cham.

23. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering

via Multidimensional Time-Series Analysis. In *Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60- 63). ACM.

24. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2017). Video shot boundary detection via sequential clustering. *International Journal "Information Theories and Applications*, *24*(1), 50-59.

25. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2018, August). Representative Based Clustering of Long Multivariate Sequences with Different Lengths. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 545-548). IEEE.

26. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)* (pp. 930-935). IEEE.

27. Бодянський, Є. В., Дейнеко, А. О., & Куценко, Я. В. (2016). Послідовне нечітке кластерування на основі нейро-фаззі підходу. *Радіоелектроніка, інформатика, управління*, (3 (38)).

28. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. *Information Technology and Management Science*, *19*(1), 23-28.

29. Schafer, J. L., & Graham, J. W. (2002). Missing data: our view of the state of the art. *Psychological methods*, *7*(2), 147.

30. Волкова, В. В., & Шафроненко, А. Ю. (2011). Нечітка кластеризація масивів даних з пропущеними значеннями. *Індуктивне моделювання складних систем*.

31. Kesemen, O., Tezel, Ö., & Özkul, E. (2016). Fuzzy c-means clustering algorithm for directional data (FCM4DD). *Expert systems with applications*, *58*, 7682.

32. Женбинг, Х., Бодянский, Е. В., Тыщенко, А. К., & Ткачев, В. Н.

(2017). Fuzzy Clustering Data Arrays with Omitted Observation. Kate, R. J. (2016). Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery*, *30*(2), 283-312.

33. Hu, Z., Mashtalir, S. V., Tyshchenko, O. K., & Stolbovyi, M. I. (2018). Clustering matrix sequences based on the iterative dynamic time deformation procedure. *International Journal of Intelligent Systems and Applications*, *10*(7), 66-73.

34. Wang, D., Lu, X., & Rinaldo, A. (2017). DBSCAN: Optimal Rates For Density Based Clustering. *arXiv preprint arXiv:1706.03113*.

35. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. International Journal of Advanced Trends in Computer Science and Engineering. 9(4). – 4701-4706.

36. Bodyanskiy, Y., Shafronenko, A., & Mashtalir, S. (2019, May). Online Robust Fuzzy Clustering of Data with Omissions Using Similarity Measure of Special Type. In *International Scientific Conference "Intellectual Systems of Decision Making and Problem of Computational Intelligence"* (pp. 637-646). Springer, Cham.

37. Mashtalir, S. V., Stolbovyi, M. I., & Yakovlev, S. V. (2019). Clustering Video Sequences by the Method of Harmonic k-Means. *Cybernetics and Systems Analysis*, *55*(2), 200-206.

38. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). Advances in Spatio-Temporal Segmentation of Visual Data (Vol. 876). Springer Nature.

39. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.

40. Hu, Z., Bodyanskiy, Y. V., Tyshchenko, O. K., & Shafronenko, A. (2019, July). Fuzzy clustering of incomplete data by means of similarity measures. In 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON) (pp. 957-960). IEEE.