СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра програмування та математики

Пояснювальна записка

До магістерської дипломної роботи

Магістр

(освітньо-кваліфікаційний рівень)

на тему «Хмарна інформаційна технологія класифікації неоднорідних даних для прогнозування серцево-судинних захворювань»

Виконав: студент _2_ курсу, групи ІСТ-21дм спеціальності

126 «Інформаційні системи та технології»

(шифр і назва спеціальності)

Мироненко М.В.

(прізвище та ініціали)

Керівник Захожай О.І.

(прізвище та ініціали)

Рецензент Кряжич О.О.

(прізвище та ініціали)

Сєвєродонецьк – 2022 рік

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки
Кафедра програмування та математики
Освітньо-кваліфікаційний рівень магістр

спеціальність 126 «Інформаційні системи та технології»

(шифр і назва спеціальності)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,
д.т.н., доцент

_____д.т.н., доц. Лифар В.О.

(підпис)

«___»_____2022р.

ЗАВДАННЯ

на магістерську дипломну роботу студенту

Мироненко Микола Володимирович

(прізвище, ім'я, по батькові)

1. Тема роботи Хмарна інформаційна технологія класифікації неоднорідних даних для прогнозування серцево-судинних захворювань.

керівник роботи доцент, к.т.н Захожай Олег Ігорович ,

(вчене звання, науковий ступінь, прізвище, ім'я, по батькові)

затверджений наказом університету від «_30_»_11_____2022 року №182/15.16

2. Строк подання студентом роботи 20 грудня 2022 р.
3. Вихідні дані до роботи Матеріали науково-дослідної практики, науково-методична література; дані інтернет-мережі; .

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Вступ
4.2 Аналітичний огляд питання (огляд публічних джерел інформації
4.3 Основна частина, в якої висвітлити методи, які будуть використовуватися для реалізації проєкту та алгоритму c-means.
4.4 Практична частина – огляд технологій, які використовуються під час реалізації проєкту.
4.4 Висновки
4.5 Перелік використаних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 8 листопада 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	8.10.2022	
2	Укладання і погодження з керівником плану і етапів виконання роботи	10.10.2022	
3	Узагальнення даних літературних джерел	13.10.2022	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху виконання завдання	17.10.2022	
5	Аналіз технічних засобів та існуючих систем	25.10.2022	
6	Реалізація практичної частини завдання	01.11.2022	
7	Укладання, оформлення та погодження пояснювальної записки з керівником	08.11.2022	
8	Здача пояснювальної записки на кафедру	20.11.2022	
9	Підготовка доповіді та презентації	27.11.2022	

Студент _____ Мироненко М.В._____
(підпис) (прізвище та ініціали)

Керівник роботи _____ Захожай О.І._____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Робота містить: 47 сторінок основного тексту, 3 сторінки додатків, 12 малюнків, 15 використаних посилань.

Метою магістерської дипломної роботи є Хмарної інформаційної технології класифікації неоднорідних даних для прогнозування серцево-судинних захворювань.

Був проведений детальний аналіз питання та методів його вирішення. Багато часу було приділено аналізам алгоритмів кластерного аналізу та їх використання в аналізі текстових даних.

В результаті виконаної роботи було розглянуто створення хмарної інформаційної технології класифікації неоднорідних даних для прогнозування серцево-судинних захворювань.

Результати даного проєкту можуть використовуватися в подальшій роботі в двох напрямках: побудові інформаційної системи автоматичної обробки даних хворих та подальшого прогнозування наявності в них серцево-судиннії захворювань та в навчанні нейромереж для визначення точних серцево-судинних захворювань у людини за різними ознаками.

Зроблено опис та розбір технологій, платформ та алгоритмів хмарного машинного навчання.

Зміст

ВСТУП

Актуальність.Серцево-судинні захворювання, які здебільшого спричинені хворобами серця, спричиняють понад одну третину всіх щорічних смертей у світі. В Україні серцево-судинні захворювання є головною причиною смертності населення. За цим показником наша країна лишається одним зі світових лідерів.

Використання хмарних інформаційних технологій для прогнозування захворювань може допомогти прийняти клінічні рішення щодо медичного діагнозу, прискорити процес діагностики, які врятують життя більшій кількості людей. Також може надати змогу кожній людині самій перевірити себе на вірогідність захворювання, просто надавши певні про свій образ життя, звички тощо.

Під час встановлення діагнозу захворювання є велика кількість інформації про патологію пацієнта, яка виражається в наборі даних у вигляді достатньої кількості ознак. Кожна ознака має унікальний вплив на результати діагностики захворювання. Часто кілька основних ознак сприяють діагностиці наявності або відсутності захворювання. Застосування методів вибору ознак перед навчанням моделі може допомогти з вибором ключових характеристик, що призведе до хороших результатів прогнозування за коротший період часу.

Об’єкт дослідження: процес розробки хмарних інформаційних систем

Предмет дослідження: хмарна інформаційна система передбачення серцевих захворювань

Мета дослідження: Створення хмарної інформаційної технології

Завдання дослідження:

- розглянути теоретичні основи хмарних інформаційних систем та технологій
- розглянити види інформаційних систем
- дослідити та проаналізувати платформи розробки хмарних інформаційних систем
- дослідити та проаналізувати процес хмарних інформаційних систем
- дослідити технології класифікації неоднорідних даних
- зробити вибір мов програмування та технологій для програмної продукту
- розробити структуру

Методи дослідження: вивчення методів створення хмарних інформаційних технологій класифікації даних для прогнозування

1. РОЗДІЛ 1. АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ

1.1.Хмарні технології

Хмара — це віртуальний простір, який існує в Інтернеті. Це місце для зберігання, де люди можуть розміщувати свої цифрові ресурси, такі як програмне забезпечення, програми та файли. Простіше кажучи, ми можемо сказати, що хмара — це віртуальний простір для зберігання в Інтернеті.

Багато людей схильні плутати хмару з Інтернетом. Однак хмара – це лише одна частина Інтернету.

Технологія хмарних обчислень дозволяє людям використовувати цифрові ресурси, що зберігаються у віртуальному просторі за допомогою мереж – часто супутникових мереж. Це дозволяє людям обмінюватися інформацією та додатками, не обмежуючись їх фізичним місцезнаходженням.

Хмара дозволяє користувачам отримувати доступ до тих самих файлів і програм майже з будь-якого пристрою, оскільки обчислення та зберігання відбуваються на серверах у центрі обробки даних, а не локально на пристрої користувача. Ось чому користувач може увійти до свого облікового запису Instagram на новому телефоні після того, як його старий телефон зламався, і все одно знайти свій старий обліковий запис

на місці з усіма своїми фотографіями, відео та історією розмов. Він працює так само з хмарними постачальниками електронної пошти, як-от Gmail або Microsoft Office 365, і з хмарними сховищами, як-от Dropbox або Google Drive.

Для компаній перехід на хмарні обчислення позбавляє від деяких IT-витрат і накладних витрат: наприклад, їм більше не потрібно оновлювати та обслуговувати власні сервери, оскільки це робитиме постачальник хмарних технологій, якого вони використовують. Це особливо впливає на малий бізнес, який, можливо, не міг дозволити собі власну внутрішню інфраструктуру, але може аутсорсити свої потреби в інфраструктурі за доступною ціною через хмару. Хмара також може полегшити міжнародну діяльність компаній, оскільки співробітники та клієнти можуть отримати доступ до тих самих файлів і програм з будь-якого місця.

Принцип роботи хмарних обчислень

Хмарні обчислення можливі завдяки технології віртуалізації. Віртуалізація дозволяє створити змодельований цифровий «віртуальний» комп’ютер, який веде себе так, ніби це фізичний комп’ютер із власним апаратним забезпеченням. Технічний термін для такого комп'ютера - віртуальна машина. При правильному застосуванні віртуальні машини на одній хост-машині є ізольованими одна від одної, тому вони взагалі не взаємодіють одна з одною, а файли та програми з однієї віртуальної

машини не видимі для інших віртуальних машин, навіть якщо вони знаходяться на та сама фізична машина.

Віртуальні машини також ефективніше використовують апаратне забезпечення, на якому вони розміщені. Завдяки одночасному запуску кількох віртуальних машин один сервер стає багатьма серверами, а центр обробки даних перетворюється на цілу низку центрів обробки даних, здатних обслуговувати багато організацій. Таким чином, хмарні провайдери можуть запропонувати використання своїх серверів набагато більшій кількості клієнтів одночасно, ніж вони могли б це зробити в іншому випадку, і вони можуть зробити це за низькою ціною.

Навіть якщо окремі сервери виходять з ладу, хмарні сервери загалом мають бути завжди онлайн і завжди доступними. Хмарні постачальники зазвичай створюють резервні копії своїх послуг на кількох машинах і в кількох регіонах.

Користувачі отримують доступ до хмарних сервісів або через браузер, або через програму, підключаючись до хмари через Інтернет — тобто через багато взаємопов’язаних мереж — незалежно від того, який пристрій вони використовують.

1.1.1. Основні моделі обслуговування хмарних обчислень

Програмне забезпечення як послуга (SaaS): Замість того, щоб користувачі встановлювали програму на свій пристрій, програми SaaS розміщуються на хмарних серверах, і користувачі отримують до них

доступ через Інтернет. SaaS — це як оренда будинку: орендодавець обслуговує будинок, але орендар здебільшого може використовувати його так, ніби він є власником. Приклади програм SaaS включають Salesforce, MailChimp і Slack.

Платформа як послуга (PaaS): у цій моделі компанії не платять за розміщені програми; натомість вони платять за речі, необхідні для створення власних програм.

Постачальники PaaS пропонують все необхідне для створення програми, включаючи засоби розробки, інфраструктуру та операційні системи, через Інтернет. PaaS можна порівняти з орендою всіх інструментів і обладнання, необхідних для будівництва будинку, замість оренди самого будинку. Приклади PaaS включають Heroku і Microsoft Azure.

Інфраструктура як послуга (IaaS): у цій моделі компанія орендує необхідні сервери та сховище у хмарного постачальника. Потім вони використовують цю хмарну інфраструктуру для створення своїх програм. IaaS схоже на компанію, яка орендує ділянку землі, на якій вони можуть будувати все, що завгодно, але їм потрібно забезпечити власне будівельне обладнання та матеріали. Постачальники IaaS включають DigitalOcean, Google Compute Engine і OpenStack.

Раніше SaaS, PaaS та IaaS були трьома основними моделями хмарних обчислень, і, по суті, усі хмарні сервіси належали до однієї з цих категорій. Однак в останні роки з'явилася четверта модель:

Функція як послуга (FaaS): FaaS, також відомий як безсерверне обчислення, розбиває хмарні програми на ще менші компоненти, які запускаються лише тоді, коли вони потрібні. Уявіть собі, якби можна було орендувати будинок потроху: наприклад, орендар платить лише за їдальню під час обіду, спальню, коли вони сплять, вітальню, коли вони дивляться телевізор, і коли вони не користуються цими кімнатами, їм не потрібно платити за них орендну плату.

FaaS або безсерверні програми все ще працюють на серверах, як і всі ці моделі хмарних обчислень. Але їх називають «безсерверними», тому що вони не працюють на виділених машинах і тому, що компаніям, які створюють програми, не потрібно керувати жодними серверами.

Крім того, безсерверні функції збільшуються або дублюються, оскільки більше людей використовують програму — уявіть, якби їдальня орендаря могла розширюватися на вимогу, коли більше людей приходило б на обід! Дізнайтеся більше про безсерверні обчислення (FaaS).

1.1.2. Типи розгортання хмар

На відміну від розглянутих вище моделей, які визначають, як послуги пропонуються через хмару, ці різні типи розгортання хмари пов’язані з тим, де знаходяться хмарні сервери та хто ними керує.

Найпоширеніші хмарні розгортання:

Приватна хмара: приватна хмара – це сервер, центр обробки даних або розподілена мережа, повністю присвячена одній організації.

Загальнодоступна хмара. Загальнодоступна хмара — це послуга, що надається зовнішнім постачальником і може включати сервери в одному або кількох центрах обробки даних. На відміну від приватної хмари, публічні хмари використовуються кількома організаціями. Використовуючи віртуальні машини, окремі сервери можуть спільно використовуватися різними компаніями. Така ситуація називається «мультитендантом», оскільки кілька орендарів орендують серверний простір на одному сервері.

Гібридна хмара: розгортання гібридної хмари поєднує загальнодоступні та приватні хмари та може навіть включати локальні застарілі сервери. Організація може використовувати свою приватну хмару для одних служб і свою загальнодоступну хмару для інших, або вони можуть використовувати публічну хмару як резервну копію для своєї приватної хмари.

Мультихмарність: мультихмарність – це тип розгортання хмари, який передбачає використання кількох загальнодоступних хмар. Іншими словами, організація з мультихмарним розгортанням орендує віртуальні сервери та послуги від кількох зовнішніх постачальників — якщо продовжити аналогію, використану вище, це схоже на оренду кількох суміжних земельних ділянок у різних орендодавців. Багатохмарні розгортання також можуть бути гібридними хмарами, і навпаки.

1.1.3. Переваги та недоліки хмар

Переваги хмарних обчислень

Хмарне програмне забезпечення надає компаніям х різних бізнес-секторів низку переваг, зокрема можливість використовувати програмне забезпечення з будь-якого пристрою через рідну програму або браузер. Завдяки цьому користувачі можуть безперешкодно синхронізувати свою роботу, ділитися файлами налаштуваннями між пристроями.

Хмарні обчислення – це мережа віддалених серверів, розміщених в Інтернеті для зберігання та отримання даних. Хмара надає низку IT-сервісів, таких як сервери, бази даних, програмне забезпечення, віртуальне сховище та мережа, серед іншого. З точки зору неспеціаліста, хмарні обчислення визначаються як віртуальна платформа, яка дозволяє зберігати та отримувати доступ до ваших даних через Інтернет без будь-яких обмежень.

Компанії, які пропонують усі згадані вище послуги, називаються хмарними провайдерами. Вони надають вам можливість зберігати та отримувати дані та запускати програми, керуючи ними через портали конфігурації.

Хмарні обчислення вирішують усі ці проблеми для бізнесу. Хмарний постачальник бере на себе всі клопоти, пов’язані з інфраструктурою, обслуговуванням і керуванням утилітами для серверів. Хмарні програми зазвичай становлять незначну частину вартості локально встановленого програмного забезпечення. Сплачується лише за час або місце на сервері, які використовуються.

Незважаючи на деякі гучні витоки хмарних даних, є багато аргументів, чому хмарні обчислення є більш безпечними, ніж внутрішні обчислення. Прямо у верхній частині списку є те, що хмарні провайдери перебувають під більшою ретельністю та повинні відповідати встановленим стандартам. Хоча всі компанії юридично зобов’язані захищати інформацію про клієнтів, вони на своїй честі, що стосується методів. Дані, що зберігаються в хмарі, менше піддаються крадіжці співробітників. Інформацію легше вкрасти, якщо у є фізичний доступ до машини, на якій вона зберігається. Хмарні обчислення відокремлюють дані від будь-яких потенційно незадоволених співробітників.

Найбільш очевидним аргументом є те, що постачальники хмарних послуг будуть постійно підтримувати протоколи безпеки та програмне забезпечення в актуальному стані, оскільки від цього залежить їхній

бізнес. Цілком ймовірно, що більшість хмарних провайдерів мають штатних працівників, які спеціалізуються на цифровій/мережевій безпеці.

Хмарні постачальники виживають завдяки резервуванню. Дані не просто зберігаються на сервері. Вони зберігається на кількох серверах. Залежно від постачальника, він навіть може зберігати їх на серверах у кількох місцях. На випадок катастрофічного збою на серверній фермі.

Це означає, що жоден апаратний збій не завадить бізнесу. Це також означає, що можна очікувати надзвичайної надійності з точки зору доступу до даних або послуг. Більшість провайдерів навіть гарантують 99,99% безвідмовної роботи.

Недоліки

Поки немає доступу до Інтернету, не можете нічого робити з хмарою.Надійні плани мобільного передавання даних можуть тимчасово вирішити цю проблему. Послуги стільникового зв’язку часто залишаються життєздатними, коли доступ до Інтернету та навіть відключення електроенергії. Звичайно, плани передачі даних обмежені, а мобільні пристрої мають обмежений час автономної роботи.

Знову ж таки, якщо вимкнеться електрика, ймовірно, виникнуть серйозніші проблеми, з доступом до хмарних служб.

Безпека, на одному рівні, є перевагою хмарних обчислень з причин, описаних вище, але є також й недоліком на іншому рівні.

Користувач є слабкою ланкою практично в усіх системах безпеки. Якщо не використовуються базові методи цифрової безпеки, хмарні обчислення настільки ж безпечні, як залишити свій ноутбук відкритим у кафе.

Безпека — це одна зі сфер, де визначення переваг і недоліків хмарних обчислень залежить від кута, з якого дивитесь на проблему.

Хмарні обчислення – це молода галузь, у якій багато компаній змагаються за бізнес. Існує ймовірність, що у вашого хмарного провайдера закінчаться гроші, і він назавжди закриє свої двері.

Чим важливіше хмара для бізнесу, тим руйнівнішим буде раптове припинення роботи постачальника. Ця проблема посилюється проблемою блокування хмарних постачальників, коли перехід від одного хмарного постачальника до іншого складний і дорогий.

1.2. Машинне навчання

Машинне навчання — це застосування штучного інтелекту, яке дозволяє системам навчатися та вдосконалюватися на основі досвіду без явного програмування. Машинне навчання зосереджується на розробці

комп’ютерних програм, які можуть отримувати доступ до даних і використовувати їх для самостійного навчання.

Подібно до того, як людський мозок отримує знання та розуміння, машинне навчання покладається на вхідні дані, такі як навчальні дані або графіки знань, щоб зрозуміти сутності, домени та зв’язки між ними. З визначенням сутностей можна починати глибоке навчання.

Процес машинного навчання починається зі спостережень або даних, таких як приклади, прямий досвід або інструкції. Він шукає шаблони в даних, щоб пізніше зробити висновки на основі наданих прикладів. Основна мета ML — дозволити комп’ютерам навчатися автономно без втручання чи допомоги людини та відповідно коригувати дії.

Машинне навчання широко використовується підприємствами в усіх секторах для просування інновацій і підвищення ефективності процесів. Вони використовується для:

- Безпека даних: моделі машинного навчання можуть виявляти вразливі місця в безпеці даних, перш ніж вони можуть перетворитися на порушення. Розглядаючи минулий досвід, моделі машинного навчання можуть передбачати майбутню високоризиковану діяльність, щоб ризик можна було завчасно зменшити.
- Фінанси: банки, торговельні брокерські компанії та фінтех-компанії використовують алгоритми машинного навчання для автоматизації торгівлі та надання фінансових консультаційних послуг

інвесторам. Bank of America використовує чат-бота Erica для автоматизації підтримки клієнтів.

- Охорона здоров’я: використовується для аналізу масивних наборів даних охорони здоров’я, щоб пришвидшити пошук методів лікування та лікування, покращити результати лікування пацієнтів і автоматизувати звичайні процеси для запобігання помилкам людини. Наприклад, компанія IBM Watson використовує інтелектуальний аналіз даних, щоб надати лікарям дані, які вони можуть використовувати для персоналізації лікування пацієнтів.
- Виявлення шахрайства: штучний інтелект використовується у фінансовому та банківському секторах для автономного аналізу великої кількості транзакцій з метою виявлення шахрайства в реальному часі. Фірма технологічних послуг Capgemini стверджує, що системи виявлення шахрайства, які використовують машинне навчання та аналітику, мінімізують час розслідування шахрайства на 70% і підвищують точність виявлення на 90%.
- Роздрібна торгівля: Дослідники та розробники штучного інтелекту використовують алгоритми машинного навчання для розробки механізмів рекомендацій штучного інтелекту, які пропонують релевантні пропозиції щодо продуктів на основі минулого вибору покупців, а також історичних, географічних і

  демографічних даних.

1.3.Хмарні платформи машинного навчання?

Машинне навчання є найбільшою частиною технологій у наш час. Природно, що сьогодні всі компанії хочуть використовувати машинне навчання для покращення свого бізнесу. Компанії використовують машинне навчання та аналітику даних, щоб краще зрозуміти свою цільову аудиторію, автоматизувати частину свого виробництва, створювати кращі продукти відповідно до ринкового попиту тощо. Усе це, у свою чергу, підвищує прибутковість компанії, що, у свою чергу, дає їй перевагу над своїми конкурентами. Адже головне в більшості випадків – прибуток!

Однак протягом тривалого часу в минулому компаніям потрібно було інвестувати багато грошей у машинне навчання, щоб отримати цей прибуток. Машинне навчання вимагало багато інфраструктур, програмістів, які були знайомі з машинним навчанням, а аналітика даних була дорогою, і було дуже мало даних для живлення цих алгоритмів машинного навчання!

Хоча для великих транснаціональних корпорацій це було не так вже й важливо, для малих і середніх компаній це було дуже важко. Але популярність і розвиток хмарних сервісів зробили все набагато простіше. Тепер компанії можуть отримати доступ до алгоритмів і технологій машинного навчання від стороннього постачальника, внесли кілька змін відповідно до своїх індивідуальних вимог і почнуть отримувати переваги з набагато меншими початковими інвестиціями.

Ось чому хмарні обчислення такі важливі для машинного навчання! Це рішення для багатьох невеликих і середніх компаній, які не хочуть створювати, тестувати та впроваджувати власні алгоритми машинного навчання з нуля.

Ці компанії можуть зосередитися на своєму основному бізнесі та отримати додаткову вартість від машинного навчання, не потребуючи стати експертами. Таким чином, вони отримують збільшені прибутки, одночасно зменшуючи ризик інвестицій, що означає, що це безпрограшна ситуація для всіх!

1. Веб-сервіси Amazon

Amazon Web Services — це платформа хмарних обчислень, яка є дочірньою компанією Amazon. Вона була запущена в 2006 році і наразі є однією з найпопулярніших платформ хмарних обчислень для машинного навчання. AWS пропонує різні продукти для машинного навчання, наприклад:

- Amazon SageMaker – використовується для створення та навчання моделей машинного навчання
- Amazon Augmented AI – використовується для перевірки моделями машинного навчання людиною
- Прогноз Amazon – це використовує машинне навчання для підвищення точності прогнозу

- Amazon Translate – використовує машинне навчання та обробку природної мови для мовного перекладу
- Amazon Personalize – створює персональні рекомендації в системах машинного навчання
- AWS Deep Learning AMI – використовується для рішень глибокого навчання
- Amazon Polly – використовується для перетворення тексту в реальне мовлення

2. Microsoft Azure

Microsoft Azure — це платформа хмарних обчислень, створена Microsoft. Спочатку він був випущений у 2010 році та є популярною платформою хмарних обчислень для машинного навчання та аналізу даних. Деякі з продуктів Microsoft Azure для машинного навчання:

- Когнітивна служба Microsoft Azure – надає розумні когнітивні служби для програм.
- Microsoft Azure Azure Databricks – забезпечує аналітику на основі Apache Spark
- Microsoft Azure Bot Service – це надає розумні та інтелектуальні бот-сервіси, які можна масштабувати
- Microsoft Azure Cognitive Search – це служба на основі машинного навчання для мобільних і веб-додатків
- Машинне навчання Microsoft Azure – використовується для створення та розгортання моделей машинного навчання в хмарі

3. Google Cloud

Google Cloud Platform – це платформа хмарних обчислень, яку надає Google. Він був запущений у 2008 році та забезпечує таку саму інфраструктуру для компаній, яку Google також використовує у своїх внутрішніх продуктах. Google Cloud пропонує різні продукти для машинного навчання, як-от:

- Google Cloud AutoML – використовується для навчання моделі машинного навчання AutoML та її розробки
- Google Cloud AI Platform – використовується для створення, навчання та керування моделями машинного навчання
- Google Cloud Speech-to-Text – це система розпізнавання мовлення для передачі мовлення в текст, яка підтримує 120 мов.
- Google Cloud Vision AI – використовується для створення моделей машинного навчання для хмарного бачення, які виявляють текст тощо.
- Google Cloud Text-to-Speech – це система створення мовлення для передачі тексту в мовлення
- Природна мова Google Cloud – призначена для обробки природної мови для аналізу та класифікації тексту

4. IBM Cloud

IBM Cloud Platform — це платформа хмарних обчислень, яку пропонує IBM. Він надає різні моделі хмарної доставки: публічні, приватні та гібридні. IBM Cloud надає різні продукти для машинного навчання, наприклад:

- IBM Watson Studio – використовується для створення моделей машинного навчання та штучного інтелекту, а також для підготовки та аналізу даних
- IBM Watson Speech-to-Text – це система розпізнавання мовлення для перетворення мовлення та звуку в письмовий текст
- IBM Watson Text-to-Speech – це система створення мовлення для перетворення тексту в аудіо з природним звучанням
- IBM Watson Natural Language Understanding – це для обробки природної мови для аналізу та класифікації тексту
- IBM Watson Visual Recognition – це використовує машинне навчання для пошуку візуальних зображень і їх класифікації
- IBM Watson Assistant – використовується для створення віртуальних помічників і керування ними

1.4.Прогнозування

«Прогнозування» означає результат алгоритму після його навчання на історичному наборі даних і застосування до нових даних під час прогнозування ймовірності певного результату, наприклад, чи повернеться клієнт через 30 днів. Алгоритм генеруватиме ймовірні значення для невідомої змінної для кожного запису в нових даних, дозволяючи конструктору моделі визначити, яким це значення буде найімовірніше.

Слово «прогноз» може ввести в оману. У деяких випадках це дійсно означає, що ви передбачаєте майбутній результат, наприклад, коли ви використовуєте машинне навчання для визначення наступної найкращої дії в маркетинговій кампанії. Однак в інших випадках «прогноз» стосується, наприклад, того, чи була транзакція, яка вже відбулася, шахрайською. У цьому випадку транзакція вже відбулася, але ви робите обґрунтоване припущення про те, чи була вона законною, що дозволяє вжити відповідних дій.

Чому прогнози важливі?

Прогнози моделі машинного навчання дозволяють компаніям робити дуже точні припущення щодо ймовірних результатів запитання на основі історичних даних, які можуть стосуватися будь-яких речей – ймовірності відтоку клієнтів, можливих шахрайських дій тощо. Вони надають компанії інформацію, що призводить до відчутної цінності для

бізнесу. Наприклад, якщо модель передбачає, що клієнт, швидше за все, залишиться, компанія може націлити на нього спеціальні комунікації та охоплення, які запобіжать втраті цього клієнта.

Розділ 2. Методи та підходи розробки інформаціних технологій

2.1 AWS Sagemaker

Amazon SageMaker є широко використовуваним сервісом і визначається як керований сервіс у хмарі Amazon Web Services (AWS), який надає інструменти для створення, навчання та розгортання моделей машинного навчання (ML) для програм прогнозної аналітики. Платформа Amazon SageMaker автоматизує незмінну роботу зі створення готових до виробництва конвеєрів штучного інтелекту (AI). Amazon SageMaker також дозволяє розробникам розгортати моделі машинного навчання на вбудованих системах і периферійних пристроях. Amazon SageMaker створює повністю керований екземпляр машинного навчання в Amazon Elastic Compute Cloud (EC2).

Він підтримує веб-програму Jupyter Notebook з відкритим кодом, яка дозволяє розробникам ділитися живим кодом і співпрацювати. Amazon SageMaker запускає блокноти для обчислювальної обробки Jupyter. Ноутбуки включають драйвери, пакети та бібліотеки для подібних платформ глибокого навчання та фреймворків. Розробники можуть запустити готовий ноутбук, який AWS постачає для різноманітних програм і варіантів використання, а потім може налаштувати їх відповідно до набору даних і схеми, які потребують подальшого навчання. Розробники також можуть використовувати спеціально створені алгоритми, написані в одному з підтримуваних фреймворків машинного навчання, або якийсь код, упакований як образ

контейнера Docker. Amazon SageMaker допомагає отримувати дані з Amazon Simple Storage Service (S3), і не існує певного практичного обмеження розміру набору даних.

Проектування та впровадження систем

У Amazon Elastic Compute Cloud(EC2) Amazon SageMaker запускає повністю керований екземпляр машинного навчання. Оскільки Jupyter Notebook з ним сумісний, розробники можуть обмінюватися з ним живим кодом. Обчислювальні дії виконуються SageMaker за допомогою блокнотів Jupyter.

Для популярних технологій і архітектур глибокого навчання ноутбуки містять пакети, драйвери та бібліотеки. AWS надає попередньо налаштовані блокноти для ряду програм і варіантів використання, які розробники можуть запускати. Потім вони можуть змінити його відповідно до схеми збору даних і навчання.

Крім того, програмісти можуть використовувати будь-який код, упакований як образ контейнера Docker, або спеціально створені алгоритми, реалізовані в одному з дозволених фреймворків машинного навчання. Немає реальних обмежень щодо обсягу збору даних, до якого SageMaker може отримати доступ через Amazon Simple Storage Service (S3).

Розробник запускає екземпляр блокнота після підключення до консолі SageMaker. SageMaker має низку вбудованих моделей навчання, таких як класифікація зображень і лінійна регресія, або розробник може імпортувати спеціальні методи.

Розташування даних у сегменті Amazon S3 і відповідний тип ілюстрації визначаються розробниками навчання моделі. Після цього вони починають тренувальний процес.

Для постійного автоматичного налаштування моделі для визначення ідеального набору параметрів або гіперпараметрів доступний монітор прототипу AWS SageMaker. На цьому етапі дані змінюються, щоб увімкнути їх збільшення.

Сервіс автоматично підтримує та розширює хмарну інфраструктуру, коли модель підготовлена до розгортання. Він використовує різні типи екземплярів AWS SageMaker, які включають багато прискорювачів GPU, створених спеціально для додатків машинного навчання.

SageMaker генерує захищені кінцеві точки HTTPS для підключення до програм, розгортає в кількох зонах доступності, виконує перевірку працездатності, встановлює оновлення безпеки та налаштовує автоматичне масштабування AWS.

Щоб відстежувати зміни продуктивності та сповіщати про них, розробник може використовувати показники Amazon CloudWatch.

Переваги Amazon SageMaker

Amazon SageMaker дає змогу більшій кількості людей впроваджувати інновації в машинне навчання завдяки вибору інструментів — інтегрованих середовищ розробки для науковців із обробки даних, інженерів з машинного навчання та візуальних інтерфейсів без коду для бізнес-аналітиків, що робить машинне навчання доступнішим. Amazon SageMaker допомагає отримати доступ, позначити та обробити великі обсяги структурованих даних (табличні дані) і неструктурованих даних (фото, відео та аудіо) для машинного навчання, таким чином допомагаючи підготувати дані в масштабі. Amazon SageMaker допомагає скоротити час навчання з годин до хвилин завдяки оптимізованій інфраструктурі, тим самим підвищуючи продуктивність команди до 10 разів за допомогою спеціальних інструментів, що прискорює розвиток машинного навчання. Amazon SageMaker допомагає автоматизувати та стандартизувати практики MLOps у всій організації для створення, навчання.

Приклади використання Amazon SageMaker

- Він забезпечує блокноти Jupyter в один клік

Ноутбуки Amazon SageMaker Studio допомагають швидше створювати моделі ML і співпрацювати з командою. Ноутбуки Amazon SageMaker Studio створюють блокноти Jupyter одним клацанням миші, з якими користувач може почати працювати за лічені секунди.

Крім того, основні обчислювальні ресурси є повністю еластичними, тому користувач може легко підключати або зменшувати доступні ресурси, а зміни відбуваються автоматично у фоновому режимі, не перериваючи вашу роботу. Amazon SageMaker також дозволяє ділитися записниками в один клік. Усі залежності коду фіксуються автоматично, тому ви можете легко співпрацювати з іншими, і вони отримають той самий блокнот, збережений у тому самому місці.

- Він забезпечує інтерфейс RStudio.

Amazon SageMaker передає наявні ліцензії RStudio та легко й безпечно переносить середовища RStudio на Amazon SageMaker. RStudio на Amazon SageMaker надає користувачам знайоме середовище розробки RStudio з ресурсами хмарних обчислень на вимогу. Користувачі можуть запускати RStudio одним клацанням миші з Amazon SageMaker, оскільки він повністю керований, а розробники R можуть виконувати обчислення за допомогою комутованого доступу з одного інтерфейсу, зменшуючи переривання роботи та підвищуючи продуктивність.

- Він забезпечує AutoML.

Автопілот Amazon SageMaker автоматично створює, навчає та налаштовує найкращі моделі машинного навчання на основі даних, дозволяючи користувачам зберігати повний контроль і видимість. Потім користувачі можуть безпосередньо розгорнути модель у виробництві лише одним клацанням миші або виконати ітерацію, щоб покращити якість моделі.

- Він надає готові рішення для моделей з відкритим кодом

Amazon SageMaker JumpStart допомагає користувачам швидко розпочати роботу з машинним навчанням за допомогою готових рішень, які можна розгорнути лише кількома клацаннями миші. SageMaker JumpStart також підтримує розгортання одним клацанням миші та точне налаштування понад 150 популярних моделей із відкритим кодом.

- Він оптимізований для основних фреймворків

Amazon SageMaker оптимізовано для різноманітних популярних фреймворків глибокого навчання, таких як TensorFlow, Apache MXNet, PyTorch тощо. Фреймворки завжди оновлюються до останньої версії та оптимізовані для продуктивності на AWS. Користувачам не потрібно вручну налаштовувати ці фреймворки, і вони можуть використовувати їх у вбудованих контейнерах.

- Він забезпечує локальний режим.

Amazon SageMaker дозволяє користувачам тестувати та створювати прототипи локально. Контейнери Apache MXNet і TensorFlow Docker, які використовуються в AWS SageMaker, доступні на GitHub. Користувачі можуть завантажувати ці контейнери та використовувати Python SDK для тестових сценаріїв перед розгортанням для навчання або хостингу.

Для більшості дослідників даних, які хочуть створити справді наскрізне рішення МЛ, AWS Sagemaker має надзвичайну цінність. Він абстрагує велику кількість можливостей розробки програмного забезпечення, необхідних для завершення роботи, будучи надзвичайно ефективним, універсальним і економічно ефективним.

Найважливіше те, що це дозволяє вам зосередитися на основних експериментах ML, доповнюючи решту необхідних можливостей простими абстрактними інструментами, які можна порівняти з нашим поточним підходом.

2.2 Principal Component Analysis

Основна ідея аналізу головних компонентів (PCA) полягає в тому, щоб зменшити розмірність набору даних, що складається з багатьох змінних, сильно або слабко корельованих одна з одною, зберігаючи при цьому варіацію, наявну в наборі даних, до максимального ступеня. Те ж

саме робиться шляхом перетворення змінних на новий набір змінних, які відомі як головні компоненти (або просто ПК) і є ортогональними, упорядкованими таким чином, що збереження варіацій, присутніх у вихідних змінних, зменшується, коли ми рухаємося вниз. в порядку. Таким чином, 1-й основний компонент зберігає максимальну варіацію, яка була присутня у вихідних компонентах. Головні компоненти є власними векторами коваріаційної матриці, а отже, вони ортогональні.

Важливо, що набір даних, на якому буде використовуватися техніка PCA, має бути масштабованим. Результати також чутливі до відносного масштабування. Це метод узагальнення даних, PCA використовують коли в є надмірно багато різних ознак, які треба узагальнити і отримати меншу кількість характеристик.

Інтуїтивно зрозуміло, що аналіз головних компонентів може надати користувачеві зображення меншого розміру, проекцію або «тінь» цього об’єкта, якщо дивитися з його найбільш інформативної точки зору.

Технічно головний компонент можна визначити як лінійну комбінацію оптимально зважених спостережуваних змінних. Результатом PCA є ці основні компоненти, кількість яких менша або дорівнює кількості початкових змінних.

Принципи роботи PCA

- Крок 1: нормалізація даних

Першим кроком є нормалізація даних, які ми маємо, щоб PCA працював належним чином. Це робиться шляхом віднімання відповідних середніх значень із чисел у відповідному стовпчику. Отже, якщо у нас є два виміри X і Y, усі X стають $\boldsymbol{x}$-, а всі Y стають $\boldsymbol{y}$-. Це створює набір даних, середнє значення якого дорівнює нулю.

- Крок 2: Обчислення коваріаційної матриці

Оскільки набір даних, який ми взяли, є двовимірним, це призведе до матриці коваріації:

$$Matrix(Covariance) = \begin{bmatrix} Var[X_1] & Cov[X_1, X_2] \\ Cov[X_2, X_1] & Var[X_2] \end{bmatrix}$$

Варто зазначити, що Var[X 1] = Cov[X 1 ,X 1] і Var[X 2] = Cov[X 2 ,X 2].

- Крок 3: Обчислення значень та векторів

Наступним кроком є обчислення власних значень і власних векторів для коваріаційної матриці. Те саме можливо, оскільки це квадратна матриця. ƛ є власним значенням для матриці A, вона є розв’язком характеристичного рівняння:

det(ƛI - A) = 0

Де I — одинична матриця тієї ж розмірності, що й A, що також є необхідною умовою для віднімання матриці в цьому випадку, а «det» — визначник матриці. Для кожного власного значення λ можна знайти відповідний власний вектор v шляхом вирішення:

$$(\lambda I - A)v = 0$$

- Крок 4: Вибір компонентів і формування вектора ознак:

Впорядковуємо власні значення від найбільшого до найменшого, щоб це дало нам компоненти в порядку чи значенні. Тут іде частина зменшення розмірності. Якщо у нас є набір даних із n змінними, то ми маємо відповідні n власних значень і власних векторів.

Виявляється, що власний вектор, який відповідає найвищому власному значенню, є головним компонентом набору даних, і ми вирішуємо, скільки власних значень ми виберемо для продовження аналізу. Щоб зменшити розміри, ми вибираємо перші p власних значень і ігноруємо решту. У процесі ми втрачаємо деяку інформацію, але якщо власні значення малі, ми не втрачаємо багато.

Далі формуємо вектор ознак, який є матрицею векторів, у нашому випадку власних векторів. Фактично, лише ті власні вектори, з якими ми хочемо продовжити. Оскільки у поточному прикладі ми маємо лише 2

виміри, ми можемо вибрати той, який відповідає більшому власному значенню, або просто взяти обидва.

Вектор ознак = (eig 1, eig 2)

- Крок 5: Формування основних компонентів:

Це останній крок, на якому ми фактично формуємо основні компоненти, використовуючи всю математику, яку ми робили до цього моменту. Для того ж ми беремо транспонування вектора ознак і множимо його зліва на транспонування масштабованої версії вихідного набору даних.

NewData = FeatureVectorT * ScaledDataT

NewData — це матриця, що складається з основних компонентів,

FeatureVector — це матриця, яку ми сформували за допомогою власних векторів, які ми вирішили зберегти, і

ScaledData — це масштабована версія оригінального набору даних

('T' у верхньому індексі позначає транспонування матриці, яка формується шляхом заміни рядків на стовпці і навпаки. Зокрема, матриця 2x3 має транспонування розміром 3x2)

Використання

PCA переважно використовується як техніка зменшення розмірності в таких областях, як розпізнавання обличчя, комп’ютерне бачення та стиснення зображень. Він також використовується для пошуку закономірностей у даних великої розмірності в галузі фінансів, аналізу даних, біоінформатики, психології тощо.

2.3 Xgboost

XGBoost — це бібліотека програмного забезпечення, яку можна завантажити та інсталювати на свій комп’ютер, а потім отримати доступ із різноманітних інтерфейсів.

Бібліотека зосереджена на швидкості обчислення та продуктивності моделі, тому в ній небагато надмірностей. Тим не менш, він пропонує ряд розширених функцій.

Реалізація моделі підтримує функції реалізацій з новими доповненнями, такими як регулярізація. Підтримуються три основні форми посилення градієнта:

- Алгоритм посилення градієнта також називається машиною підвищення градієнта, включаючи швидкість навчання.
- Підвищення стохастичного градієнта з підвибіркою в рядках, стовпцях і стовпцях на рівні поділу.

- Регуляризоване посилення градієнта з регуляризацією L1 і L2.

Бібліотека надає систему для використання в різних обчислювальних середовищах, зокрема:

- Розпаралелювання побудови дерева з використанням усіх ядер ЦП під час навчання.
- Розподілені обчислення для навчання дуже великих моделей за допомогою кластера машин.
- Позаядерні обчислення для дуже великих наборів даних, які не вміщуються в пам’ять.
- Оптимізація кеш-пам’яті структур даних і алгоритму для найкращого використання обладнання.

Особливості алгоритму

Реалізація алгоритму розроблена для ефективного використання часу обчислення та ресурсів пам’яті. Мета дизайну полягала в тому, щоб якнайкраще використати доступні ресурси для навчання моделі. Деякі ключові особливості реалізації алгоритму включають:

- Реалізація Sparse Aware з автоматичною обробкою відсутніх значень даних.
- Блокова структура для підтримки розпаралелювання побудови дерева.

- Продовжуйте навчання , щоб ви могли додатково посилити вже підігнану модель на нових даних.

Бібліотека XGBoost реалізує алгоритм дерева рішень із посиленням градієнта.

XGBoost працює, поєднуючи низку слабких учнів, щоб сформувати сильного учня. Слабкий учень — це модель машинного навчання, яка лише трохи краща за випадкове вгадування. Однак, коли слабкі учні поєднуються, вони можуть сформувати сильний учень, який є набагато точнішим.

XGBoost працює шляхом навчання кількох дерев рішень. Кожне дерево навчається на підмножині даних, а прогнози з кожного дерева об’єднуються для формування остаточного прогнозу.

XGBoost є вдосконаленням алгоритму GBM. Основна відмінність полягає в тому, що XGBoost використовує більш упорядковану модель, яка допомагає запобігти переобладнанню.

XGBoost має низку параметрів, які можна налаштувати для покращення продуктивності алгоритму.

Найбільш важливими параметрами є:

max_depth: максимальна глибина дерев рішень.

eta: швидкість навчання.

gamma: мінімальне зменшення втрат, необхідне для поділу.

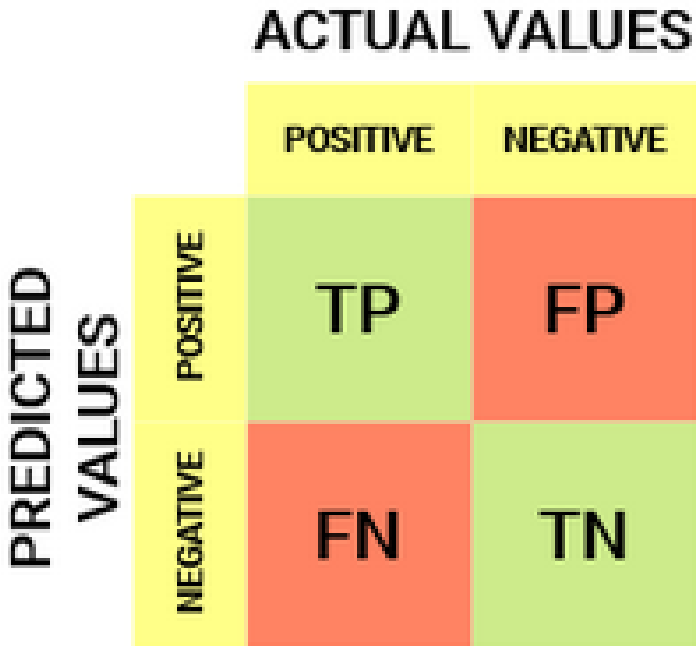
підвибірка: Частка навчальних даних, яка використовується для навчання кожного дерева.

Цей алгоритм має багато різних назв, наприклад посилення градієнта, множинні дерева адитивної регресії, стохастичне підвищення градієнта або машини підвищення градієнта..

2.4 Матриця плутанини

Матриця плутанини — це матриця N x N, яка використовується для оцінки продуктивності моделі класифікації, де N — кількість цільових класів. Матриця порівнює фактичні цільові значення з прогнозованими моделлю машинного навчання. Це дає нам цілісне уявлення про те, наскільки добре працює наша модель класифікації та які помилки вона допускає.

Для задачі двійкової класифікації ми матимемо матрицю 2 x 2, як показано нижче на мал 1, із 4 значеннями:



Мал.1 Загальний вигляд матриці плутанини

Обчислення з використанням матриці плутанини:

Розшифруємо матрицю:

- Цільова змінна має два значення: позитивне або негативне
- Стовпці представляють фактичні значення цільової змінної
- Рядки представляють прогнозовані значення цільової змінної

Справжній позитивний (TP) :

- Прогнозоване збігається з фактичним
- Фактичне значення було позитивним, а модель передбачала позитивне значення

Істинно негативний (TN) :

- Прогнозоване збігається з фактичним
- Фактичне значення було від'ємним, а модель передбачила від'ємне значення

Хибнопозитивний результат (FP) – помилка типу 1:

- Передбачене значення було помилково передбаченим
- Фактичне значення було негативним, але модель передбачила позитивне значення

Помилково негативний (FN) – помилка типу 2:

- Передбачене значення було помилково передбаченим
- Фактичне значення було позитивним, але модель передбачила негативне значення

За допомогою цієї матриці ми можемо виконувати різні обчислення для моделі:

- Точність класифікації: це один із важливих параметрів для визначення точності проблем класифікації. Він визначає, як часто модель передбачає правильний результат. Його можна розрахувати як відношення кількості правильних прогнозів, зроблених класифікатором,

до всієї кількості прогнозів, зроблених класифікаторами. Формула наведена нижче:

$$Accuracy = \frac{TP + TN}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + TN}$$

- Рівень неправильної класифікації: його також називають коефіцієнтом помилок і він визначає, як часто модель дає неправильні прогнози. Значення коефіцієнта помилок можна розрахувати як кількість невірних прогнозів до всієї кількості прогнозів, зроблених класифікатором. Формула наведена нижче:

$$Error\ rate = \frac{FP + FN}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + TN}$$

- Точність: це можна визначити як кількість правильних вихідних даних, наданих моделлю, або кількість із усіх позитивних класів, які правильно передбачила модель, скільки з них були дійсно вірними. Його можна розрахувати за наведеною нижче формулою:

$$Precision = \frac{TP}{\mathrm{TP} + \mathrm{FP}}$$

- Відкликання: це визначається як із загальної кількості позитивних класів, як наша модель правильно передбачила. Відкликання має бути якомога вищим.

$$Recall = \frac{TP}{\mathrm{TP} + \mathrm{FN}}$$

- F-міра: якщо дві моделі мають низьку точність і високу запам’ятовуваність або навпаки, важко порівняти ці моделі. Отже, для цієї мети ми можемо використовувати F-оцінку.

Ця оцінка допомагає нам одночасно оцінити пам’ять і точність. F-показник є максимальним, якщо пригадування дорівнює точності. Його можна розрахувати за наведеною нижче формулою:

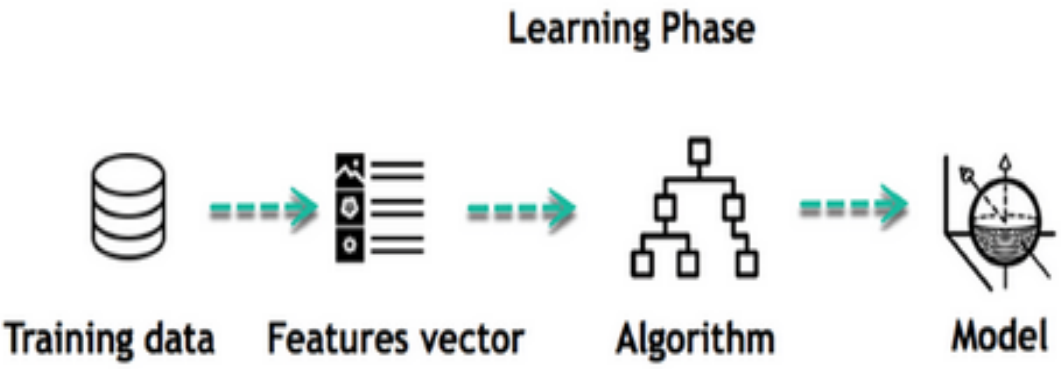
$$F\_measure = \frac{2Recall * Precision}{Recall + Precision}$$

Розділ 3 Реалізація хмарної інформаційної технології

3.1 Контрольоване машинне навчання

Контрольоване машинне навчання – це алгоритм, який навчається на основі позначених навчальних даних, щоб допомогти вам передбачити результати для непередбачених даних. Під час навчання під наглядом ви навчаєте машину, використовуючи дані, які добре «марковані». Це означає, що деякі дані вже позначені правильними відповідями. Це можна порівняти з навчанням у присутності керівника чи вчителя.

Контрольоване машинне навчання використовує навчальні набори даних для досягнення бажаних результатів. Ці набори даних містять вхідні та правильні вихідні дані, які допомагають моделі навчатися швидше. Принцип роботи зображено на мал.2



Мал.2 Принцип контрольованого навчання

3.2 Дерево рішень

Алгоритми дерева рішень — це тип структурної моделі, подібної до дерева ймовірностей, яка постійно розділяє дані, щоб класифікувати або робити прогнози залежно від результатів попереднього набору запитань. Модель аналізує дані та надає відповіді на запитання, щоб допомогти вам зробити більш обґрунтований вибір.

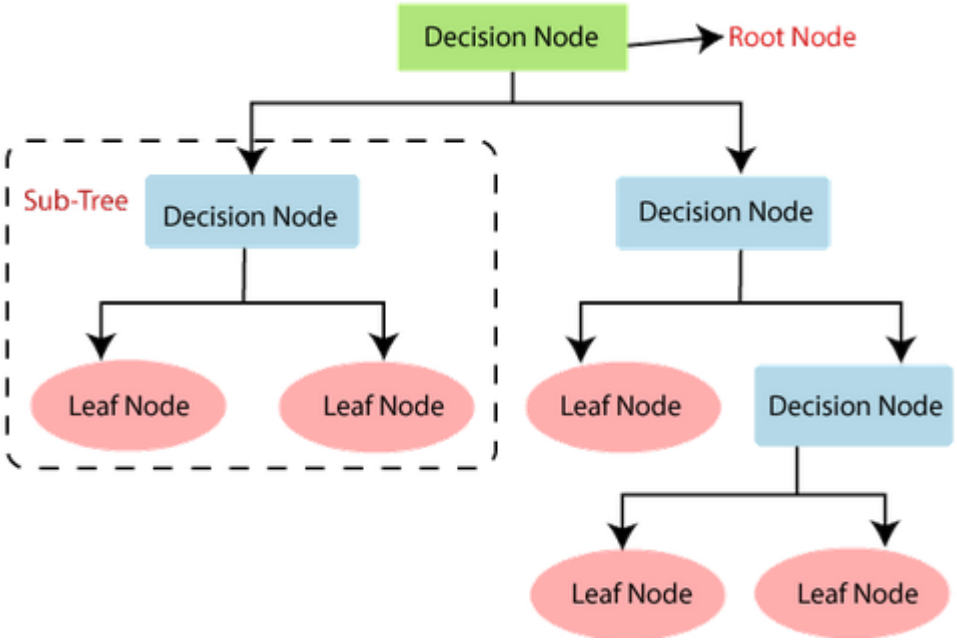
Дерево має два вузли, які є вузлами рішення та листовим вузлом. Вузли рішень відповідаючи своїй назві використовуються для прийняття рішень можуть розгалуджуватися навідміну від листових вузлів, які є результатами рішень і не мають дочірніх елементів.

Рішення чи перевірка виконуються на основі певних особливостей даного кожного набору даних.

Це графічне представлення для отримання всіх можливих рішень проблеми за заданими умовами і називається деревом рішень, оскільки, подібно до дерева, воно бере початок з кореневого вузла, який розгалуджується на подальші вузли утворює подобу гілок та собою нагадує деревоподібну структуру.

Щоб побудувати дерево, використовуємо алгоритм CART, який розшифровується як алгоритм дерева класифікації та регресії.

Дерево рішень просто задає запитання, а на основі відповіді (Так/Ні) розбивається дерево на піддерева. Схема на малюнку 3 пояснює загальну структуру дерева рішень:



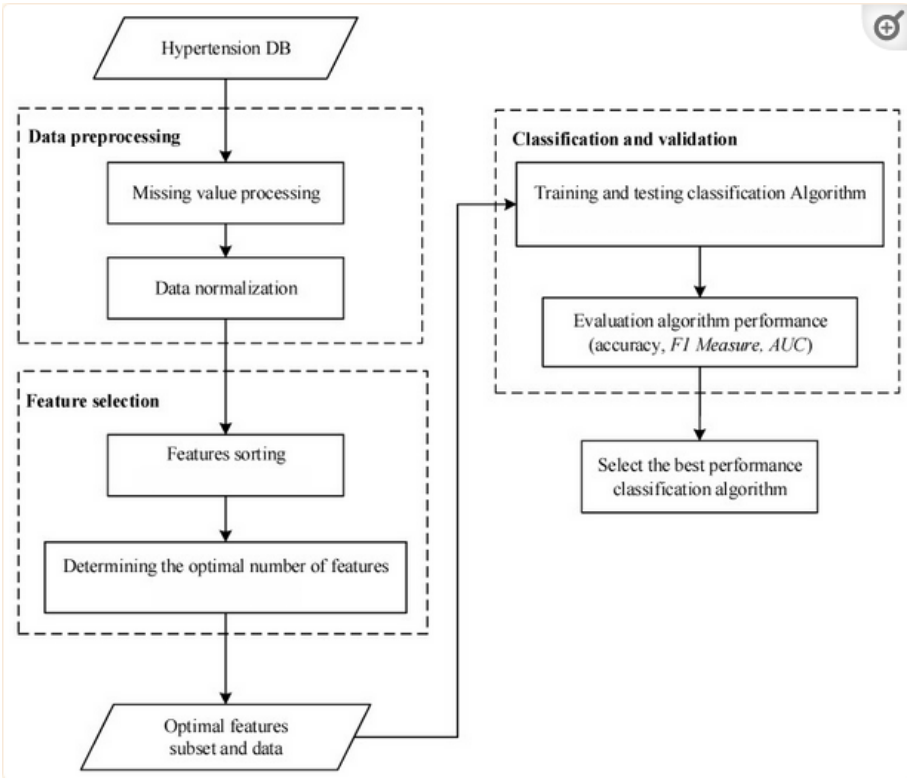
Мал.3 Структура дерева рішень

В поданій робоні використовується один з різновидів дерева рішень-XGBoost, він може надати покращені механізми, ніж інші алгоритми моделі ML. По суті, з моменту свого створення він перетворився на «найкращий у своєму класі» алгоритм моделі ML для керування організованою інформацією.

3.1 Робота в AWS Sagemaker

Для створення інформаційної системи обираемо необхідні параметри за якими відбудеться прогнозування наявності проблем серця.До цих ознак відносими: вік, стать, вагу, зріст, паління, вживання алкоголю, рівень глюкози та холестерину, фізичну активність, систолічний та діастолічний тиски, наявність в минулому проблем серця.

Отримані дані слід в csv-форматі слід обробити, нормалізувати, вибрати основні характеристики необхідні для створення моделі передбачення зображено на мал.4



Мал.4 Схема тренування моделі прогнозування

Для початку в середовищі Sagemaker слід додати необхідні бібліотеки показано на малюнку 5

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Мал.5 Додавання компонентів в Sagemaker

Pandas - це один із інструментів машинного навчання, який використовується для очищення й аналізу даних. Він має функції, які використовуються для дослідження, очищення, перетворення та візуалізації даних.

NumPy - базовий інструмент для вивчення машинного навчання. Багато його функцій дуже корисні для виконання будь-яких математичних чи наукових розрахунків. Оскільки відомо, що математика є основою машинного навчання, більшість математичних завдань можна виконувати за допомогою.

Seaborn - це бібліотека візуалізації даних Python на основі matplotlib. Він забезпечує інтерфейс високого рівня для малювання привабливої та інформативної статистичної графіки.

Matplotlib - це бібліотека для побудови графіків для мови програмування Python та її розширення чисельної математики NumPy. Він надає об’єктно-орієнтований API для вбудовування графіків у програми за допомогою наборів інструментів загального призначення GUI, таких як Tkinter, wxPython, Qt або GTK.

Після цього додаємо csv-файл з даними в AWS Sagemaker і зчитуємо дані малюнок 6.

```
cardio_df = pd.read_csv("cardio_train.csv", sep=";")
```

Мал.6 Отримання даних з csv-файлу

Для подальшого використання зчитаних даних визначаємо їх тип, та підготовлюємо для їх майбутньоЇ обробки в для тестування та прогнозування показано на малюнку 7.

#	Column	Non-Null Count	Dtype
0	age	70000 non-null	float64
1	gender	70000 non-null	int64
2	height	70000 non-null	int64
3	weight	70000 non-null	float64
4	ap_hi	70000 non-null	int64
5	ap_lo	70000 non-null	int64
6	cholesterol	70000 non-null	int64
7	gluc	70000 non-null	int64
8	smoke	70000 non-null	int64
9	alco	70000 non-null	int64
10	active	70000 non-null	int64
11	cardio	70000 non-null	int64

Мал.7 Дані отримані після зчитування

Після цього використовуючи seaborn перетворюємо дані в матрицю кореляцій, яка дасть нам змогу відстежити певну залежність між основними параметрами візуально мал код відображення матриці, зображено на малюнку 8.

Мал.8 Матриця кореляцій

Продовжуючи роботу з даними поділяємо дані на дві групи:

Перша відома як навчальні дані — це частина нашого фактичного набору даних, яка вводиться в модель машинного навчання для виявлення та вивчення шаблонів. Таким чином він навчає нашу модель.

Дані навчання зазвичай більші, ніж дані тестування. Це тому, що ми хочемо надати моделі якомога більше даних, щоб знайти та вивчити значущі шаблони. Коли дані з наших наборів даних передаються в алгоритм машинного навчання, він вивчає шаблони з даних і приймає рішення.

Алгоритми дозволяють машинам вирішувати проблеми на основі минулих спостережень. Схоже, як навчатися на прикладі, як і люди. Єдина відмінність полягає в тому, що машини вимагають набагато більше прикладів, щоб мати можливість бачити закономірності та вчитися.

Оскільки моделі машинного навчання піддаються більш релевантним навчальним даним, тим більше вони покращуються з часом.

Друга - дані для перевірки вашої моделі. Ці дані називаються даними тестування, і ви можете використовувати їх, щоб оцінити продуктивність і прогрес навчання ваших алгоритмів, а також налаштувати або оптимізувати їх для покращення результатів.

Дані тестування мають два основні критерії. Він повинен:

- Представляє фактичний набір даних
- Бути достатньо великим, щоб генерувати значущі прогнози
-

Тестові дані забезпечують остаточну реальну перевірку невидимого набору даних, щоб підтвердити, що алгоритм машинного навчання було навчено ефективно.

У науці про дані зазвичай ваші дані поділено на 80% для навчання, та 20% для тестів зображено на малюнку 9.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df_final, df_target, test_size = 0.2)
```

Мал.9 Поділ даних для навання та тестів

3.2 Використання PCA та Xgboost

В Amazon SageMaker PCA працює в двох режимах залежно від сценарію:

- звичайний: для наборів даних з розрідженими даними та помірною кількістю спостережень і ознак.
- рандомізовані: для наборів даних із великою кількістю спостережень і ознак. У цьому режимі використовується наближений алгоритм.

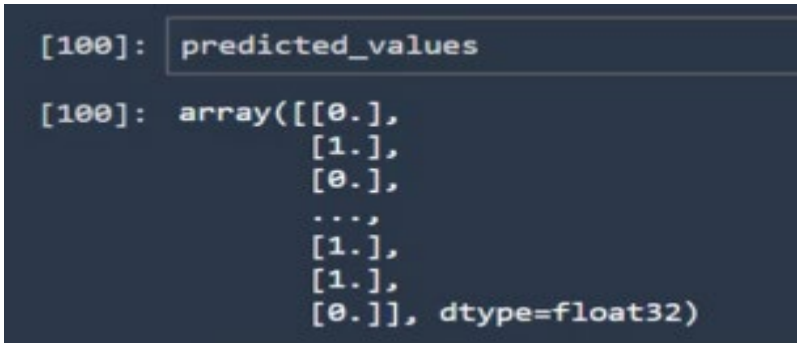
Через невелику кількість параметрів цій роботі використовується PCA в звичайному режимі.

PCA використовує табличні дані.

Рядки представляють спостереження, які треба вбудувати в нижчий вимірний простір. Стовпці представляють функції, для яких треба знайти зменшене наближення. Алгоритм обчислює коваріаційну матрицю (або її наближення розподіленим способом), а потім виконує сингулярне розкладання цього підсумку для отримання головних компонентів.

Впроцесі виконання алгоритму обираємо шість основних компонентів над якими працюватиме алгоритм, це забезпечить більшу точність майбутнього прогнозування та пришвидшить навчання моделі

передбачень. Результат виконання PCA отримуємо в вигляді масиву передбачень показано на малюнку 10.

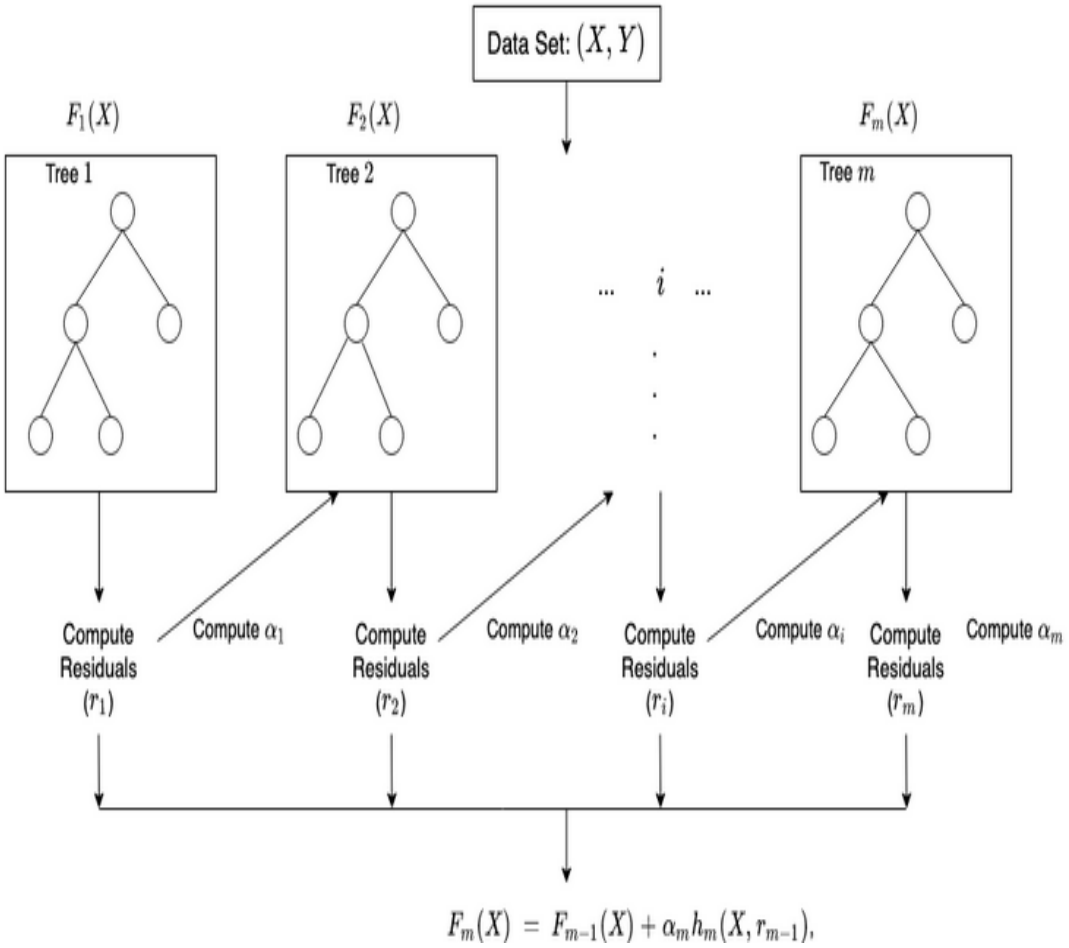


Мал.10 Масив передбачень

Далі використовуємо XGBoost для прогнозування використовуючи дані оброблені методом головних компонентів.

За допомогою SageMaker можна використовувати XGBoost як вбудований алгоритм або як структуру. Використовуючи XGBoost як структуру є більше гнучкості та доступ до більш розширених сценаріїв, таких як k-кратна перехресна перевірка, оскільки можна налаштувати власні сценарії навчання.

В процесі роботи з алгоритмом знову ділими дані на навчальні та тренувальні, але вже з урахування даних оброблених PCA, це дає змогу отримати передбачення та перевірити їх точність. Принцип роботи XGBoost в sagemaker показано на малюнку 11.



Мал.11 Принцип роботи XGBoost

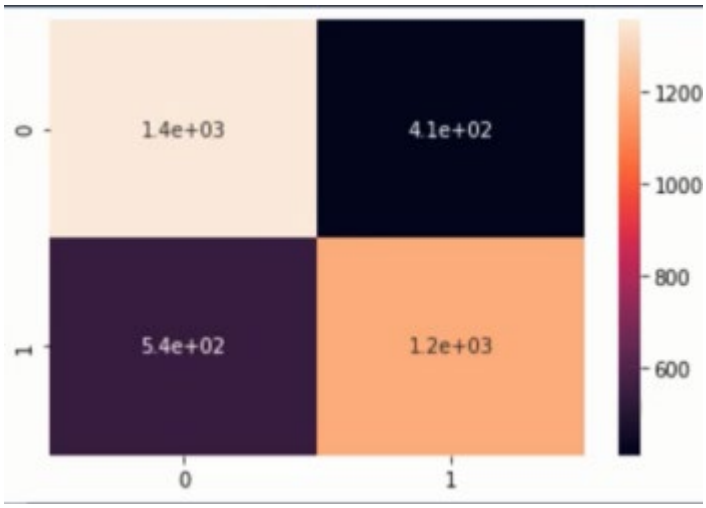
З подальшого використання алгоритму отримуємо точність моделі 72 відсотки,що дає змогу досить точно визначати наявніть проблем з серцем в людей мал 12

```
print("Precision = {}".format(precision_score(y_test, predicted_values, average='macro')))
print("Recall = {}".format(recall_score(y_test, predicted_values, average='macro')))
print("Accuracy = {}".format(accuracy_score(y_test, predicted_values)))
```

```
Precision = 0.7278338413425784
Recall = 0.7262011126481104
Accuracy = 0.7265714285714285
```

Мал.12 отримана точність моделі

Також отримуємо матрицю плутанини для моделі, яка показує істиність передбачувань в графічній формі мал



Мал.12 Отримана матриця плутанини

Висновки:

У цій роботі розглянуто основні теоретичні та практичні питання створення хмарних інформаційних систем. Було досліджено методи машинного навчання в хмарі.Також були проаналізовані сучасні технології та платформи хмарної розробки, які використовуються для машинного навчання, хмарних обчислень. Було досліджено алгоритми машинного навчання,та їх практичне використання.

Поведено аналіз програмних для реалізації системи. У ході виконання атестаційної роботи в хмарному середовищі AWS Sagemaker було розроблено концепт системи.

Як подальший розвиток цього проекту, буде розроблений інтернет ресурс, в якому будь який користувач зможе перевірити себе на наявність чи відсутність серцевих проблем за заданими параметрами.

Джерела

1. Data Mining: Practical Machine Learning Tools and Techniques [Електронний

ресурс] Доступно: https://www.cs.waikato.ac.nz/~ml/weka/book.html

2. A. K. Jain. Data clustering: a review / A. K. Jain, M. N. Murty, P. J. Flynn.– ACMComput. Surv.–1999. – No31. - 60 с.

3. Моделі та методи прийняття рішень : навч. посіб. для студ. вищ. навч. закл. /О.Ф. Волошин, С.О. Мащенко. – 2-ге вид., перероб. та допов. – К. :Видавничо-поліграфічний центр «Київський університет», 2010. – 336 с.

4. Методи кластеризації [Електронний ресурс] Доступно: http://pzs.dstu.dp.ua/DataMining/cluster/index.html

5. Charu C. Aggarwal Data Mining. The Textbook. / C.A.Charu; — Springer, 2015 .—746 p.

6. Python Documentation [Електронний ресурс] Доступно: https://www.python.org/doc/

7. Основи програмування. Python. Частина 1[Електронний ресурс]: підручник для студ. спеціальності 122 «Комп’ютерні науки», спеціалізаії «Інформаційні технології в біології та медицині» / А.В. Яковенко; КПІ ім. Ігоря Сікорського. –Електронні текстові данні (1 файл: 1,59 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 195 с.

8. Machine Learning - Машинне навчання. [Електронний ресурс].Доступно:https://www.it.ua/knowledge-base/technology-innovation/machine-learning

9. Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2

10. Khurana, P.; Sharma, S.; Goyal, A. Heart Disease Diagnosis: Performance Evaluation of Supervised Machine Learning and Feature Selection Techniques. In Proceedings of the 8th International Conference on Signal Processing and Integrated Networks,SPIN 2021, Matsue, Shimane Prefecture, Japan, 18–22 October 2021.

11. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.

12. AWS Sagemaker [Електронний ресурс].Доступно: https://docs.aws.amazon.com/sagemaker/index.html

13. XGBoost [Електронний ресурс].Доступно: https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost.html

14. Principal Component Analysis [Електроннийресурс] . Доступно:https://docs.aws.amazon.com/sagemaker/latest/dg/pca.html

15. Amazon Cloud [Електронний ресурс].Доступно: https://docs.aws.amazon.com/

Додаток А

Частина лістинку коду використання PCA

```
pca = sagemaker.estimator.Estimator(container,
                                    role,
                                    train_instance_count=1,
                                    train_instance_type='ml.c4.xlarge',
                                    output_path=output_location,
                                    sagemaker_session=sagemaker_session)



pca.set_hyperparameters(feature_dim=11,
                        num_components=6,
                        subtract_mean=False,
                        algorithm_mode='regular',
                        mini_batch_size=100)

pca_reduction.content_type = 'text/csv'
pca_reduction.serializer = csv_serializer
pca_reduction.deserializer = json_deserializer

result = pca_reduction.predict(np.array(df_final))

result

predictions = np.array([r['projection'] for r in result['projections']])

predictions

predictions.shape
```

Додаток В

Частина лістинку коду використання XGBoost

```
Xgboost_classifier = sagemaker.estimator.Estimator(container,
                            role,
                            train_instance_count=1,
                            train_instance_type='ml.m4.xlarge',
                            output_path=output_location,
                            sagemaker_session=sagemaker_session)

Xgboost_classifier.set_hyperparameters(max_depth=3,
                        objective='multi:softmax',
                        num_class= 2,
                        eta = 0.5,
                        num_round = 150
                        )

train_input = sagemaker.session.s3_input(s3_data = s3_train_data,
content_type='csv',s3_data_type = 'S3Prefix')
valid_input = sagemaker.session.s3_input(s3_data = s3_valid_data,
content_type='csv',s3_data_type = 'S3Prefix')

Xgboost_classifier.fit({'train': train_input, 'validation': valid_input})
```

Додаток С

Діаграми параметрів