РЕФЕРАТ/ABSTRACT

Кваліфікаційна робота магістра містить: с. – __, рис. – __, табл. – ___, джерел – __.

МЕДИЧНІ АНАЛІЗИ, АЛГОРИТМ, АНСАМБЛЕВІ МЕТОДИ, ВИЯВЛЕННЯ АНОМАЛІЙ, PYTHON, МАШИННЕ НАВЧАННЯ

Об'єктом дослідження є медичнадокументація, формується на основі оглядів пацієнтів лікуючим лікарем.

Ціль роботи - розробка алгоритмічного і програмного забезпечення аналізу і візуалізації результатів медичних оглядів

У ході дослідження був проведено аналітичний огляд по літературним джерелам з метою уточнення досягнень світовийтехнологічної науки в галузі, що розглядається; аналіз та візуалізація результатів медичних оглядів; визначення аномалій у результатах медичні дослідження.

У результаті дослідження проведено аналіз і візуалізація результатів медичних оглядів; розроблений алгоритм визначення аномалій в результатах медичних досліджень.

Область застосування: охорона здоров'я

Економічна ефективність/значимість роботи - застосування алгоритму дозволить автоматично визначати аномалії в медичних даних, що скоротить час роботи фахівця, що витрачається на данупроцедуру.

У майбутньому планується робота по впровадження алгоритму в медичні системи охорони здоров'я.

MEDICAL ANALYSIS, ALGORITHM, ENSEMBLE METHODS, ANOMALITY DETECTION, PYTHON, MACHINE LEARNING

The object of the study is medical documentation, formed on the basis of examinations of patients by the attending physician.

The goal of the work is the development of algorithmic and software analysis and visualization of the results of medical examinations

In the course of the study, an analytical review of literary sources was conducted in order to clarify the achievements of world technological science in the field under consideration; analysis and visualization of the results of medical examinations; determination of anomalies in the results of medical research.

As a result of the research, an analysis and visualization of the results of medical examinations was carried out; developed an algorithm for determining anomalies in the results of medical research.

Scope: health care.

Economic efficiency/significance of work - the application of the algorithm will allow to automatically identify anomalies in medical data, which will reduce the specialist's time spent on this procedure.

In the future, work is planned to implement the algorithm in medical health care systems.

ЗМІСТ

ВСТУП

З моменту появи інформаційних технологій у багатьох сферах діяльності були впроваджені зміни, які покращують, оптимізує та автоматизує певні процеси. Галузь охорони здоров'я також застосовує інформаційні технології для удосконалень, а саме використання інформаційних систем для введення інформації і подальшого формування звітності і документації. Також аналітика даних і машинне навчання дозволяють отримувати необхідну і корисну інформацію з медичних досліджень.

Виявлення аномалій - важлива задача, яка широко вивчається в багатьох областях, пов'язаних з інтелектуальним аналізом даних. Застосування в охороні здоров'я може виявити проблеми зі спалахами захворюваності, неправильним діагнозом, і, отже, неправильним лікуванням. З машинним навчанням, ми можемо навчити алгоритм класифікувати аномальні дані, вивчаючи їх характеристики. Дані надаються в різних форматах: таблиці, зображення, аудіо, відео, текст тощо. Кожен тип представлення даних має свій підхід до обробки та аналізу даних. Тернопільським національним медичним університетом ім І.Я. Горбачевського були надано деперсоналізовані медичні дослідження. Які були застосовані в даній роботі.

Метою роботи є підвищення ефективності галузі охорони здоров'я під час аналізу медичних досліджень шляхом розробки алгоритму виявлення аномальних даних. В рамках роботи здійснюється обробка та підготовка медичних досліджень до аналізу. Також пропонується алгоритм виявлення аномальних даних в медичних дослідженнях.

Об'єкт дослідження - медичні документи, сформовані на підставі медичних оглядів.

Предметом дослідження є процес розробки алгоритмічного і програмного аналізу результатів медичних обстежень.

Методи дослідження - пошук літератури і джерел, аналіз інформаційних матеріалів, порівняння, консультація зі спеціалістами, матеріали та інтернет ресурси. Робота буде реалізована на мовою програмування "Python".

1. ОСНОВНА ЧАСТИНА

1.1 Огляд літератури

Багато сфери діяльності вже знайомі з поняттям «великідані». Медичні, освітні, фінансові та інші установи генерують великі обсяги інформації. До жаль, не Усе можуть правильно працювати з таким обсягом даних, в більшості випадків дані не зберігаються або зберігаються без подальшого застосування і аналізу.

Міжнародною консалтинговою компанією McKinsey визначила 11 методів і прийомів, які застосовуються до великих даних, які, зокрема, включають в себе машинне навчання і візуалізація аналітичних даних [2]. Машинне навчання відноситься до побудови алгоритмів, які можливо навчити за допомогою математичного апарату теорії імовірності та математичної статистики, теорії графів, чисельних методів та інших. Візуалізація аналітичних даних – це відображення вихідної інформації та результатів аналізу даних у найбільш зручній для сприйняття і інтерпретації формі [14].

Алгоритми виявлення аномалій лежать в основі будь-який системи контролю якості даних Під аномалією розуміється відхилення від номінальних умов експерименту.

В цій роботі представлені основні теорії машинного навчання, описані алгоритми, що використовуються для аналізу медичних даних, а також дані, використовуються в цій роботі.

Інтелектуальний аналіз даних (Data Mining) – це область, яка дозволяє виділити нову значиму інформацію в великому кількості даних. Основна мета полягає у застосуванні методів опрацювання даних для виявлення явно невидимих взаємозв'язків та несподіваних змін при використанні їх.

Знаходження аномалій -це пошук невідповідних даних не відповідних певному розумінню нормальної, очікуваної поведінки. Процес пошукуаномалій складається з визначення області, яка представляє нормальне поведінку, та визначення аномалій у даних, які не належать до цієї нормальної області. Виходячи з цього, аномалії

визначаються не по їх характеристиках, а в порівняння з тим, що є нормою [1].

Одним з етапів роботи є поділ медичних досліджень на два класи: нормальні дані і дані з аномаліями. Будь-яке класифікаційне завдання ставиться наступним чином:існують об'єкти, поділені на класи. Є кінцевий набір, для якого відомо, до яких класів належать об'єкти цього набору. Цей набір називають навчальним (training set). Клас решти об'єктів невідомий. Потрібно побудувати алгоритм, котрий назве ім'я класу, до якого належить вільний об'єкт із вихідного набору [7]. Для перевірки роботи навченого алгоритму необхідно підготувати тестову вибірку (validation set). Саме на цьому наборі буде оцінюватися точність класифікації.

На рисунку 1.1 наведено найбільше поширена схема, яка містить основні етапи роботи таких алгоритмів:

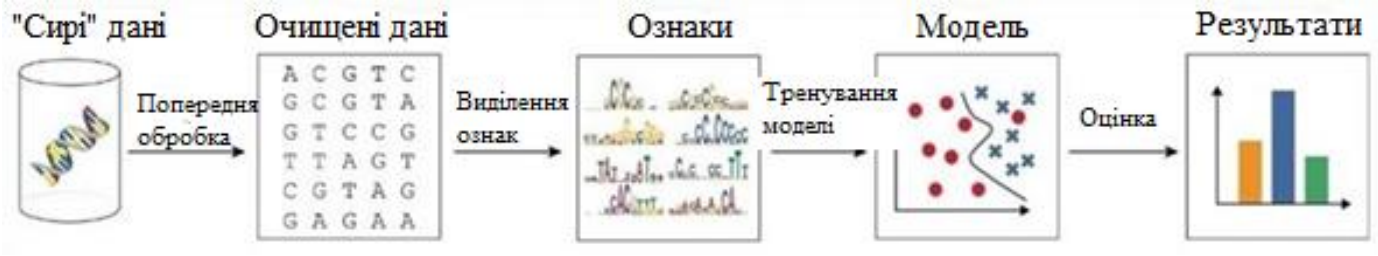


Рисунок. 1.1.– Класична схема етапів машинного навчання

Існує безліч наборів даних, містять медичні дані, як в нормі, так і при патології. Після вивчення кількох прикладів наборів даних, перевага було надана наборам даних, отриманих у ТНМУ ім. І.Я.Горбачевського [3] і завантаженим із інформаційної платформи "Kaggle".

1.1.1 Платформа Kaggle.

Спочатку Kaggle, який позиціонує себе як «ваш будинок для науки про дані», був місцем для змагань з машинного навчання, але зараз там можна знайти ресурси з науки про дані. Варто згадати кілька головних особливостей Kaggle:

Datasets (набори даних): безліч наборів даних різних типів та розмірів, які

можна безкоштовно завантажити. Тут можна знайти цікаві дані вивчення чи тестування своїх навичок моделювання.

Machine Learning Competitions (змагання з машинного навчання): колись були серцем Kaggle, такі тести на моделювання — найкращий спосіб вивчити нові види машинного навчання та відточити ваші здібності за допомогою цікавих проблем, що базуються на реальних даних.

Learn (Навчання): серія навчальних матеріалів з вивчення даних, що охоплюють SQL та глибоке навчання (Deep Learning), що подаються в Jupyter Notebooks.

Discussion (обговорення): місце, де можна поставити свої питання та отримати поради від тисяч експертів за аналітичними даними (data scientist) у спільноті Kaggle.

Kernels (ядра): онлайн-середовище для програмування, яке працює на серверах Kaggle. У ній можна писати Python/R-скрипти та працювати в Jupyter Notebooks. Такі ядра абсолютно безкоштовні (можна навіть додати GPU) та ідеальні для тестування: не потрібно настроювати середовище у себе на комп'ютері. Їх можна використовувати для аналізу будь-якого набору даних, змагань з машинного навчання або виконання завдань розділу «Навчання». Можна скопіювати або змінити існуюче ядро іншого користувача, а також поділитися своїм, щоб його змогли оцінити інші.

Короткий опис вкладок:

Overview: короткий опис проблеми, метрики, оцінки, призи та часова шкала.

Data: всі дані, необхідних участі в конкурсі, інші дані не допускаються. Можна завантажити всю інформацію до себе на комп'ютер, але ми цього робити не будемо, оскільки використовуватимемо ядро Kaggle, до якого можна просто підключити дані.

Kernels: попередні роботи, зроблені вами чи іншими учасниками. Для змагання це найважливіший ресурс. Ви можете вивчити інші скрипти та notebooks, а потім скопіювати (називається Forking), щоб змінити їх і запустити.

Discussion: ще одна корисна вкладка, де можна знайти обговорення, в яких беруть участь як організатори, так і учасники змагання. Тут можна поставити уточнюючі питання або отримати нову інформацію з відповідей іншим учасникам.

Leaderboard: можна подивитися, хто в топі, та ваш рейтинг.

Rules: правила не дуже цікаво, але їх варто знати.

Team: тут можна керувати членами команди, якщо ви вирішите створити свою.

My Submissions: можна переглянути ваші попередні матеріали та вибрати фінальний варіант, який братиме участь у змаганні.

Змагання Kaggle з машинного навчання, хоч вони і називаються так, варто називати скоріше «спільними проектами», тому що головною метою є не так виграти, як попрактикуватися і повчитися в друга-експерта. Як тільки ви усвідомлюєте, що тут головне – не перевершити інших, а покращити свої навички, ви отримаєте від змагань максимальну користь. Коли ви реєструєтеся на Kaggle, ви отримуєте не тільки доступ до всіх ресурсів, але й можливість стати частиною спільноти експертів з аналітичних даних.

Можна як поділитися своїми доробками ядра, так і поставити питання у гілці обговорень. Звичайно, перспектива викласти свою роботу в загальний доступ лякає, але це дозволить отримати відгук на свою роботу та виправити існуючі помилки, а також не робити їх у майбутньому. Усі починають, як новачки, а спільнота експертів за аналітичними даними дуже підтримує своїх на всіх рівнях підготовки.

Створення нових обговорень та використання чужого ядра не тільки не забороняється, а й заохочується. Це надзвичайно важлива навичка командної роботи.

Найкращий спосіб участі у змаганні — знайти чуже ядро з хорошим результатом у таблиці лідерів, скопіювати його та спробувати покращити результат. Потім поділитися своїм ядром із спільнотою, щоб інші могли використати його. Експертна спільнота за аналітичними даними стоїть не на плечах атлантів, а на спинах тисяч людей, які поділилися своєю роботою з іншими.

Оточення ядра для Notebook. Стандартний Jupyter Notebook в якому можна писати код на Python або звичайний текст (використовуючи синтаксис Markdown) так само, як у Jupyter, а потім запускати код на хмарному сервері Kaggle. Однак ядра Kaggle мають деякі відмінні риси, недоступні в Jupyter Notebook.

На вкладці Data відображаються набори даних, до яких наше ядро підключено. У цьому випадку ми маємо всі дані зі змагання, але ми також можемо підключити інші дані з Kaggle або завантажити свої. Файли з даними лежать у директорії ../input/.

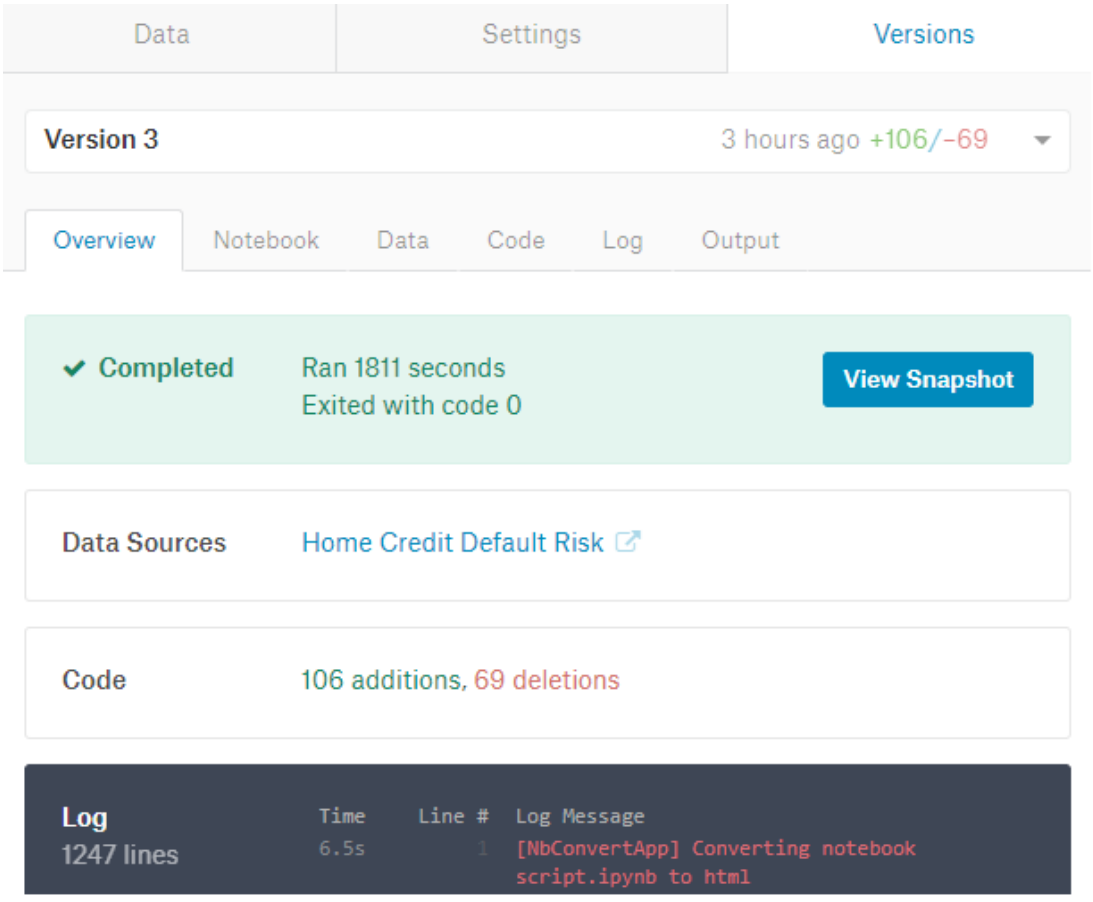
Код доступу до них:

```
import os
# List data files that are connected to the kernel
os.listdir('../input/')
```

```
['POS_CASH_balance.csv', 'bureau_balance.csv', 'application_train.csv', 'previous_application.csv', 'installm
ents_payments.csv', 'credit_card_balance.csv', 'sample_submission.csv', 'application_test.csv', 'bureau.csv']
```

Вкладка Settings дозволяє контролювати різні технічні аспекти ядра. Ми можемо додати GPU, змінити видимість або встановити Python, якого ще немає в оточенні.

Остання вкладка Versions дозволяє переглянути попередні коміти. Ми можемо дивитися зміни в коді, переглядати лог-файли запуску, бачити notebook, згенерований під час запуску, і завантажувати вихідні дані прогону.



Щоб запустити весь notebook та записати нову версію, потрібно натиснути блакитну кнопку Commit & Run у правому верхньому кутку ядра. Ця дія виконає весь код та збереже будь-які файли, які будуть створені під час запуску. Закомітивши

ноутбук, ми зможемо отримати доступ до будь-яких прогнозів, зроблених нашою моделлю, і подати їх на оцінювання.

Notebook Outline. Потрібно починати з основ: переконатися, що зрозумілі дані та завдання, що використовуються. Для цієї проблеми є один головний тренувальний набір даних із вже розставленими мітками та 6 додаткових файлів. У першому notebook ми використовуємо лише головний файл, який принесе гідний результат, але подальша робота передбачає, що ми будемо використовувати всі файли з даними, щоб бути конкурентоспроможними.

Щоб зрозуміти дані, варто почитати документацію, наприклад опис колонок кожного файлу. Так як використовується кілька файлів, потрібно зрозуміти, як вони взаємопов’язані, хоча для першого ноутбука ми будемо використовувати один файл, щоб спростити роботу. Читання інших ядер також допоможе нам ознайомитися з даними та зрозуміти, які змінні важливі.

Як тільки ми розібралися з даними та проблемою, ми можемо почати структурувати завдання машинного навчання. Це передбачає роботу з категоріальними змінними (через one-hot encoding), заповнення пропущених значень (imputation) та масштабування змінних у діапазоні. Ми можемо проводити аналіз дослідницьких даних, наприклад, пошук закономірності з ярликом, і малювати такі закономірності.

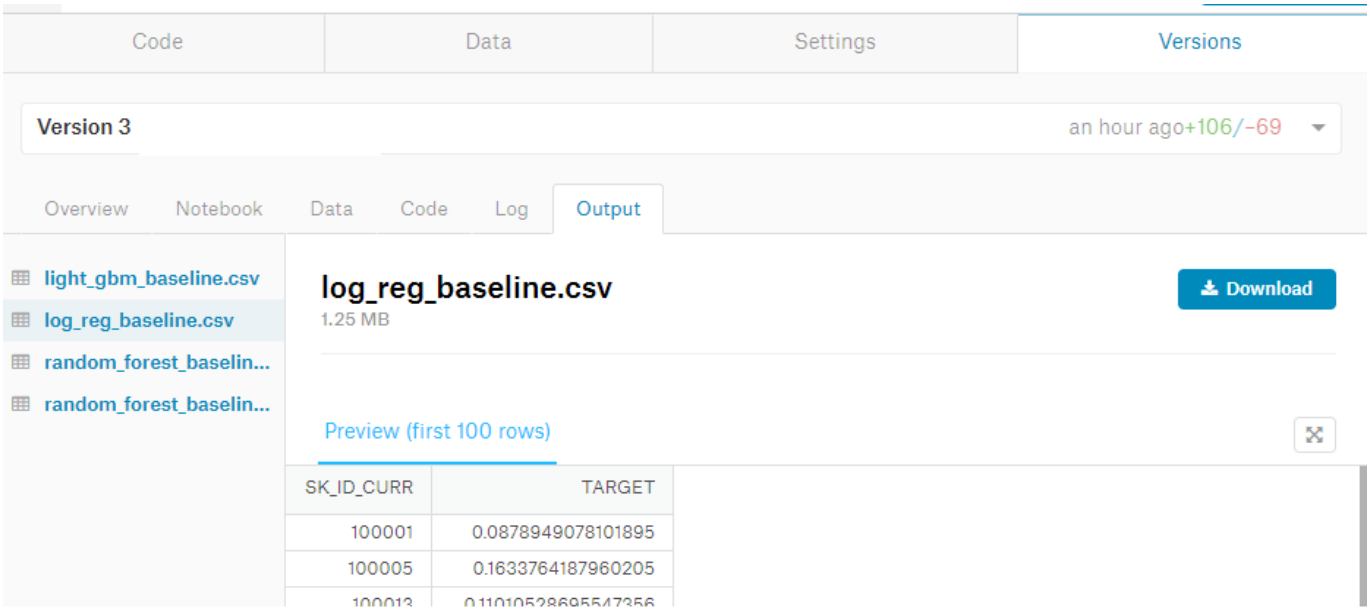
Після ретельного вивчення даних та забезпечення прийнятності для машинного навчання ми переходимо до створення базових моделей. Однак, перш ніж повністю перейдемо до етапу моделювання, важливо зрозуміти показник продуктивності для змагань. У змаганні Kaggle все зводиться до одного числа - метрики за тестовими даними.

Хоча інтуїтивно здається, що потрібно використовувати точність для завдання бінарної класифікації, це буде неправильно, оскільки тут проблема дизбалансу класу. Замість точності рішення оцінюються за допомогою ROC AUC (Receiver Operating Characteristic curve Area Under the Curve). Щоб вести підрахунки за допомогою ROC AUC, нам потрібно робити прогнози в термінах ймовірностей, а не бінарні — 0 або 1. ROC показує справжню позитивну оцінку порівняно з помилково позитивною

оцінкою як функцію порога, згідно з яким ми класифікуємо екземпляр як позитивний.

Зазвичай потрібно робити базове передбачення, але в даному випадку випадкові здогади по завданню дорівнюють 0,5 по ROC AUC. Тому для нашої моделі ми будемо використовувати дещо складніший метод — логістичну регресію. Це популярний простий алгоритм задач бінарної класифікації, який допоможе встановити низький поріг для проходження майбутніми моделями.

Після застосування логістичної регресії можна зберегти результат у csv-файл для відправки. Коли ноутбук закоммічений, будь-які вихідні файли з'являться на вкладці Output Versions.



З цієї вкладки ми можемо завантажити файли на комп'ютер, а потім вивантажити їх для участі в змаганні. У цьому notebooky ми зробили чотири різні моделі. Gradient Boosting Machine (використовувалась бібліотека LightGBM) відпрацював найкраще. Ця модель перемагає практично у кожному структурованому Kaggle змаганні (де дані представлені в табличному форматі) і нам, ймовірно, доведеться використати якусь форму цієї моделі, якщо ми хочемо гідно конкурувати з іншими учасниками.

Отже, ключовим критерієм вибору була кількість даних, яка міститься в наборі,

оскільки однієї з особливостей алгоритмів навчання є потреба в максимально можливій кількості вихідних даних. Вибрані набори даних в сукупності містять трохи більше 1500 тисяч даних і знаходяться в вільному доступі [5].

1.2 Об'єкт і методи дослідження

1.2.1. Методи машинного навчання (Machine Learning).

Дані методи відносяться до класу методів штучного інтелекту, з не прямим вирішенням проблеми, а навчанням у процесі застосування рішень багатьох подібних проблем. Використовується, коли програмування високопродуктивних алгоритмів важко або неможливо. Якість і повнота навчальною вибірки безпосередньо залежить від якості налаштування системи. Машинне навчання можна використовувати при вивченні закономірностей в даних, які потім використовуються для виявлення аномальної поведінки.

Машинне навчання – це один із підрозділів штучного інтелекту, який вивчає та розробляє алгоритми та моделі, що надають дозвіл комп'ютерам навчатися прогнозувати чи приймати рішення на основі наявних даних, без явного програмування.

На відміну від традиційного програмування, де розробник явно задає правила та інструкції для виконання завдання, у машинному навчанні комп'ютер самостійно витягує закономірності та узагальнює інформацію з даних, щоб приймати рішення чи робити прогнози.

Основна ідея такого навчання полягає у створенні моделі, яка може навчатися на основі досвіду та даних, та використовувати цю модель для вирішення нових завдань або передбачення результатів.

Машинне навчання широко застосовується, включаючи розпізнавання образів, класифікацію даних, прогнозування, кластеризацію, рекомендаційні системи та багато іншого.

У машинному навчанні є кілька основних типів завдань, які можна розв'язувати за допомогою алгоритмів. Ось деякі з них:

Класифікація – це завдання, де модель повинна віднести об'єкти до певних класів або категорій на основі відомих даних. Наприклад, модель класифікує, чи є певний пацієнт хворим чи здоровим.

Регресія – це завдання, у якому модель має передбачити чисельне значення з урахуванням наявних даних. Наприклад, модель може передбачати ціну нерухомості на основі її характеристик або передбачати кількість продажів залежно від рекламних витрат.

Кластеризація – тут модель має групувати об'єкти з урахуванням їх подібності. Наприклад, модель може кластеризувати користувачів інтернет-магазину на основі їхньої купівельної поведінки, щоб виявити сегменти клієнтів.

Виявлення аномалій – модель має визначити незвичайні чи аномальні об'єкти даних. Наприклад, модель може виявити шахрайські транзакції на основі звичайних покупок.

Рекомендаційні системи – системи, де модель повинна пропонувати рекомендації користувачеві на основі його переваг та поведінки. Наприклад, модель може рекомендувати фільми чи товари на основі історії переглядів чи покупок користувача.

Обробка природної мови – в даному алгоритмі модель має аналізувати та розуміти природну мову, таку як текстові документи чи розмови. Наприклад, модель може класифікувати тексти на теми або визначати тональність тексту.

Це лише деякі з типів завдань, які можна вирішувати за допомогою машинного навчання. Кожне завдання вимагає свого підходу та алгоритму.

Основні алгоритми машинного навчання. У машинному навчанні існує безліч алгоритмів. Нижче наведено деякі з них:

Лінійна регресія – це алгоритм, який використовується для передбачення безперервної залежної змінної на основі однієї чи кількох незалежних змінних. Він будує лінійну модель, яка найкраще відповідає даним.

Логістична регресія – алгоритм, який використовується для класифікації даних.

Він передбачає можливість приналежності об'єкта до певного класу. Логістична регресія використовує логістичну функцію для перетворення лінійної комбінації ознак на ймовірність.

Вирішальні дерева – це алгоритми, що використовуються для класифікації та регресії. Вони являють собою структуру у вигляді дерева, де кожен вузол є тестом на одній з ознак, а кожна гілка відповідає можливому значенню цієї ознаки. Вирішальні дерева дозволяють робити прогнози, ґрунтуючись на послідовному застосуванні тестів.

Опорні вектори (SVM) – побудова гіперплощини у багатовимірному просторі, яка поділяє різнокласові об'єкти та намагається знайти оптимальну гіперплощину розділяючи максимально класи.

Наївний Баєсівський класифікатор – в основі закладена теорема Байєса і робить припущення щодо незалежності ознак. Він обчислює ймовірність приналежності об'єкта до кожного класу та вибирає клас із найбільшою.

Кластерний аналіз –використовується з метою групування об'єктів з врахуванням їхньої подібності. Він дозволяє виявити приховані структури даних і розділити об'єкти на кластери та може бути використаний при сегментації.

Слід зазначити, що алгоритми мають певні індивідуальні особливості.

Принципи роботи алгоритмів полягають в адаптованості моделі до даних та передбачення чи прийнятті рішення:

– Навчання на основі даних.

Основна ідея – навчити модель з урахуванням наявних даних. Модель аналізує дані, знаходить закономірності та шаблони, та використовує їх для прогнозування чи прийняття рішень. Чим більші та якісніші дані, тим краще модель може навчитися.

– Вибір та налагодження моделі

При виборі моделі необхідно враховувати тип завдання, доступні дані та вимоги до точності передбачень. Після цього необхідно налаштувати її параметри, щоб вона краще відповідала даним та давала найкращі результати.

– Поділ даних на навчальну та тестову вибірки

Для оцінки якості моделі дані поділяють на навчальну вибірку, яка

використовується для навчання моделі та тестову– щоб встановити її точність та ефективніст. При цьому оцінюється, чи добре модель узагальнює дані та чи може передбачитинові дані.

– Оцінка та покращення моделі

Після навчання та перевірки моделі на тестовій вибірці необхідно оцінити її якість та ефективність. Для цього використовуються точність, повнота, F-міра та інші. Якщо модель не дає достатньо хороших результатів, можна спробувати її поліпшити, змінюючи параметри моделі, додаючи нові ознаки або застосовуючи інші методи обробки даних.

– Обробка та попередня обробка даних

Один із важливих кроків у машинному навчанні, включає видалення викидів, заповнення пропущених значень, масштабування ознак та інші методи обробки даних. Метою передобробки даних є узгоджений набір даних, який використовується для навчання моделі.

Переваги машинного навчання :

– Автоматизація та оптимізація процесу

Працюють із складними процесами, які раніше вимагали великої кількості часу та ресурсів.

– Обробка великих об’ємів даних

Можливість обробляти та аналізувати великі об’єми даних. Це необхідно в областях, де дані є важливими, наприклад, медицині, фінансах або маркетингу.

– Пошук прихованих закономірностей та патернів у даних

Це дозволяє виявляти нові знання та робити передбачення з врахуванням цих закономірностей.

– Підвищення точності та якості передбачень

Створює прогнозуючі моделі. Це особливо корисно у завданнях прогнозування, класифікації та кластеризації.

Недоліки:

– Залежність від якості даних

Машинне навчання вимагає наявності якісних та узгоджених даних для

навчання моделі. Якщо дані містять помилки, шум або невідповідність, це може призвести до неправильних передбачень і низької якості моделі.

– Необхідність великих обчислювальних ресурсів

Деякі алгоритми машинного навчання вимагають використання потужних комп'ютерів чи графічних процесорів.

– Складна інтерпретація результатів

Деякі моделі машинного навчання можуть бути складними в інтерпретації та поясненні результатів..

– Нестача гнучкості

Окремі моделі машинного навчання є жорсткими і не гнучкими для адаптації до умов, що змінюються, або нових даних. Це може погіршити якість передбачень або нездатність моделі адекватно реагувати на нові ситуації.

Застосування машинного навчання :

– Рекомендаційні системи

Багато платформ та сервісів (Netflix, Amazon, Spotify), використовують його для створення рекомендацій для своїх користувачів.

– Обробка природної мови

Дозволяє комп'ютерам розуміти та генерувати текст. Це застосовується в автоматичному перекладі, аналізі тональності тексту, чат-ботах та інших додатках.

– Медицина

Допомага в діагностиці захворювань, прогнозуванні їх розвитку та визначенні найбільш ефективного лікування. Алгоритми можуть аналізувати медичні дані, такі як знімки, результати аналізів та історії пацієнтів, щоб допомогти лікарям приймати більш точні рішення.

В цілому, машинне навчання є важливою та перспективною областю, яка продовжуватиме розвиватися та застосовуватись у майбутньому.

1.2.2. Навчання з вчителем (Supervised learning)

Це підкатегорія, яка поєднує машинне навчаня та штучний інтелект, в

залежності від того, яким чином вхідні дані вводяться до моделі, вона коригує свої характеристики до тих пір, поки модель не буде належно підігнана, а помилка буде мінімізована. Воно використовується при класифікації, прогнозуванні та для передбачення, напрклад, купівельної спроможності з огляду на попередні моделі. За допомогою контрольного навчання можна вирішити реальні проблеми, наприклад, відокремлювати результати аналізів крові.

1.2.3. Навчання без вчителя (Unsupervised learning)

Використовується для дослідницького аналізу даних, але не як основний алгоритм. Алгоритм спочатку навчання отримує немарковані дані в якості навчального набору. У процесіроботи алгоритм знаходить закономірності та схожі дані, і не пов'язує їх, таким чином отримуючи можливість знайти цікаві і несподівані дані,дізнатися про дані більше, які раніше не були виявлено.

При навчанні вивчається масив даних, в яких виявляються приховані взаємозв'язки між змінними. Метод використовується як математичний процес для систематичного зменшення надлишок або структурування даних по подобі.

Цей варіант має своє особливе застосування для компаній, якимнеобхідно, до наприклад, об'єднати дані з різних джерел, щоб створити загальну картину споживачів. До таких завдань можна віднести: кластеризація, зменшення розмірності та вивчення правил асоціації.

1.2.4.Навчання із частковим залученням вчителя (Semi - Supervised learning)

Є змішаним: з вчителем і без нього, у тому випадку, коли відсутні якісні дані. Алгоритм приймає дані як з міткою, так і без її, і в більшості без міток.

Доведено, що це навчання видає точні результати і може застосовуватися до багатьом реальним завданням. Наприклад, розпізнання осіб в великому кількості різних людей на фотографії. Так само, метод може використовуватися для детектування шахрайських дій. Метод дозволяє створювати алгоритми, для

визначення таких аномалії.

1.2.5. Навчання з підкріплен ням (Reinforcement learning)

При навчанні з підкріпленням можливий зв'язок з навколишнім світом. На додаток дозволено самостійне навчання. Цей спосіб є складним, вирішує завдання способом експериментів і помилок, реалізується в світі і отриманні нагород.

Одне із використань навчання є, для прикладу, сортування аналізів. При автоматизації процесу застосовують навчання з підкріпленням і глибоке навчання.

1.2.6. Глибоке навчання (Deep learning)

Навчання використовує програмовану нейронну мережу, яка дозволяє машинам приймати точні рішення без допомоги людини, реалізовуватись як без вчителя, так і з підкріпленням. Глибоке навчання структурує алгоритми пошарово для створення штучної нейроної мережі,яка самостійно навчається і приймає рішення. Подібно до людським висновків.

Нейро-мережі можуть обробляти значну кількість зображень за короткий час та витягувати функції запам’ятовані алгоритмом. До прикладів глибокого навчання відноситься автоматичне розпізнання мови.

1.2.7. Ансамблеві методи (Ensemble Methods)

В ансамблевих методах використовується ідея комбінування кількохслабших прогнозних алгоритмів для отримання якісних прогнозів, чим кожен із алгоритмів може дати як така. Це дуже потужний клас, і тому він дуже популярний. Наприклад, алгоритм випадкового лісу - це спосіб ансамблю, що пов'язує безліч дерев рішень, навчених з різними наборами даних. Підсумки якості моніторингом випадкового ліси краще, ніж якість моніторингів одного дерева рішень.

Методи ансамблю - це як спосіб зменшити дисперсію і систематичну помилку однієї моделі машинного навчання. Одна модель може бути точної при безперечно

заданих умовах, але неточною при інших умовах. З інший моделлю відносна точність може бутизворотної. Об'єднуючи дві моделі, якість прогнозів балансується.

Далі, як методи, використані для виявлення аномалій,будуть розглянуті три алгоритми і їх ансамблі.

2. РОЗРАХУНКИ І АНАЛІТИКА

2.1 Вибір програмного забезпечення

Серед мов програмування по роботі з даними лідируючі позиції займає "Python" [6]. "Python" володіє відкритим вихідним кодом, є платформно незалежними, необхідні пакети легко встановлюються, налаштовуються та застосовуються.

У "Python", при наявності готових бібліотек, крім аналізу даних реалізується візуалізація даних. "Python" є популярним, має значну кількість масивів документації, необхідної розробникам для створення додатків. До недоліків "Python" відносять складність відстеження помилок у коді.

Що стосується простоти використання, то Python можна віднести до тієї групі мов, які підходять на роль першої мови програмування. Перевагою «Python» також є особливості синтаксису, стимулюючі розробника писати зручний для читання код.

2.1.1. Використовувані бібліотеки "Python"

У ході реалізації проекту було використано деякі «Python» бібліотеки для роботи з певними типами даних та використання спеціальних функцій, розширюючих можливості мови.

Бібліотека "NumPy" - це бібліотека, додає підтримку багатомірних масивів і матриць, а також математичних операцій надцими масивами.

Бібліотека "pandas" - високорівнева "Python" бібліотека, побудована поверх бібліотеки "Python". Ця бібліотека надає можливості використання структур «Дата Frame» і "Series". У ході роботи початковий набір даних зберігався і оброблявся в структурі «Дата Frame», а також була використана функція "pandas.read_csv" для зчитування файлуз вихідними даними в структуру "Data Frame".

Бібліотека "Matplotlib" представляє собою модуль-пакет для "Python", з допомогою якого можна, можливо створювати малюнки і графіки різних форматів. У ході даної роботи засоби бібліотеки були використані для візуалізації етапів роботи і

отриманих результатів.

Бібліотека "Scikit - learn" дає можливості реалізації ряду алгоритмів навчання, скорочення розмірності даних, вилучення та відбору ознак. Декілька алгоритмів навчання, а також функції для вилучення і відбору ознак були використані в поточною роботи.

2.2 Завантаження і попередній аналіз даних

Вихідні дані зберігаються у файлах із розширеннями ".xls". Excel- файли містять в собі документи, містять інформацію промедичні дослідження.

З допомогою бібліотеки "pandas" формується об'єкт «Дата Frame», в котрий зчитуються транспоновані таблиці з файлів з розширенням ".xls".

Елементи Data Frame піддаються аналізу: перевірка полів на викиди. Під викидами розуміють такі значення показників, які значно відрізняються від основного безлічі значень.

2.3 Поділ набору даних на тестову і тренувальну вибірки

Для забезпечення перевірки якості роботи моделі класифікації виникає необхідність в поділі спільного набору даних на тренувальну та тестову вибірки. Тренувальна вибірка призначена для навчання моделі, тестова вибірка - для оцінки якості роботи моделі.

Для розбиття набору даних на відповідні вибірки використовується метод "sklearn.model_selection.train_test_split", в якості параметра було вказано, що тестова вибірка буде складати 30% від спільногонабору даних.

2.4 Вибір алгоритму класифікації

У ході роботи для визначення найкращого алгоритму класифікації були розглянуті наступні алгоритми та їх ансамблі:

- "Ізолюючий лліс(Isolation Forest);
- "Однокласовий метод опорних векторів" (One Class SVM);
- «Еліптична огинаюча» (Elliptic Envelope);

2.4.1 Ізолюючий ліс (Isolation Forest)

Це версія ідеї випадкового лісу: проста і надійна. Працює за принципом алгоритму дерева рішень (рис.2.1). Він ізолює викиди, вибираючи функцію випадковим чином із заданогонабору функцій, а потім випадковим чином вибираючи значення поділу між максимальним і мінімальним значеннями обраною функції. Отже, коли ліс випадкових дерев дає більш короткі шляхи для конкретних точок, в такому випадку вони з великий ймовірністю будуть аномальними. Для кожного предмета нормальністю називається середня арифметичне значення глибини листя, в які він ізолювався [8].

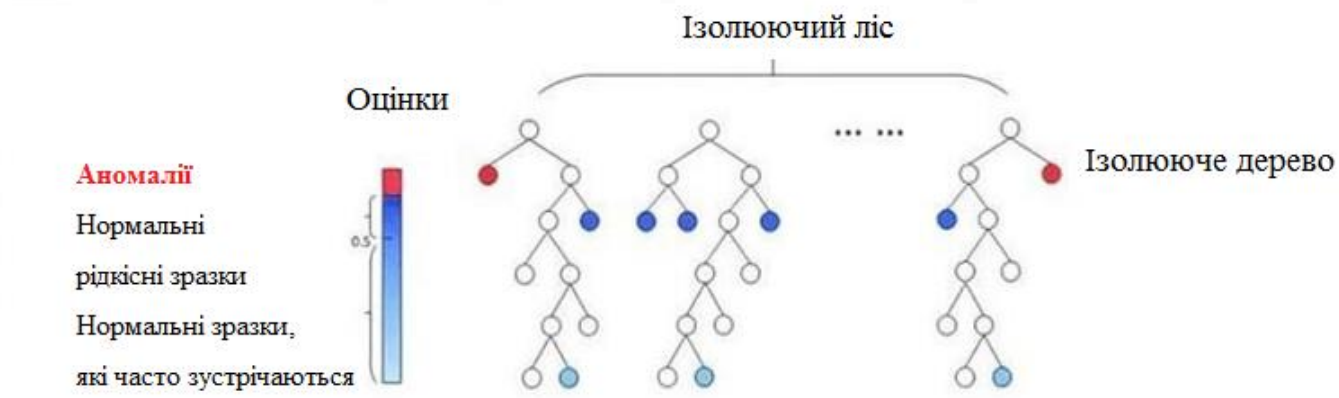


Рисунок 2.1 – Схема виявлення аномальності в алгоритмі «Ізолюючий ліс»

Логіка методу така, що з описаному методі побудови дерев викиди будуть впадати в листя, тобто. викиди легше ізолювати, а дерево буде будуватися поки

довільний об'єкт не виявиться в окремому листі.

Нижче, на рисунку 2.2, показані спостереження декількох наборів даних, отримані в результаті аналізу набору даних, ідентифіковані цим методом як аномалії.

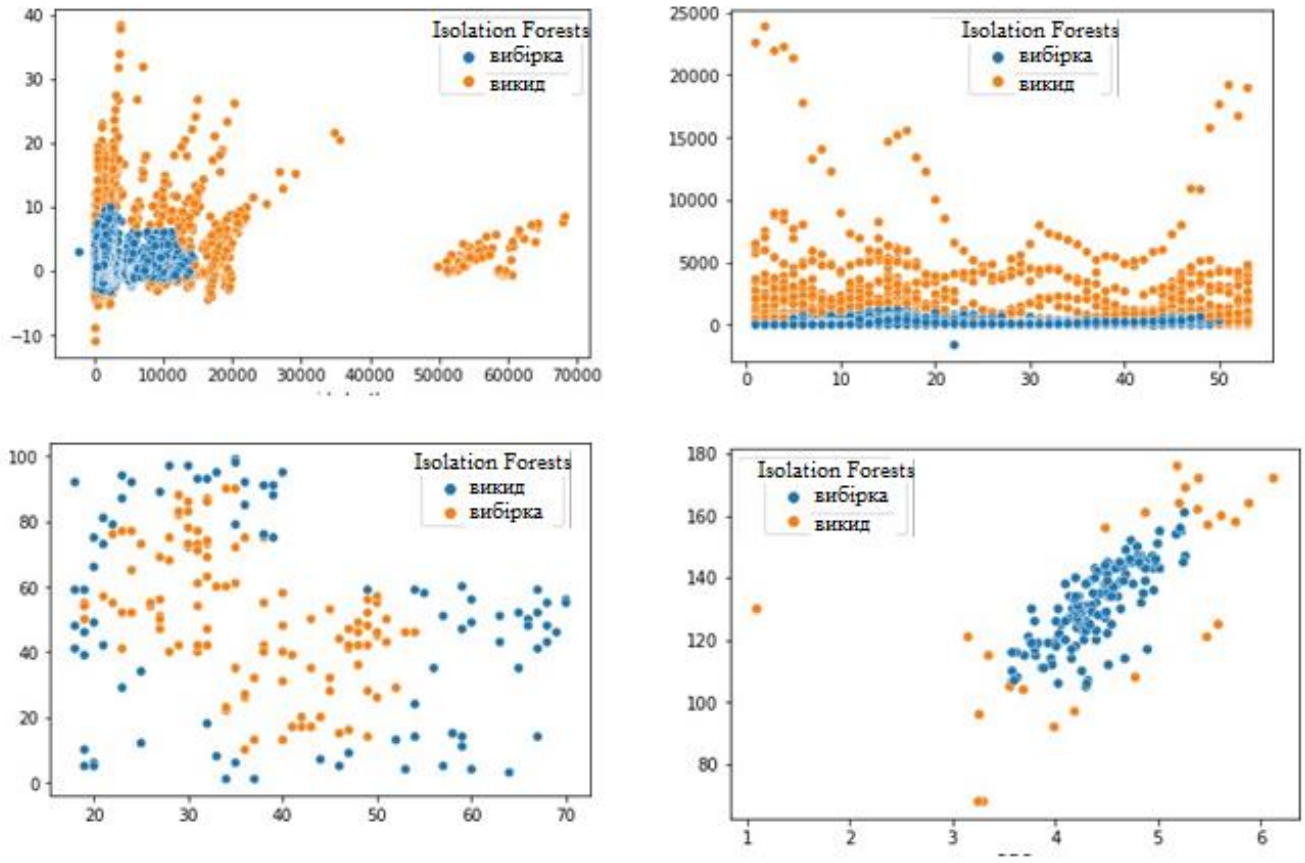


Рисунок 2.2 – Isolation Forests

2.4.2 Однокласовий метод опорних векторів (One Class SVM)

Це один із класичних алгоритмів, але для його навчання нам достатньо всього одного класу [9].

У результаті ми отримуємо границю, по один бік якої максимально щільно зібрані спостереження з вибірки, а по інший, будуть знаходитись аномальні значення (рис. 2.3).

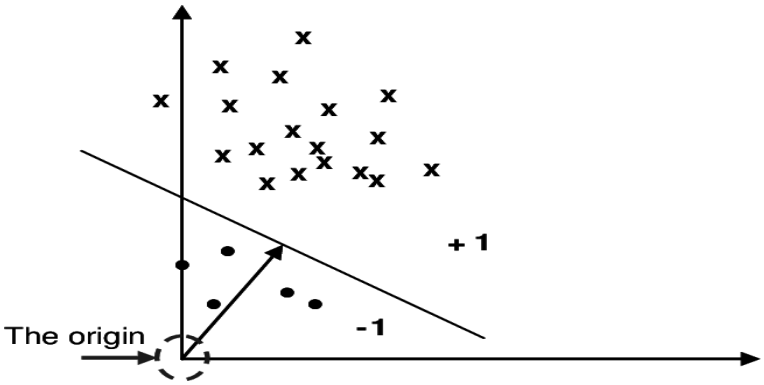


Рисунок 2.3 – Класифікація One Class SVM

На наступному рисунку, наведено спостереження кількох наборів даних , які отримані при аналізі набору даних, ідентифікованих, за даним методом, як аномалії.

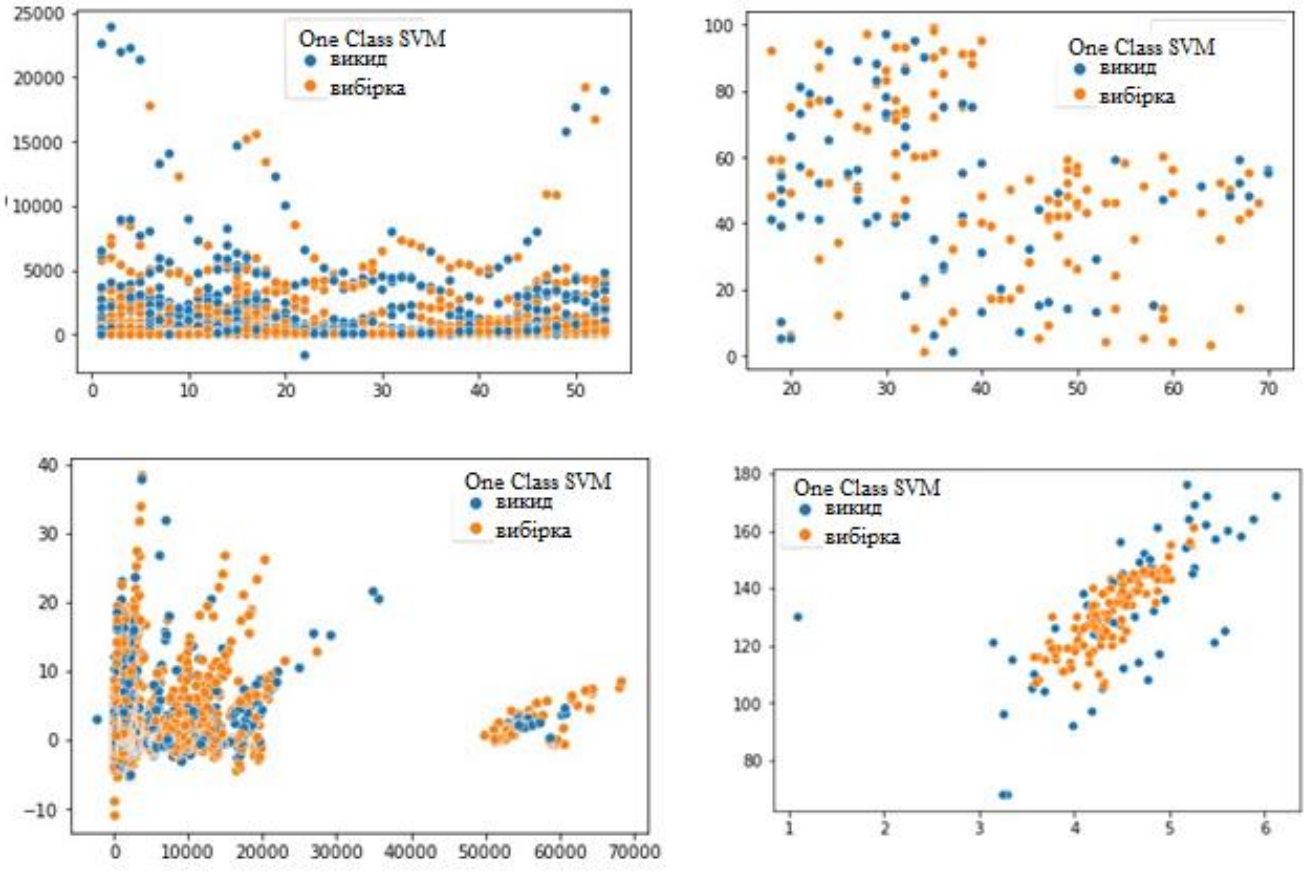


Рисунок 2.4 – One Class SVM

2.4.3. Алгоритм еліптичної обгинальної (Elliptic Envelope)

Даний алгоритм створює уявну еліптичну область навколо набору даних. Дані,

попадаючі в цю область, вважаються нормальними, а все, що знаходиться поза межами, повертається як викиди. Алгоритм працює краще всього, якщо дані мають гаусівський розподіл [10].

Далі наведено спостереження кількох наборів даних , отриманих при аналізі набору та ідентифіковані даними методом як аномалії.

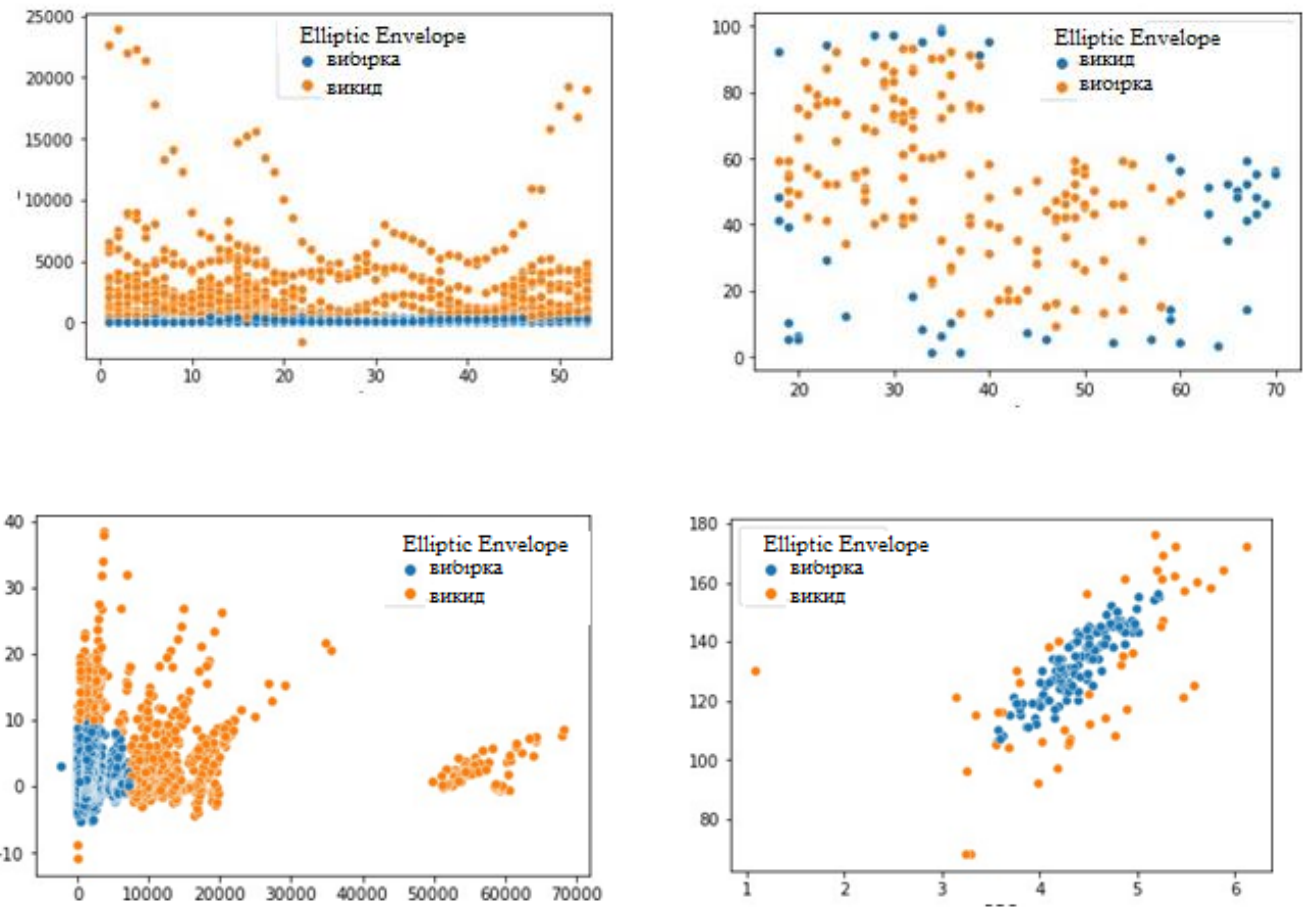


Рисунок 2.5 – Викиди Elliptic Envelope

2.4.4 Ансамблі алгоритмів

Ансамблі (Ensemble, Multiple Classifier System) це тип машинного навчання , в якій декілька моделей навчаються для вирішення однієї і тої самої проблеми та об'єднуються для отримання найкращих результатів. Основна гіпотеза наступна – при вірному поєднанні слабких моделей ми отримуємо точніші та більш надійні моделі [12]. Процес побудови ансамблів називається ансамблевим навчанням (ensemble

learning) [11]. Самий простий приклад ансамблю в регресії - усереднення кількох алгоритмів:

$$\mathrm{a(x)} = \frac{1}{\mathrm{n}}(b_1(x) + \ldots + b_n(x)) \tag{2.1}$$

Алгоритми, що становлять ансамбль (b_n) називаються базовими алгоритмами (base learners). Якщо ми розглянемо значення базових алгоритмів на об'єкті як незалежні випадкові величини з однаковим середнім значенням і однаковою кінцевою дисперсією стане ясно, що випадкова величина (2.2) має таке ж середнє значення, але меншу дисперсію [11]:

$$\begin{aligned} &\xi = \frac{1}{n}(\xi_1 + \ldots + \xi_n); \\ &\mathrm{E}\,\xi = \frac{1}{n}(\mathrm{E}\xi_1 + \ldots + \mathrm{E}\xi_n) = \frac{\mathrm{E}\xi_i}{n}; \\ &\mathrm{D}\,\xi = \frac{1}{n^2}(\mathrm{D}\,\xi_1 + \ldots + \mathrm{D}\,\xi_n) = \frac{\mathrm{D}\,\xi_i}{n^2}. \end{aligned} \tag{2.2}$$

Для визначення найкращого алгоритму класифікації були розглянуті наступні ансамблі алгоритмів, які ідентифіковані ансамблевим методом як аномалії:

1. Isolation Forest + One Class SVM.

На рисунку 2.6, наведено спостереження кількох наборів даних, які отримані внаслідок їх аналізу. 2 методи SVM+IF викид 2 викид 1 вибірка

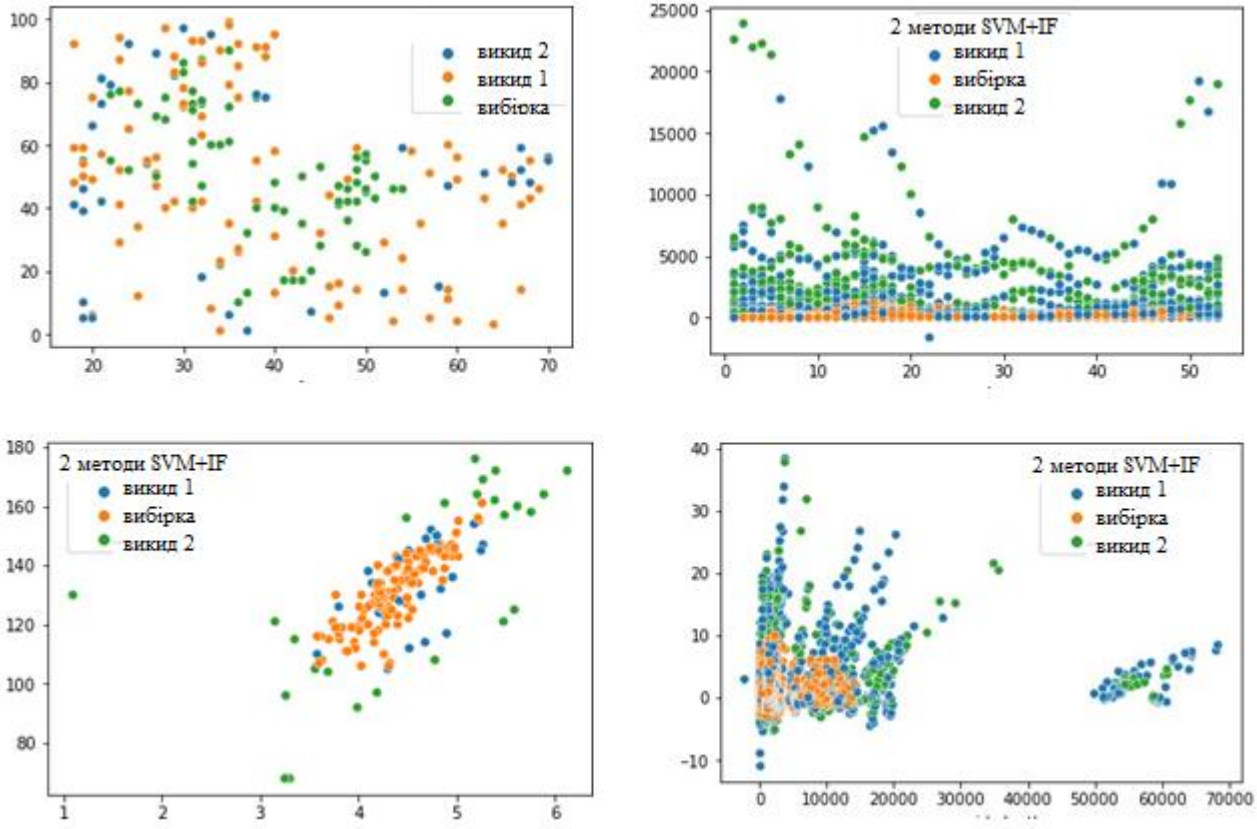


Рисунок 2.6 – Результат роботи ансамблів алгоритмів

2. One Class SVM + Elliptic Envelope

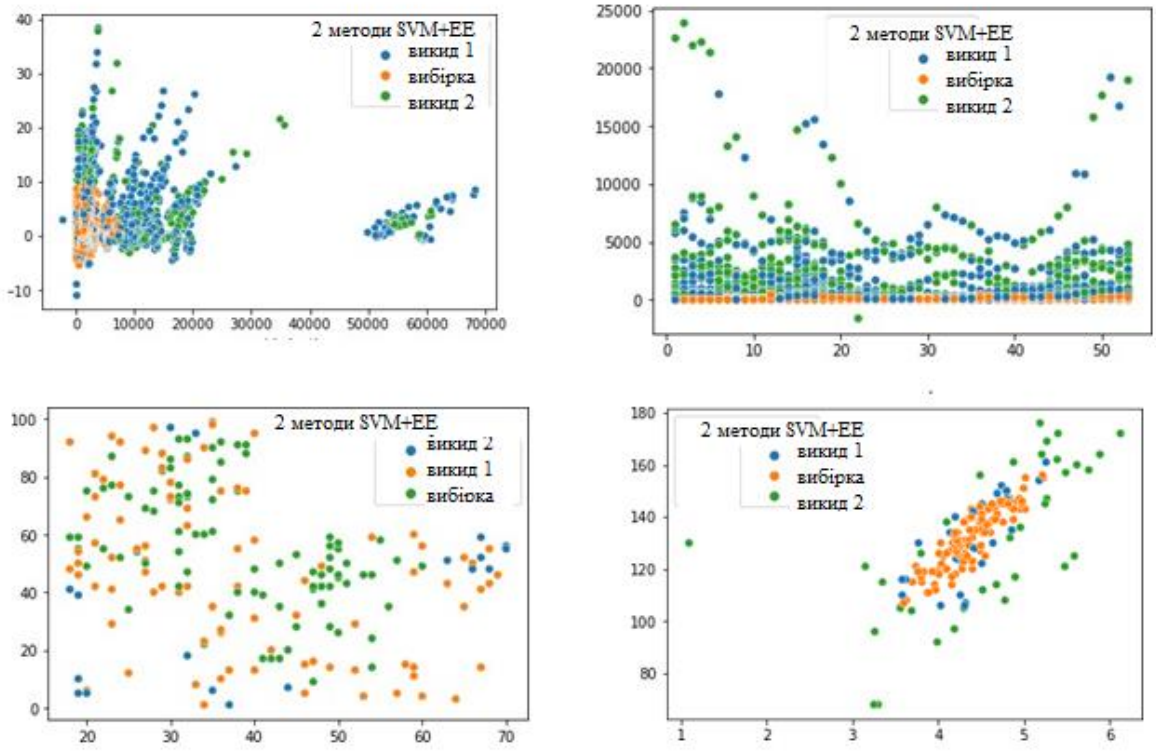


Рисунок 2.7 – Результат роботи ансамблю алгоритмів

3. Elliptic Envelope + Isolation Forest

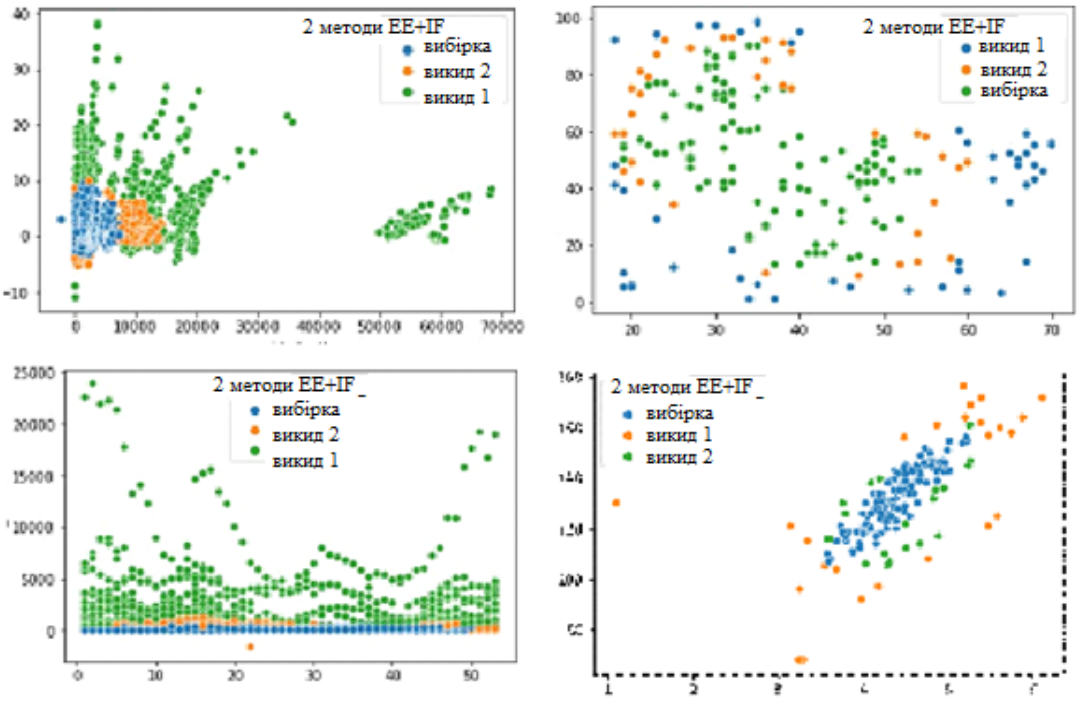


Рисунок 2.8 – Результат роботи ансамблю алгоритмів

3. Isolation Forest + One Class SVM + Elliptic Envelope

Отримані дані внаслідок спостережень наборів та ідентифіковані вищевказаним методом відображено на рисунку 2.9.

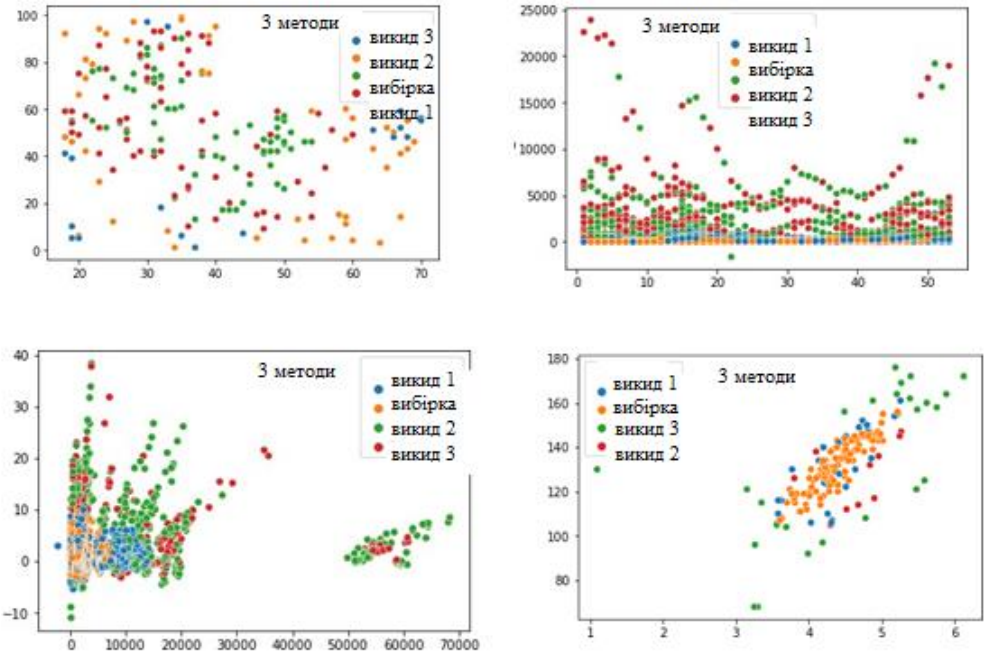


Рисунок 2.9 – Результат роботи ансамблю алгоритмів

2.5. Результати проведеного дослідження

Зі значеннями параметрів по замовчуванням були побудовані сім моделей класифікації. Результати роботи таких моделей, як Isolation Forest, Local Outlier Factor, Elliptic Envelope та їх ансамблі, а саме, значення точності їх передбачень, представлені в таблиці.

Таблиця 2.1 – Точність передбачення моделей класифікації

№	EE + SVM	SVM + IF	SVM	IF + EE	EE	IF + EE + SVM	IF
1	0,94	0,90	0,91	0,92	0,91	0,94	0,91
2	0,93	0,83	0,80	0,92	0,97	0,89	0,97
3	0,96	0,76	0,86	0,97	0,87	0,89	0,87
4	0,89	0,81	0,81	0,82	0,92	0,96	0,92
5	0,91	0,81	0,91	0,88	0,92	0,93	0,92
6	0,95	0,95	0,75	0,95	0,85	0,93	0,85
7	0,96	0,56	0,86	0,91	0,93	0,97	0,93
8	0,89	0,89	0,80	0,89	0,89	0,94	0,89
9	0,91	0,81	0,88	0,94	0,82	0,95	0,82
Середнє	**0,927**	**0,813**	**0,842**	**0,911**	**0,898**	**0,933**	**0,898**

Для забезпечення перевірки якості роботи алгоритмів використовуються фактичні дані та матриця помилок.

Матриця помилок, представляє собою таблицю з чотирма різними комбінаціями, використовується для оцінки ефективності алгоритмів. Кількість вірних і невірних прогнозів підсумовуєтьсяз фактичними значеннями як справжні та хибні діляться за класів.

Наприклад, дана вибірки $x_i (i = 1,\ldots, N, y_i)$ - Мітка класу i-го об'єкта, $y_i \{1,2,\ldots, C\}$, кожен об'єкт якої відноситься до одного із 3-ьох класів і класифікатор a , який їх передбачає.

Матрицею помилок для такого класифікатора є матриця:

$$M = \{m_{ij}\}_{i,j=0}^{C}, m_{ij} = \sum_{k=0}^{N} [[a(x_k) = j]][[y_k = i]] \quad (2.3)$$

Така матриця показує, скільки об'єктів класу *i* було розпізнано як об'єкти класу.

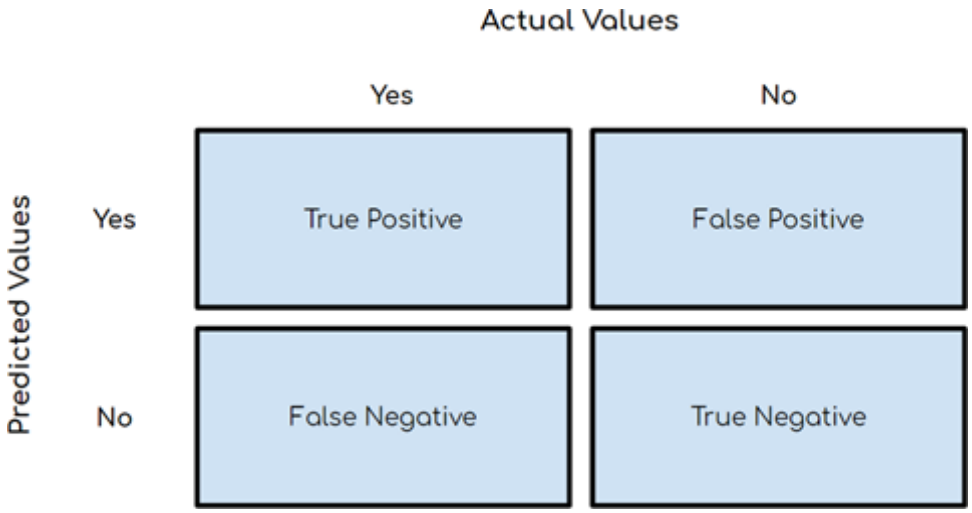


Рисунок 2.10 – Матриця помилок

Структура матриці помилок 2×2 показана на рисунку. У випадку бінарного розбиття, мітка класу «y» отримує значення $+1$ (позитивний клас) або -1 (негативний). У якостіприкладу припустімо, що було десять випадків, коли алгоритм класифікації передбачав «Так», як і фактичне значення. Тоді число десять буде у верхньому лівому кутку істинного квадранта позитивного результату. Це підводить нас до декількох ключових термінів:

1. Позитивний (P): Спостереження позитивне (наприклад, викид).
2. Негативний (N): Спостереження негативне (наприклад, це не викид).
3. Істинно-позитивний (TP): результат, при якому алгоритм правильно передбачає позитивний клас.

$$TP = \sum_{i=0}^{n} [a(x_i) = +1][y_i = +1]$$

4. Істинно-негативний (TN): результат, при якому алгоритм правильно

передбачає негативний клас.

$$TN = \sum_{i=0}^{n} [a(x_i) = -1][y_i = -1]$$

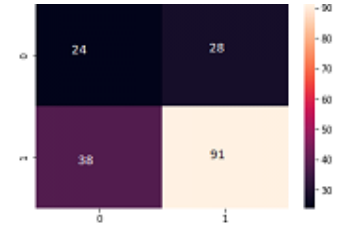
5. Невірно - позитивний (FP): також називається помилкою типу 1, результатом, коли алгоритм невірно передбачає позитивний клас, але насправді негативний.

$$FP = \sum_{i=0}^{n} [a(x_i) = +1][y_i = -1]$$

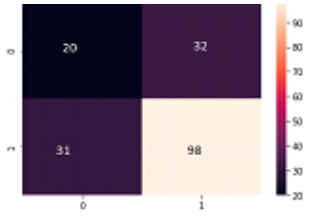
6. Невірно - негативний (FN): також називається помилкою 2-го типу, приякому алгоритм неправильно передбачає негативний клас, хоча насправді він позитивний.

$$FN = \sum_{i=0}^{n} [a(x_i) = -1][y_i = +1]$$

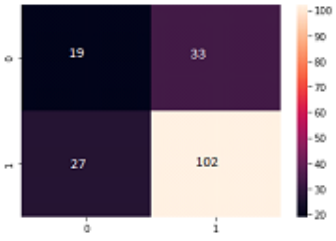
Побудуємо матрицю помилок за допомогою "sklearn.metrics.confusion_matrix" і візуалізуємо результати роботи таких алгоритмів, як «IsolationForest, Local Outlier Factor, EllipticEnvelope та їх ансамблі. Використовуючи парні дані, заповнимо матрицю помилок.



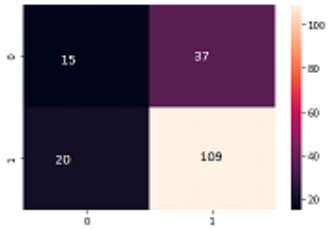
Elliptic Envelope
+ One Class SVM



One Class SVM & One Class SVM
+ Isolation Forest



Isolation Forest+
Elliptic Envelope
& Elliptic Envelope



Isolation Forest+
Elliptic Envelope+
One Class SVM

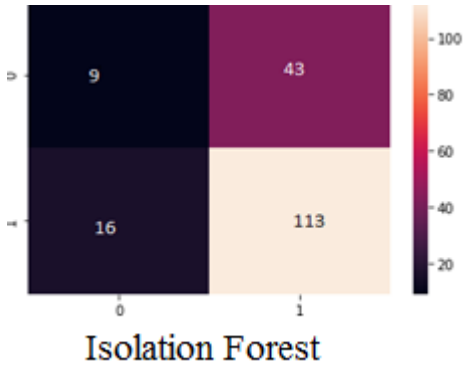


Рисунок 2.11 – Матриця помилок за допомогою "sklearn.metrics.confusion_matrix"

Виходячи з отриманих даних можна, можливо дізнатися про наших моделях кілька речей: наприклад, з ансамблю алгоритмів «EllipticEnvelope + One ClassSVM» у таблиці видно, що у 62/181- викид, тоді як на самому справі було 52/181 є викидами. Алгоритм має точність 115/181 або63%. Нам потрібно, щоб алгоритм якнайточніше виявила викид. Набагато гірше, якщо алгоритм передбачить неправдиві дані, що призведе до хибному діагнозу.

Також для оцінки якості алгоритми використовуються значення таких метрик якості як правильність (*Accuracy*), повнота (Re*call*), точність (Pr*ecision*) і *F*1, які представлені таблично.

Таблиця 2.2 – Точність передбачення алгоритмів класифікації для викидів

Алгоритм	accuracy	recall	precision	f1
IF+EE+SVM	0.69	0.29	0.29	0.34
EE + SVM	0.64	0.46	0.39	0.42
IF+EE	0.67	0.37	0.41	0.39

Відси встановлено, що максимальну середню точність має ансамбль алгоритмів «Elliptic Envelope+OneClassSVM». Значення всіх метрик досить високі, що говорить про високу якість роботи алгоритму для пошуку аномалій (викидів).

Далі проведемо оцінку якості алгоритмів і їх ансамблів на точністьвизначення вибірки по метрикам:

- правильність (accuracy)

$$Accuracy = TP + TN + HP + FN$$

- повнота (Recall)

$$\mathrm{Re}\,call = \frac{TP}{TP + FN}$$

- точність (Precision)

$$\mathrm{Pr}\,ecision = \frac{TP}{TP + FP}$$

- F1

$$F1 - score = 2\frac{\mathrm{Pr}\,ecision \cdot \mathrm{Re}\,call}{\mathrm{Pr}\,ecision + \mathrm{Re}\,call}$$

Результати оцінки якості наведено нижче.

Таблиця 2.3 – Точність передбачення алгоритмів класифікації для вибірки

Алгоритм	accuracy	recall	precision	f1
IF+EE+SVM	0.69	0.84	0.75	0.79
EE + SVM	0.64	0.71	0.76	0.73
IF+ EE	0.67	0.79	0.76	0.77

Виходячи з отриманих даних, бачимо, що найбільшу середньою точність має ансамбль трьох алгоритмів «Isolation Forest+EllipticEnvelope +OneClass SVM», так ж можна, можливо виділити ансамбль «IsolationForest + Elliptic Envelope». Значення всіх метрик достатньо високі, що каже про високому якості роботи алгоритму.

2.6. Висновок по розділу 2

Вибір алгоритму пошуку аномалій залежить в першу черга від початкового завдання, даних та доступної вихідної інформації. Як було продемонстровано в цією

роботі, перед дослідженням виникає потреба попередньою обробки даних, по крайньої мірою, в області, в якої очікуються аномальні дані. Навчена мережа не завжди здатна правильно відображати поведінка системи. Це спричинено необхідністю мати велику кількість даних для навчання на прикладах, тоді як нейронна мережа не може передбачити ситуації, які були відсутні в історичних даних

Грунтуючись на порівняльний аналіз, найбільш стабільним та загальним алгоритмом як визначення аномалій, так визначення вибірки залишається ансамбль алгоритмів "Isolation Forest+EllipticEnvelope". Оскільки акцент роботи спрямовано визначення аномалій, варто виділити ансамбль алгоритмів "EllipticEnvelope+OneClass SVM + Isolation Forest" для вирішення цього завдання, і саме він вважається найкращим.

3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Питання охорони праці людини необхідно вирішувати на всіх стадіях трудового процесу незалежно від виду професійної діяльності.

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних, шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

У даному розділі вирішується питання охорони праці та безпеки в надзвичайних ситуаціях програміста на стадії розробки ним системи автоматизованого пошуку аномалій в результатах медичних досліджень.

3.1 Охорона праці

3.1.1 Організація робочого місця.

Приміщення, в якому працює програміст, має загальну площу 20 м^2, висоту стелі 3 м. У приміщенні знаходиться 7 робочих місць з ПК. Кожне робоче місце обладнане робочім столом площею 1,2 м^2, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші. Слід відзначити, що площа одного робочого місця оператора ПК не повинна бути меншою за 6м^2, а об'єм не менший за 20м^3, тобто площі та об'єму даного приміщення не вистачає для розташування семи робочих місць операторів ПК.

Аналіз умов праці показує, що у приміщенні лабораторії на програміста можуть негативно впливати наступні фізичні та психофізіологічні фактори:

- підвищена або знижена температура повітря робочої зони;
- підвищена або знижена вологість повітря;

- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- підвищена іонізація повітря;
- підвищений рівень електромагнітних випромінювань;
- нервово-психічні перевантаження (розумова перенапруга, перенапруга аналізаторів);
- фізичні перевантаження (одноманітна поза викликає статичну втому).

3.1.2 Мікроклімат робочої зони програміста

Робота програміста за енерговитратами відноситься до категорії легких робіт Іа, Іб, тому повинні дотримуватися наступні вимоги згідно ДСН 3.3.6.042-99:

- оптимальна температура повітря – 22°С (допустима – 20-24°С);
- оптимальна відносна вологість – 40-60% (допустима – не більше 75%);
- швидкість руху повітря не більш 0,1 м/с.

Виміряні за допомогою приладів (психрометр Августа) температура та вологість у лабораторії відповідають вказаним у таблиці для теплого періоду року.

Розташовані у приміщенні 7 ПК являються джерелами тепловиділень, крім того для підтримання у приміщенні в холодний період року оптимальних параметрів мікроклімату використовуються нагріті поверхні опалювальної системи. Нормованим показником ІЧВ є гранично допустима густина потоку енергії $J_{г.д}$, Вт/м^2, яка встановлюється в залежності від площі опромінюваної поверхні тіла людини ($S_{опр}$). Нормовані рівні складають: $J_{г.д}$=35 Вт/м^2 при $S_{опр}$ > 50%; $J_{г.д}$=70 Вт/м^2 при $S_{опр}$ ~ 25-50%; $J_{г.д}$=100 Вт/м^2 при $S_{опр}$ < 25%.

3.1.3 Освітлення робочого місця

Нормованим параметром природного освітлення згідно ДБН В.2.5–28 –

2006 є коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт. Робота програміста відноситься до робіт середньої точності (IV розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5-1,0мм), для яких при використанні бокового освітлення КПО=1,5%. Для штучного освітлення нормованим параметром виступає Емін – мінімальний рівень освітленості, та Кп – коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж 20%. Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300-500 лк.

В приміщенні використовуються світильники типу ЛПО. Кожен світильник комплектується двома лампами. Тобто необхідно використовувати 6 світильників із 12 працюючими лампами в них.

3.1.4 Вплив шуму на програміста

Як було вказано вище, в кабінеті знаходиться сім робочих місць з ПК, кожне з яких устатковане монітором, вінчестером в системному блоці, трьома вентиляторами системи охолоджування ПК та клавіатурою. Крім того поряд працює периферійна техніка. Таким чином у приміщенні мають місце шуми механічного і аеродинамічного походження, широкосмугові із аперіодичним підсиленням при роботі принтерів.

3.1.5 Виробничі випромінювання

Допустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітору комп'ютера представлені в таблиці 4.1. Нормованим параметром невикористаного рентгенівського випромінювання виступає потужність експозиційної дози. На відстані 5 см від поверхні екрану монітору її рівень не повинен перевищувати 100 мкР/год. Максимальний рівень рентгенівського випромінювання на робочому місці програміста зазвичай не

перевищує 20 мкР/год.

На відстані 5-10 см від екрана і корпуса монітора рівні напруженості можуть досягати 140 В/м по електричній складовій, що значно перевищує допустимі значення.

Таким чином, у підрозділі враховано вимоги охорони праці та техніки безпеки на робочому місці програміста при розробці автоматизованої системи.

3.2 Безпека в надзвичайних ситуаціях

3.2.1 Електробезпека

Персональні комп’ютери, периферійні пристрої, інше устаткування (апарати управління, контрольно-вимірювальні прилади, світильники), електропроводи та кабелі за виконанням і ступенем захисту мають відповідати класу зони, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів [14]. Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, застосовувати негорючу ізоляцію. Лінія електромережі для живлення персональних комп’ютерів і периферійних пристроїв виконується як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники. Площа перерізу

нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту.

У приміщенні, де одночасно експлуатуються понад п’ять персональних комп’ютерів і периферійних пристроїв, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Персональні комп’ютери і периферійні пристрої повинні підключатися до електромережі тільки за допомогою справних штепсельних з’єднань і електророзеток заводського виготовлення. У штепсельних з’єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз’єднання при відключенні має бути зворотним. Не допускається підключати персональні комп’ютери та периферійні пристрої до звичайної двопровідної електромережі, в тому числі з використанням перехідних пристроїв.

Електромережі штепсельних з’єднань та електророзеток для живлення персональних комп’ютерів та периферійних пристроїв потрібно виконувати за магістральною схемою, по 3-6 з’єднань або електророзеток в одному колі. Штепсельні з’єднання та електророзетки для напруги 12В та 42В за своєю конструкцією мають відрізнятися від штепсельних з’єднань для напруги 127В та 220В. Штепсельні з’єднання та електророзетки, розраховані на напругу 12В та 42В, мають візуально (за кольором) відрізнятися від кольору штепсельних з’єднань, розрахованих на напругу 127В та 220В. Індивідуальні та групові штепсельні з’єднання та електророзетки необхідно монтувати на негорючих або важкогорючих пластинах. Електромережу штепсельних розеток для живлення

персональних комп’ютерів і периферійних пристроїв при розташуванні їх уздовж стін приміщення прокладають по підлозі поруч зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах, а також у пластикових коробах і пластмасових рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. При розміщенні в приміщенні до п’яти персональних комп’ютерів і периферійних пристроїв допускається прокладання трипровідникового захищеного проводу або кабелю в оболонці з негорючого чи важкогорючого матеріалу по периметру приміщення без металевих труб та гнучких металевих рукавів. Не допускається в одній трубі прокладати кола до 42В та вище 42В.

При організації робочих місць операторів електромережу штепсельних розеток для живлення персональних комп’ютерів, периферійних пристроїв і у центрі приміщення прокладають у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах. При цьому не допускається застосовувати провід і кабель в ізоляції з вулканізованої гуми та інші матеріали, які містять сірку.

3.2.2 Запобігання виникненню надзвичайних ситуацій.

Найбільш ефективний спосіб зменшення шкоди та збитків від надзвичайних ситуацій – запобігти їх виникненню, а в разі виникнення виконувати відповідні до даної ситуації заходи. Запобігання виникненню надзвичайних ситуацій – це підготовка та реалізація комплексу заходів, спрямованих на регулювання безпеки, проведення оцінки рівнів ризику, завчасне реагування на загрозу виникнення надзвичайної ситуації на основі даних моніторингу (спостережень), експертизи, досліджень та прогнозів щодо можливого перебігу подій з метою недопущення їх переростання у надзвичайну ситуацію або пом’якшення її можливих наслідків.

Зазначені функції запобігання надзвичайним ситуаціям техногенного і природного характеру в нашій країні покликана виконувати Єдина державна

система цивільного захисту (ЄДСЦЗ), затверджена Постановою Кабінету Міністрів України від 9 січня 2014 р. №11. ЄДСЦЗ включає в себе центральні та місцеві органи виконавчої влади, виконавчі органи рад, державні підприємства, установи та організації з відповідними силами і засобами, які здійснюють нагляд за забезпеченням техногенної та природної безпеки, організують проведення роботи із запобігання надзвичайним ситуаціям і реагування у разі їх виникнення з метою захисту населення і довкілля, зменшення матеріальних втрат. ЄДСЦЗ складається з постійно діючих функціональних та територіальних підсистем і має чотири рівні: загальнодержавний, регіональний, місцевий та об'єктовий. Кожен рівень ЄДСЦЗ має координуючі та постійні органи управління. Координуючими органами ЄДСЦЗ є:

на загально державному рівні:

- Державна комісія з питань техногенно-екологічної безпеки та надзвичайних ситуацій;

- Національна рада з питань безпечної життєдіяльності населення;

на регіональному рівні – комісії обласних державних адміністрацій з питань техногенно-екологічної безпеки та надзвичайних ситуацій;

на місцевому рівні – комісія районних державних адміністрацій і виконавчих органів рад з питань техногенно-екологічної безпеки та надзвичайних ситуацій;

на об'єктовому рівні – комісії з питань надзвичайних ситуацій об'єктів.

До систем повсякденного управління ЄДСЗР входять оснащені необхідними засобами зв'язку, оповіщення, збирання, аналізу і передачі інформації: центри управління в надзвичайних ситуаціях, оперативно-чергові служби уповноважених органів з питань надзвичайних ситуацій та цивільного захисту населення усіх рівнів; диспетчерські служби центральних та місцевих органів виконавчої влади, державних підприємств, установ та організацій.

До складу сил і засобів ЄДСЦЗ входять військові і спеціальні цивільні аварійно-рятувальні (пошуково-рятувальні) формування, які укомплектовуються з урахуванням необхідності проведення роботи в автономному режимі не менше

трьох діб і перебувають у стані постійної готовності, а також недержавні (добровільні) рятувальні формування. Залежно від масштабів і особливостей надзвичайної ситуації, що прогнозується або виникла, може існувати один із таких режимів функціонування ЄДСЦЗ: повсякденної діяльності, підвищеної готовності, діяльності у надзвичайній ситуації , діяльності у надзвичайному стані. З метою ліквідації наслідків надзвичайної ситуації у мирний час може поводитися цільова мобілізація.

Ефективність функціонування систем захисту населення і територій досягається через завчасну підготовку, оперативне реагування та ефективне управління під час надзвичайних ситуацій, своєчасне відновлення життєдіяльності населення в їх зоні.

ВИСНОВОК

Результатом виконання КРМ проаналізовано існуючі алгоритми машинного навчання і розроблено ансамблі алгоритмів для виявлення аномалій у медичних дослідженнях. Алгоритм створено для спрощення і прискорення роботи медичних працівників, а також точності визначення аномалій для постановки вірного діагнозу.

Зроблено порівнювальний аналіз точності роботи алгоритмів і ансамблів алгоритмів. В результаті роботи найкращим алгоритмом у рамках даної роботиє ансамбль трьох алгоритмів «EllipticEnvelope+OneClassSVM+IsolationForest».

Науковою новизною даної роботи можна, можливо назвати використання ансамблів алгоритмів у галузі медицини для роботи з великим масивом данних.

Розділ «Охорона праці та безпека в надзвичайних ситуаціях» висвітлює питання, щодо тематики роботи.

Враховуючи вищесказане зроблено висновок щодо повноти виконання посталих завдань.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anomaly Detection Principles and Algorithms 2017 Edition

2. Hands-On Machine Learning with scikit-learn and Scientific Python Toolkits(Release)

3. Beginning Anomaly Detection Using Python-Based Deep Learning: With Kerasand PyTorch 1st ed. 2019

4. Outlier Analysis 2nd ed. 2017 Edition

5. Outlier Detection: Techniques and Applications 1st Ed. 2019 Edition

6. Топ-5 мов для машинного навчання [Електронний ресурс]/З. Стельмах. Електрон. текстові дані. 2019. URL: https://www.itweek.ru/ai/article/detail.php

7. Kaggle [Електронний ресурс] URL:https://ua.wikipedia.org/wiki/Kaggle

8. Scikit-learn.org [Електронний ресурс] URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

9. Scikit-learn.org [Електронний ресурс] URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

10. Scikit-learn.org [Електронний ресурс] URL: https://scikit-learn.org/stable/modules/generated/sklearn.covariance.EllipticEnvelope.html

11. Історія розвитку ансамблевих методів класифікації в машинному навчанні (Робота Ю.В. Кашніцького) 2015р.

12. Ensemble methods: bagging, boosting and stacking [Електронний ресурс] URL: https://towardsdatascience.com/ensemble-methods-bagging-boosting- and-stacking-c9214a10a205.

13. Дроздов В.Я., Яворська Є.Б., Андрійчук Н.Є. Розробка програмно-апаратних засобів відбору та аналізу біосигналів Матеріали XI науково-технічної конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 13-14

грудня 2023 р.). – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2023. – 257 с.

14. Стручок В.С. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної бо та заочної (дистанційної) форм навчання / В.С.Стручок. — Тернопіль: ФОП Паляниця В. А., 2022. — 156 с.

ДОДАТКИ

ДОДАТОК А

Object and research методів

1.1. Machine learning techniques

Перші методи came від pur statistics в '50s. Вони solved formal math tasks — searching for patterns in numbers, evaluating the proximity of data points, and calculating vectors' directions.

Лише результат машинного навчання є наслідком результатів, що базуються на incoming data. Великий різновид в те, що ви маєте, easier it is to find relevant patterns and predict the result. Therefore, we need three components to teach the machine (Figure 1):

1. Data

There are two main ways to get the data - manual and автоматичний. Manually collected data contains far fewer errors but takes more time to collect - that makesit more expensive in general. Automatic approach is cheaper - you're gathering everything you can find and hope for the best.

2. Features

Такожвідомо як параметри або параметри. Those could be car mileage, user's gender, stock price, word frequency в тексті. В інших словах, вони є factorsfor a machine to look at.

3. Algorithms

Most obvious part. Any problem can ru solved Різноманітно. The метод you choose affects the precision, performance, і size of the final model. There is one Важлива nuance though: if the data is crappy, even the best algorithm won't help. Sometimes it's referred as "garbage in - garbage out". So don't pay too much attention to the percentage of accuracy, try to acquire more data Перший.

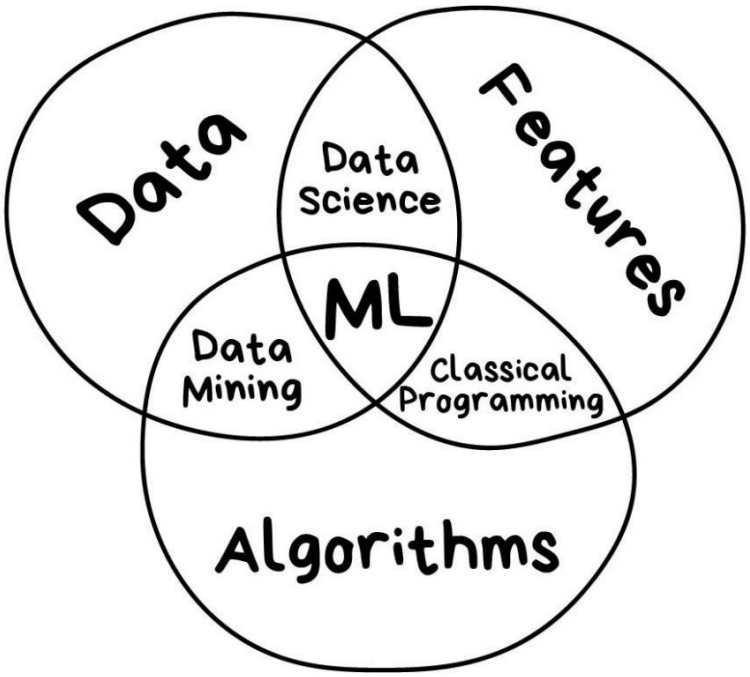


Figure 1 Components to teach the machine

1.1.1. Supervised Learning

Today used for:

- Spam filtering
- Language detection
- Fraud detection
- Stock price forecasts
- Medical diagnosis
- Any number-time correlations

How it works: Цей algoritm consist of a target / outcome variable (or dependent variable) which is to ru predicted from a given set of predictors (Independent variables). За допомогою цього набору variables, we generate a function that map inputs to desired outputs. Training process continues until the model achieves a desired level of accuracy on the training data. Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

1.1.2. Unsupervised Learning

Nowadays used:

- To analyze and label new data

- To detect abnormal behavior
- To analyze web surfing patterns
- Topic modeling and similar document search
- Fake image analysis
- Risk management

How it works: У цій algorithm, ми не маємо будь-якого target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means.

1.1.3. Reinforcement Learning

Nowadays used for:

- Self-driving cars
- Robot vacuums
- Games
- Automating trading
- Enterprise resource management

How it works: Використовуючи цей algoritm, машина трейдується до конкретногоdecisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process.

1.1.4. Deep Learning

Used today for:

- Replacement of all algorithms above
- Object identification on photos and videos
- Speech recognition and synthesis
- Image processing, style transfer
- Machine translation

Deep Learning is a modern method of building, training, and using neural

networks. Основним чином, це нова архітектура. Nowadays in practice, no one separatesdeep learning від "ordinary networks". We even use the same libraries for їм. The general rule is to compare things on the same рівень.

1.1.5. Reinforcement Learning

Nowadays used for:

- Self-driving cars
- Robot vacuums
- Games
- Automating trading
- Enterprise resource management

Reinforcement learning is used in cases when your problem is not related todata at all, but you have an environment to live in. Like a video game world or a cityfor self-driving car.

Knowledge of all the road rules in the world will not Teach the autopilot howto drive on the roads. Regardless of how much data we collect, we still can't foreseeall the possible situations. This is why its goal is to minimize error, not to predict allthe moves. Surviving in an environment is a core idea of reinforcement learning.

1.1.6. Ensemble Methods

Nowadays is used for:

- Everything that fits classical algorithm approaches (but works better)
- Search systems
- Комп'ютер vision
- Object detection

Ensembles and neural networks є двома основними бойовиками, які пливуть нами. singularity. Today they are producing the most accurate results and are widely usedin production.

Despite all the effectiveness the idea behind these is overly simple. If you таке a bunch of inefficient algorithms and force them to correct each іншimistakes, the overall

quality of a system will ru higher than even the best individualalgorithms.

Ви будете бути зарезервовані результати, якщо ви будете most untable algorithms, що are predicting completely different results on small noise in input data. Like Regression and Decision Trees. These algorithms are so sensitive to even a single outlier in input data to have models go mad.

A visualization of the machine learning методів described above is shownin Figure 2.

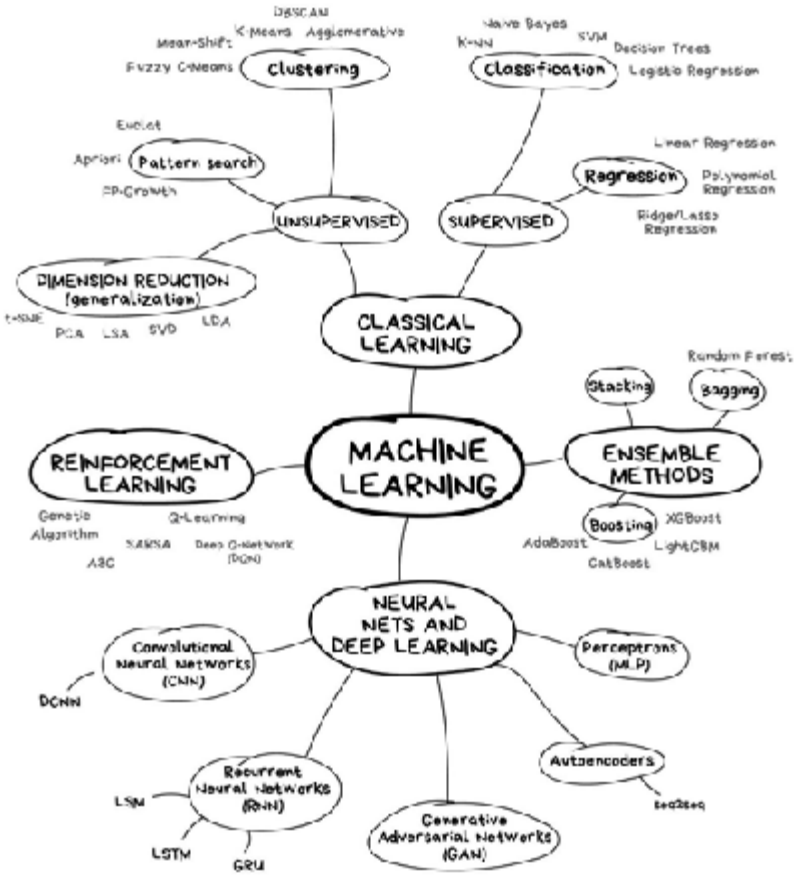


Figure 2 Machine learning методів

ДОДАТОК Б

Програмний код

```
import numpy as np
import matplotlib.pyplot as pltimport pandas as pd
from sklearn import svm
з sklearn.covariance import EllipticEnvelopeimport seaborn as
sns
from matplotlib import rc
df0=pd.read_csv('all_weekly_excess_deaths.csv')
df=df0[['week','covid_deaths']]
x=df.values
від sklearn.ensemble import IsolationForest
forest=IsolationForest(random_state=0) forest.fit(x)
y = forest.predict(x) print(y) outliers_fraction = 0.25
svmoutl=svm.OneClassSVM(nu=0.95 * outliers_fraction +
0.05,kernel="rbf", gamma=0.1)
svmoutl.fit(x)
svmy = svmoutl.predict(x)print(svmy)
el=EllipticEnvelope(contamination=outliers_fraction)el.fit(x)
ely = el.predict(x)print(ely)
from sklearn.cluster import DBSCAN
dbs = DBSCAN(eps=0.3, min_samples=10).fit(x)
df['outlierForest']=y
df['outliersvm']=svmydf['outlierel']=ely
df.to_excel('out.xlsx') midwest=pd.read_excel('out.xlsx')
sns.scatterplot(x="week", y="covid_deaths",
hue="outlierForest",data = df)
sns.scatterplot(x="week", y="covid_deaths", hue="outlierel",
data = df)
df['pic1']=df['outliersvm']+df['outlierel']
df.to_excel('out22.xlsx') pic=pd.read_excel('out22.xlsx')
df['2 методу SVM+EE']=''mask1=df['pic1']==-2
```

```
mask2=df['pic1']==0 mask3=df['pic1']==2
    df['2 методу SVM+EE'][mask1]='викид2' df['2 методу SVM+EE'][mask2]='викид1'df['2 методу SVM+EE'][mask3]='вибірка'
    sns.scatterplot(data=df, x="week", y="covid_deaths", hue="2 методу SVM+EE")
    df['pic10']=df['outliersvm']+df['outlierForest'] df.to_excel('out44.xlsx') pic=pd.read_excel('out44.xlsx')
    df['2 методу SVM+IF']=''mask1=df['pic10']==-2 mask2=df['pic10']==0 mask3=df['pic10']==2
    df['2 методу SVM+IF'][mask1]='викид2' df['2 методу SVM+IF'][mask2]='викид1'df['2 методу SVM+IF'][mask3]='вибірка'
    sns.scatterplot(data=df, x="week", y="covid_deaths", hue="2 методу SVM+IF")
    df['pic110']=df['outlierel']+df['outlierForest'] df.to_excel('out410.xlsx') pic=pd.read_excel('out410.xlsx')
```

```
df['2 методу EE+IF']=''mask1=df['pic110']==-2
mask2=df['pic110']==0 mask3=df['pic110']==2
df['2 методу EE+IF'][mask1]='викид1' df['2 методу
EE+IF'][mask2]='викид2'df['2 методу EE+IF'][mask3]='вибірка'
sns.scatterplot(data=df, x="week", y="covid_deaths", hue="2
методу EE+IF")
df['pic100']=df['outlierel']+df['outlierForest']+df['outliersv
m'
]
df.to_excel('out400.xlsx') pic=pd.read_excel('out400.xlsx')
df['3 методу']='' mask1=df['pic100']==-3 mask2=df['pic100']==-1
mask3=df['pic100']==3 mask4=df['pic100']==1
df['3 методу'][mask1]='викид3' df['3 методу'][mask2]='викид2'
df['3 методу'][mask3]='вибірка'df['3 методу'][mask4]='викид1'
sns.scatterplot(data=df, x="week", y="covid_deaths", hue="3
методу")
# Confusion Matrix
від sklearn.metrics import confusion_matrixconfusion_matrix(y,
svmy)
a=[[ 350, 349], [732, 1764]]
df=pd.DataFrame(a, range(2), range(2))sns.heatmap(df)
plt.show()
from sklearn.metrics import accuracy_score
accuracy_score(ely, y) confusion_matrix(ely, svmy) b=[[ 414,
385], [ 668, 1728]]
df=pd.DataFrame(a, range(2), range(2))sns.heatmap(df)
plt.show() confusion_matrix(ely, y) c=[[ 596, 203], [103,
2293]]
df2=pd.DataFrame(c, range(2), range(2))sns.heatmap(df2)
plt.show()# Accuracy
від sklearn.metrics import accuracy_scoreaccuracy_score(ely, y)
# Recall
від sklearn.metrics import recall_scorerecall_score(ely, y,
average=None)
# Precision
```

```
    від sklearn.metrics import precision_scoreprecision_score(ely,
y, average=None) #F1score
    від sklearn.metrics import f1_scoref1_score(ely, y,
average=None) #Classification_report
    від sklearn.metrics import classification_report
print(classification_report(ely, y))
```

ДОДАТОК В

Тези конференції

УДК 303.01:303.447: 612.17

Дроздов В.Я., Яворська Є.Б., к.т.н., доц., Андрійчук Н.Є.
Тернопільський національний технічний університет імені Івана Пулюя, Україна
Відокремлений структурний підрозділ «Тернопільський фаховий коледж Тернопільського національного технічного університету ім. І.Пулюя», Україна

РОЗРОБКА ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ ВІДБОРУ ТА АНАЛІЗУ БІОСИГНАЛІВ

V. Drozdov, E. Yavorska, Ph.D., Assoc. Prof , N. Andriychuk,

DEVELOPMENT OF SOFTWARE AND HARDWARE TOOLS FOR SELECTING AND ANALYZING BIOSIGNALS

Цифрова обробка сигналів відноситься до числа областей інженерної діяльності, які найбільш динамічно розвиваються. Медицина, системи сотового зв'язку, телекомунікації, internet-технології, обробка звуку та зображень, навігація – це далеко неповний перелік прикладів, в яких активно використовуються сигнальні процесори або процесори цифрової обробки сигналів (DSP – від англ. digital signal processors). DSP є різновидом мікропроцесорів, які розраховані на обробку в реальному часі цифрових потоків даних, утворених в результаті оцифровування аналогових сигналів. При наявності архітектури, яка пристосована для цифрової обробки сигналів, DSP дозволяють створювати ефективні системи обробки та передачі сигналів в реальному часі. Застосування сигнальних процесорів для цифрової обробки біосигналів потребує розробку ефективних алгоритмів та програм. Виконання даної задачі також пов'язано з вибором типу сигнального процесора згідно наступних параметрів: - формат даних та розрядність; - швидкість; - організація пам'яті; - енергоспоживання; - зручність розробки програм. Алгоритм програми для сигнального процесора складається з декількох етапів. Перший етап передбачає визначення періоду кореляції періодично-нестаціонарного біосигналу. На другому етапі здійснюється вибір методу обробки сигналу: компонентний, когерентний (синфазний) або фільтровий. На третьому етапі виконується оцінювання спектру потужності сигналу. На четвертому етапі проводиться вибір відліків через період кореляції, який визначається на першому етапі. На п'ятому етапі – параметрична коваріація та швидке перетворення Фур'є. Шостий етап включає оцінку спектру потужності сигналу.

Результати виконання програми виводяться на дисплей для подальшої оцінки. В якості сигнального процесора вибрано процесор фірми Texas Instruments моделі TMS320C600. Для розробки програмного коду використано програмне середовище Matlab 7.0, в якому є можливість компіляції даних на мову асемблера. Для тестування та відлагодження результуючої програми використано симулятор сигнального процесора - програмне середовище Code Composer Studio фірми Texas Instruments.

Література

1. Р.А. Ткачук, Г.Б. Цуприк, і Б.І. Яворський, "Підвищення інформативності та швидкодії біотехнічних систем", Опт-ел. інф-енерг. техн., вип. 24, вип. 2, с. 81–85, Жов 2013.

2. Цуприк Г.Б.. Верифікація методу оцінювання результату активного інформаційного дослідження біооб'єкту. Матеріали XVIII наукової конференції ТНТУ ім. І. Пулюя, 2014, 107-108.

3. Яворська Є.Б. Математичні моделі та методи опрацювання ритмокардіосигналів для визначення характеристик серцевої ритміки з прогнозованою вірогідністю. Тернопіль, ТНТУ, 2009.