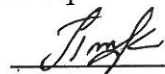


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

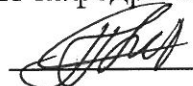
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Хмарне рішення задачі класифікації
медичних даних на основі Конструктора
машинного навчання Azure»

Виконала студентка 4-го курсу, групи КІ-3
спеціальності 123 «Комп'ютерна інженерія»

 М.О. Побережна

Керівник к.т.н., доцент, доцент кафедри КІ та КБ

 І.В. Пулеко

Рецензент к.т.н., асистент кафедри КН

 В.В. Галицький

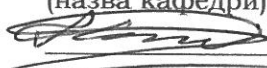
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
(назва прописними)

КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА КІБЕРБЕЗПЕКИ
(назва прописними)

СПЕЦІАЛЬНІСТЬ 123 «КОМП'ЮТЕРНА ІНТЕЖЕНЕРІЯ»
(назва прописними)

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної
інженерії та кібербезпеки
(назва кафедри)

 Ефіменко А.А.

« 12 » квітня 2021 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу

Студента Побережної Марії Олександрівни

Тема роботи: Хмарне рішення задачі класифікації медичних даних на основі
конструктора машинного навчання Azure

Затверджена Наказом університету від « 12 » квітня 2021 р. № 133/с

Термін здачі студентом закінченої роботи

Вихідні дані роботи (зазначається предмет і об'єкт дослідження)

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що
їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Пулеко І.В.	12.04.21	16.06.21
2	Пулеко І.В.	12.04.21	16.06.21
3	Пулеко І.В.	12.04.21	16.06.21

Керівник



(підпис)

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання	12.04.21	виконано
2	Розробка плану роботи	15.04.21 – 20.04.21	виконано
3	Розробка розгорнутого плану	21.04.21 – 29.04.21	виконано
4	Подача чернетки керівнику для перевірки	20.05.21	виконано
5	Доопрацювання роботи та перевірка керівником	21.05.21 – 3.06.21	виконано
6	Затвердження графічних матеріалів у керівника	8.06.21	виконано
7	Проведення попереднього захисту	14.06.21	виконано
8	Отримання відгуку керівника	15.06.21	виконано
9	Допуск завідувача кафедри	15.06.21	виконано
10	Подання роботи на рецензування	16.06.21	виконано

Студент 
(підпис)

Керівник 
(підпис)

РЕФЕРАТ

Випускна робота бакалавра присвячена дослідженню методів класифікації та пошуку оптимальної інформаційної технології на базі цих методів, для прогнозування наявності захворювань у результаті аналізу медичних даних.

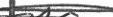
Проведено обґрунтування актуальності теми, аналіз методів машинного навчання та аналіз існуючих алгоритмів класифікації, а також досліджено аналітичні можливості обраного хмарного сердовища Azure.

Метою роботи є розробка хмарної інформаційної технології на базі алгоритмів класифікації, для прогнозування наявності захворювань у результаті аналізу даних пацієнтів.

Об'єктом дослідження є процес побудови моделі класифікації для діагностування хвороб та аналіз її можливостей з використанням Студії машинного навчання Azure. Предметом дослідження є технологія рішення задач класифікації для діагностування хвороб на основі медичних даних пацієнта з використанням Конструктора машинного навчання Azure.

Пояснювальна записка до випускної бакалаврської роботи містить 78 сторінок, 52 рисунки, 1 додаток та 36 літературних джерел.

КЛЮЧОВІ СЛОВА: аналіз даних, модель класифікації, машинне навчання, штучний інтелект, хмарні технології, дерево прийняття рішень, медицина.

					КІ.КР.Б – 123 – 21 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Хмарне рішення задачі класифікації медичних даних на основі Конструктора машинного навчання Azure Пояснювальна записка	Літ.	Арк.	Аркушів
Розроб.		Побережна М.О.		16.06.21				
Керівник		Пулеко І.В.		16.06.21			4	78
Реценз		Галицький В.В.		16.06.21		Житомирська політехніка, група КІ-3		
Н. контр.								
Зав. каф.		Єфіменко А.А.		16.06.21				

ABSTRACT

Bachelor's qualification work is devoted to the research of methods and defining optimal tech information for researched methods, therefore is used to predict potential deceases based on medical analysis.

The topic relevance was justified, both machine learning and algorithm classification methods analyzed, Azure cloud studio researched.

The main aim of this work is to create a cloud computing technology based on classification algorithms, that will predict deceases in effect or potential ones, as a result of patient's medical data.

The main object of this research is the creation process of a classification model for illness prediction and treatment, and analyzing its capabilities via Azure machine learning studio.

The subject of this research is case solving technology for medical forecasting made via Azure machine learning designer.

The explanatory note to bachelor's work contains 78 pages, 52 pictures, 1 addition and 36 literary sources.

KEYWORDS: data analysis, classification model, machine learning, artificial intelligence, cloud computing, decision tree, medicine.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ РОБОТИ.....	10
1.1 Обґрунтування актуальності теми роботи	10
1.2 Аналіз методів машинного навчання.....	14
1.3 Аналіз існуючих методів класифікації	21
1.4 Аналіз аналітичних можливостей хмарного середовища Azure.....	31
1.5 Постановка завдання на розробку	35
РОЗДІЛ 2 РОЗРОБКА ХМАРНОГО РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ МЕДИЧНИХ ДАНИХ НА ОСНОВІ КОНСТРУКТОРА МАШИННОГО НАВЧАННЯ AZURE.....	37
2.1 Розробка архітектури рішення з використанням студії машинного навчання Azure	37
2.2 Аналіз вхідних даних та їх попередня обробка	42
2.3 Розробка та навчання моделі для класифікації.....	48
2.4 Оцінка та аналіз якості моделі.....	53
РОЗДІЛ 3 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ	61
3.1 Розгортання моделі у вигляді кінцевого продукту	61
3.2 Дослідження та порівняння алгоритмів прогнозування студії машинного навчання Azure	65
3.3 Рекомендації по застосуванню розробленої інформаційної технології.....	69
ВИСНОВКИ.....	71
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	73
ДОДАТКИ.....	77

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Машинне навчання – МН

Штучний інтелект – ШІ

Метод опорних векторів – МОВ

Операційна система – ОС

Open Neural Network Exchange – ONNX

Software Development Kit – SDK

Machine Learning Ops – MLOps

Global Burden of Disease – GBD

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. У нинішню епоху цифрових технологій з'явилася велика кількість даних, таких як дані Інтернету речей (IoT), дані кібербезпеки, мобільні дані, бізнес-дані, дані соціальних мереж, дані про здоров'я і т. д. Для інтелектуального аналізу цих даних і розробки відповідних інтелектуальних і автоматизованих додатків ключем є знання штучного інтелекту (ШІ), зокрема машинного навчання (МН). Машинне навчання засноване на ідеї, що системи можуть навчатись, виявляти закономірності і приймати рішення з мінімальним втручанням людини, що пришвидшує вирішення поставлених завдань. Актуальність теми обумовлена впровадженням інтелектуальних технологій в такі сфери людства, як соціальні мережі, фінансові послуги, медицина, транспорт, уряд, торгівля, пошук корисних копалин, тощо. МН відіграє чималу роль у галузі охорони здоров'я, включаючи розробку нових медичних процедур, обробку даних і записів пацієнтів, прогнозування та лікування хронічних захворювань. Це дає змогу більш точно ставити діагнози і зменшити кількість летальних результатів.

Досить часто МН використовують для побудови моделі аналізу та прогнозування наявності хвороби пацієнта, застосовуючи для цього алгоритми класифікації. Особливо доступним це стає завдяки застосуванню хмарних сервісів. Тому, **тема роботи**, що направлена на розробку хмарного рішення задачі класифікації медичних даних на основі конструктора машинного навчання Azure є **актуальною**.

Мета і завдання роботи. Метою роботи є розробка хмарної інформаційної технології на базі алгоритмів класифікації, для прогнозування наявності захворювань у результаті аналізу даних пацієнтів.

Досягнення поставленої мети передбачає розв'язання таких завдань:

- проведення аналізу існуючих технологій у різних сферах діяльності людини, зокрема в медицині;
- аналізу існуючих методів та можливостей інтелектуального аналізу даних з використанням хмарних сервісів Microsoft Azure в медицині;

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

- розробки моделі класифікації на основі медичних даних та прогнозування класу з використанням студії машинного навчання Azure Machine;
- побудови моделі прогнозування класу, навчання та оцінка якості для знаходження кращої моделі прогнозування;
- перевірки та тестування працездатності розробленої технології з використанням даних серцевих захворювань.

Об’єктом дослідження є процес побудови моделі класифікації для діагностування хвороб та аналіз її можливостей з використанням Студії машинного навчання Azure.

Предметом дослідження є технологія рішення задач класифікації для діагностування хвороб на основі медичних даних пацієнта з використанням Конструктора машинного навчання Azure.

Методи дослідження. В кваліфікаційній роботі було використано такі методи: двохкласової класифікації, візуального програмування та математичного моделювання, веб-програмування.

Практичне значення роботи полягає у такому:

- розроблена інформаційна технологія дозволяє якісно прогнозувати наявність хвороби з використанням Студії машинного навчання Azure, що допомагає в роботі лікарям;
- розроблені пропозиції використання інформаційної технології в медицині, зокрема в сфері діагностики та прогнозування серцевих захворювань.

Апробація результатів роботи. Матеріали кваліфікаційної роботи доповідалися на XII Міжнародній науково-технічній конференції: “Інформаційно-комп’ютерні технології – 2021 (ІКТ-2021)” 01-03 квітня 2021 р [36].

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ АКТУАЛЬНОСТІ ТЕМИ РОБОТИ

1.1 Обґрунтування актуальності теми роботи

Машинне навчання (МН) - одна з сучасних інновацій, яка допомогла людині поліпшити не тільки багато виробничих і професійних процесів, але і поліпшити повсякденне життя. Але що таке машинне навчання? Це підмножина штучного інтелекту, яка фокусується на використанні статистичних методів для створення інтелектуальних комп'ютерних систем, щоб вчитися на доступних йому базах даних[3]. Простіше кажучи, машинне навчання дозволяє користувачеві завантажувати в комп'ютерний алгоритм величезну кількість даних і змушувати комп'ютер аналізувати і приймати рекомендації та рішення, засновані тільки на вхідних даних. Якщо будуть виявлені які-небудь виправлення, алгоритм може включити цю інформацію, щоб поліпшити своє прийняття рішень в майбутньому.

В останні роки доступність великих наборів даних у поєднанні з вдосконаленням алгоритмів та експоненціальним зростанням обчислювальної потужності призвела до безпрецедентного сплеску інтересу до теми машинного навчання [4]. В наш час алгоритми машинного навчання успішно застосовуються для задач класифікації, регресії, кластеризації або зменшення розмірності великих наборів, особливо вхідних даних великих розмірів. Насправді, машинне навчання перевершило людські здібності у багатьох сферах (наприклад, гра в Го, самокеровані машини, класифікація зображень тощо). Як результат, величезні частини нашого повсякденного життя, наприклад, розпізнавання зображень та мови, веб-пошук, виявлення шахрайства, фільтрація електронної пошти або спаму, кредитні оцінки та багато іншого забезпечуються алгоритмами машинного навчання.

Щоб зрозуміти ще більшу актуальність та важливість машинного навчання, розглянемо декілька сфер з реального життя, де воно застосовується. А також ознайомимось з певними деталями роботи.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

Платформи соціальних мереж використовують машинне навчання в інтересах користувачів, наприклад, персоналізують новинну стрічку для кращого розміщення доцільної реклами та багато інших часто задіяних функцій, які використовуються без найменшого уявлення про те, що ці вони є додатками машинного навчання. Ось деякі з цих функцій[5]:

- «Люди, яких ви можете знати»: в Facebook користувач завжди бачить цю функцію «Люди, яких ви можете знати». Технологія машинного навчання допомагає додаткам Facebook відслідковувати друзів, з якими зв'язуються, профілі, які дуже часто відвідуються, особливі інтереси, робоче місце користувача або загальну групу і т. д. На основі безперервного навчання, Facebook пропонує своїм користувачам подружитися з «людьми, яких ви можете знати».
- Розпізнавання осіб: як тільки фотографія з другом завантажується користувачем, Facebook відразу ж розпізнає цього друга. Прогнози і унікальні особливості фотографії відстежуються в Facebook, а потім результати зіставляються з людьми зі списку друзів користувача.

Банки та інші підприємства у фінансовій галузі використовують технологію машинного навчання для двох основних цілей: для виявлення важливих аналітичних даних та запобігання шахрайству. Ці ідеї можуть визначити інвестиційні можливості або допомогти інвесторам дізнатися, коли краще торгувати. Інтелектуальний аналіз даних також дозволяє виявляти клієнтів з профілями високого ризику або використовувати кібернадзор для виявлення попереджувальних ознак шахрайства.

Веб-сайти, що рекомендують товари, які можуть вам сподобатися на основі попередніх покупок, використовують машинне навчання для аналізу історії покупок. Рітейлери покладаються на машинне навчання для збору даних, їх аналізу та використання для персоналізації покупок, реалізації маркетингової кампанії, оптимізації цін, планування поставок товарів і отримання інформації про клієнтів.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

Пошук нових джерел енергії. Аналіз мінералів в землі. Прогнозування відмови датчика нафтопереробного заводу. Оптимізація розподілу масла для підвищення його ефективності і рентабельності [6]. Кількість варіантів використання машинного навчання в цій галузі величезна - і продовжує зростати.

Можна навести безліч прикладів використання машинного навчання у різних галузях людства, але ознайомимось ще з одним важливим і перспективним застосуванням методів МН це область охорони здоров'я, починаючи від діагностики і прогнозу до розробки ліків.

Дві сфери, які можуть отримати користь від застосування методів МН у медичній галузі, - це діагностика та прогнозування результатів. Це включає можливість виявлення високого ризику для надзвичайних ситуацій, таких як рецидив або перехід в інший стан захворювання[7]. Нещодавно алгоритми МН були успішно застосовані для класифікації раку шкіри з використанням зображень[8], які можна порівняти з точністю отриманою від кваліфікованого дерматолога та прогнозування переходу від переддіабету до діабету другого типу за допомогою регулярно збираних даних електронних медичних карт.

Постійно зростаюче число додатків машинного навчання в сфері медицини, дозволяє нам зазирнути в майбутнє, в якому дані, аналіз та інновації працюють рука об руку, щоб допомогти великій кількості пацієнтів, навіть незадіюючи їх. Незабаром, досить часто можна буде знайти додатки на основі машинного навчання, впровадженні в дані про пацієнтів в режимі реального часу та доступні з різних систем охорони здоров'я в багатьох країнах, що підвищить ефективність нових варіантів лікування, які раніше були недоступні.

Розглянемо декілька додатків машинного навчання в охороні здоров'я, а також познайомимось з деякими компаніями, які використовують його можливості на користь пацієнтів і постачальників медичних послуг[9].

1. KenSci(Сіетл, Вашингтон) використовує машинне навчання для прогнозування захворювань та їх лікування, щоб допомогти лікарям втручатися раніше, прогнозувати ризик для здоров'я населення шляхом виявлення закономірностей

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

та маркерів високого ризику та моделювання прогресуючого захворювання тощо.

2. Сіох Health (Альфаретта, Грузія) використовує машинне навчання для поліпшення «управління та обмін медичною інформацією» з метою модернізації робочих процесів, полегшення доступу до клінічних даних і підвищення точності і потоку медичної інформації.

3. Технологія PathAI (Кембридж, Массачусетс) використовує машинне навчання, щоб допомогти патологоанатам швидше і точніше ставити діагнози, а також виявляти пацієнтів, яким можуть бути корисні нові види лікування або терапії.

4. Компанія Quantitative Insights (Чикаго, Іллінойс) прагне підвищити швидкість і точність діагностики раку грудей за допомогою своєї комп'ютерної станції МРТ молочної залози Quantx. Мета: кращі результати для пацієнтів за рахунок більш точної діагностики рентгенологами.

5. Машинне навчання і наука про дані в поєднанні з передовими лабораторними технологіями допомагають недавньому стартапу Insitro (Сан-Франциско, Каліфорнія) розробляти ліки з метою швидшого лікування пацієнтів при менших витратах.

6. Компанія Prognos (Нью-Йорк, Нью-Йорк) стверджує, що її реєстр містить 19 мільярдів записів про 185 мільйонів пацієнтів. За допомогою машинного навчання платформа штучного інтелекту Prognos полегшує раннє виявлення захворювань, визначає потреби в терапії, виділяє можливості для клінічних випробувань, відзначає прогалини в лікуванні та інші фактори для ряду захворювань.

7. Платформа Berg Interrogative Biology (Фрамінгем, Массачусетс) на базі штучного інтелекту використовує машинне навчання для картографування захворювань та лікування онкології, неврології та інших захворювань. Використовуючи біологію та дані, орієнтовані на пацієнтів, компанія дозволяє постачальникам медичних послуг застосовувати більш прогнозуючий підхід, а не покладатися на метод спроб і помилок.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

Серцево-судинні захворювання, переважно ішемічна хвороба серця та інсульт, є основними причинами смертності й одними з основних факторів інвалідності в усьому світі. Такі висновки отримані з дослідження Глобального тягаря хвороб (GBD – Global Burden of Disease) за 2019 рік [10].

В Україні серцево-судинні захворювання є головною причиною смертності населення. За цим показником наша країна лишається одним зі світових лідерів.

Тому тема роботи, що направлена на застосування методів класифікації для виявлення хвороби серця на основі медичних даних є особливо актуальною і важливою для нашої країни.

1.2 Аналіз методів машинного навчання

Методи машинного навчання використовуються для того, щоб змусити електронно-обчислювальну машину навчатись використовуючи дані. З них найбільш використовуваними є контрольоване навчання (навчання з вчителем) та неконтрольоване навчання (навчання без вчителя), які далі підрозділяються на такі методи, як класифікація, регресія та кластеризація (рис. 1.1).

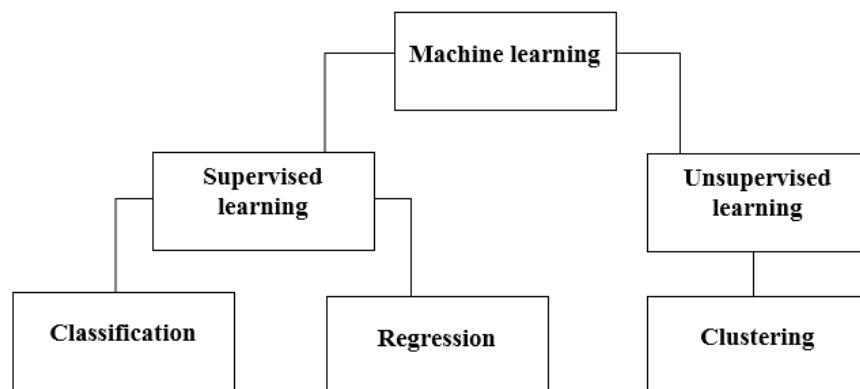


Рисунок 1.1 – Методи машинного навчання

Розглянемо детальніше кожен метод.

Алгоритми контрольованого навчання навчаються з використанням прикладів з міткою, таких як вхідні дані для яких відомий бажаний результат. Наприклад, частина обладнання може мати точки даних, помічені як «F» (збій) або «R»

(запуски). Алгоритм навчання отримує набір вхідних даних разом з відповідними правильними вихідними даними, і алгоритм навчається, порівнюючи свій фактичний вихід з правильними вихідними даними, щоб знайти помилки. Потім він відповідним чином модифікує модель. За допомогою таких методів, як класифікація, регресія, прогнозування та підвищення градієнта, контрольоване навчання використовує шаблони для прогнозування значень мітки для додаткових даних без міток. Навчання з вчителем зазвичай використовується в додатках, де історичні дані пророкують ймовірні майбутні події. Наприклад, можна передбачити, коли транзакції за допомогою кредитної картки можуть бути шахрайськими, або якій страховій компанії клієнт може подати позов[12].

Алгоритми регресії і класифікації - це алгоритми контрольованого навчання. Обидва алгоритми використовуються для прогнозування в машинному навчанні та роботи з поміченими наборами даних. Але різниця між ними полягає в тому, як вони використовуються для вирішення різних завдань машинного навчання.

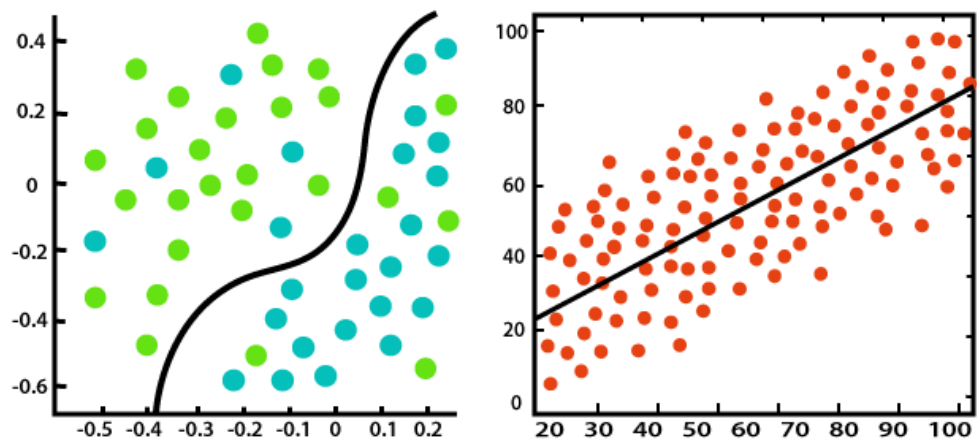
Основна відмінність між алгоритмами регресії та класифікації полягає в тому, що алгоритми регресії використовуються для прогнозування цифрових значень, таких як ціна, зарплата, вік тощо, а алгоритми класифікації використовуються для прогнозування або класифікації дискретних значень, таких як чоловічий чи жіночий, істинний чи хибний, спам чи не спам тощо.

Різницю між алгоритмами можемо наглядно побачити використовуючи діаграму, яка показана на рисунку 1.2

Класифікація - це процес пошуку функції, яка допомагає розділити набір даних на класи на основі різних параметрів (ознак). У класифікації комп'ютерна програма навчається на базі даних навчального набору, і на основі цього навчання вона класифікує дані за різними класами.

Завдання алгоритму класифікації - знайти функцію для відображення входу (x) в дискретний вихід (y).

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		15



Classification

Regression

Рисунок 1.2 – Діаграма алгоритмів класифікації і регресії

Детальніше про метод класифікації буде викладено пізніше.

Регресія - це контрольована техніка навчання, яка допомагає знайти кореляцію між змінними та дозволяє прогнозувати безперервну вихідну змінну на основі однієї або декількох змінних-предикторів. Вона в основному використовується для прогнозування, моделювання часових рядів та визначення причинно-наслідкового зв'язку між змінними[13].

В регресії ми будуємо графік між змінними, який найкраще відповідає заданим точкам даних, використовуючи цей графік, модель машинного навчання може робити прогнози щодо даних. Простими словами, "Регресія показує лінію або криву, яка враховує усі точки даних на графіку прогнозування цілі таким чином, що вертикальна відстань між точками даних та лінією регресії є мінімальною". Відстань між точками даних та лінією вказує, чи зафіксувала модель сильний зв'язок чи ні.

Лінійна регресія, мабуть, один з найвідоміших і добре вивчених алгоритмів в статистиці і машинному навчанні. Вона є традиційною формою регресії, де залежна змінна є безперервною. Для прогнозування цієї змінної встановлюється лінійний зв'язок між нею та незалежними змінними. Тут ми отримуємо пряму лінію, яка проходить через більшість точок даних, і ця лінія діє як передбачення. Оскільки формулою для прямої є $Y = mx + c$, ми маємо дві невідомі, m і c , і ми

вибираємо ці значення m і s , що забезпечує нам мінімальну похибку. Рядок, що забезпечує мінімальну похибку, відомий як лінія максимальної відповідності умовам. Однак визначення помилки може змінюватися залежно від метрики точності. У лінійній регресії, коли похибка обчислюється із використанням суми квадратичної помилки, цей тип регресії відомий як МНК, тобто метод найменших квадратів у випадку квадратичної регресії. Ми можемо мати подібні типи помилок, такі як MAD Regression (Середнє-Абсолютне відхилення регресії), яка використовує середнє абсолютне відхилення для обчислення лінії максимальної відповідності умовам [14].

Неконтрольоване навчання використовується для даних, які не мають історичних міток. Системі не говорять «правильну відповідь». Алгоритм повинен з'ясувати, що показано. Завдання машини при неконтрольованому навчанні - знайти зв'язку між окремими даними, виявити закономірності, підібрати шаблони, упорядкувати дані або описати їх структуру, виконати класифікацію даних. Навчання без вчителя добре працює з транзакційними даними[12]. Наприклад, воно може ідентифікувати сегменти клієнтів зі схожими атрибутами, з якими потім можна так само звертатися в маркетингових кампаніях. Або воно може знайти основні атрибути, які відокремлюють сегменти клієнтів один від одного. До популярних методів відносяться самоорганізовані карти, відображення найближчих сусідів, кластеризації k-середніх та сингулярне розкладання. Ці алгоритми також використовуються для сегментації текстових тем, рекомендованих елементів і виявлення неточних даних.

Для вирішення проблем неконтрольованого машинного навчання використовують широко поширений метод кластеризації.

Використання алгоритму кластеризації означає, що ви збираєтеся надати алгоритму багато вхідних даних без міток і дозволити йому знаходити будь-які угруповання в даних, які він може.

Ці групи називаються кластерами. Кластер - це група точок даних, які схожі один на одного в залежності від їх ставлення до оточуючих точок даних.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

Кластеризація використовується для таких речей, як розробка функцій або виявлення шаблонів.

Коли ви починаєте з даних, про які нічого не знаєте, кластеризація може бути хорошим методом, щоб отримати потрібну інформацію.

Існують різні типи алгоритмів кластеризації, які обробляють всі види унікальних даних: на основі щільності, на основі розподілу, на основі центроїдів та на основі ієрархії[15].

Метод кластеризації на основі щільності пов'язує зони з високою щільністю в кластери, і розподіли довільної форми формуються до тих пір, поки щільна область може бути з'єднана (рис.1.3). Алгоритм працює, ідентифікуючи різні кластери в наборі даних, і об'єднуючи області з високою щільністю в кластери. Щільні області в просторі даних відокремлені один від одного більш розрідженими областями.

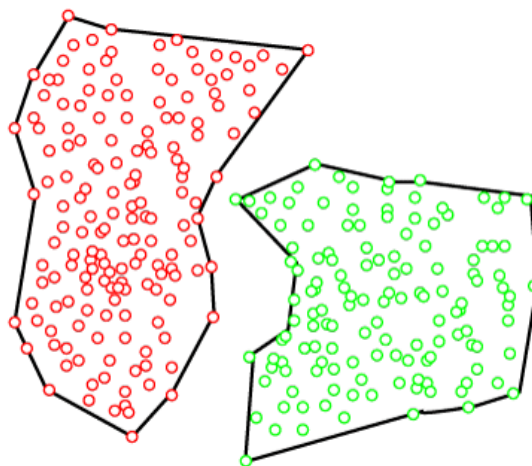


Рисунок 1.3 – Метод кластеризації на основі щільності

Ці алгоритми можуть зіткнутися з труднощами при кластеризації точок даних, якщо набір даних має різну щільність і великі розміри.

У методі кластеризації на основі моделі розподілу дані діляться залежно від ймовірності належності набору даних до певного розподілу. Групування здійснюється шляхом припущення деяких розподілів, зазвичай гауссовського розподілу (рис.1.4).

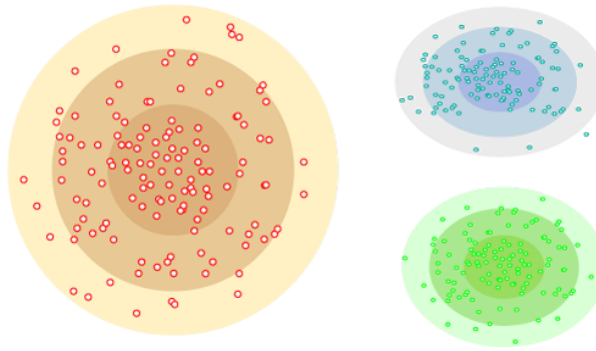


Рисунок 1.4 – Метод кластеризації на основі моделі розподілу

Прикладом цього типу є алгоритм кластеризації очікувань-максимізації, який використовує моделі Гауссової суміші.

Метод на основі центроїдів являється таким типом кластеризації, при якому дані діляться на неієрархічні групи. Найпоширенішим прикладом цього методу є алгоритм кластеризації К-середніх [15].

У цьому типі набір даних ділиться на набір k груп, де k використовується для встановлення кількості заздалегідь визначених груп. Центр кластера створюється таким чином, щоб відстань між точками даних одного кластера була мінімальною у порівнянні з центроїдом іншого кластера (рис.1.5).

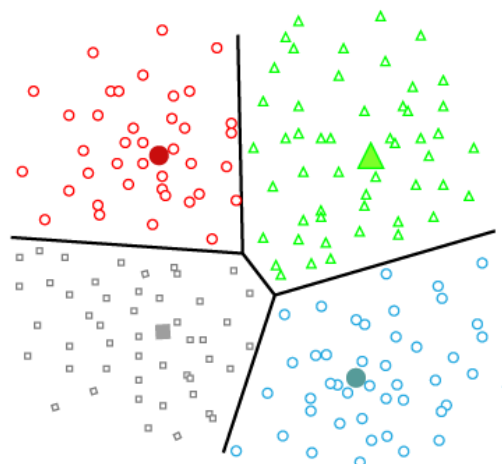


Рисунок 1.5 – Метод на основі центроїдів

Ієрархічна кластеризація може використовуватися в якості альтернативи розділеній кластеризації, оскільки немає необхідності попередньо вказувати кількість кластерів, які потрібно створити. У цьому методі набір даних ділиться на

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

кластери, щоб створити деревоподібну структуру, яку також називають дендрограмою (рис.1.6). Спостереження або будь-яку кількість кластерів можна вибрати, розрізавши дерево на потрібному рівні. Найпоширенішим прикладом цього методу є агломеративний ієрархічний алгоритм[16].

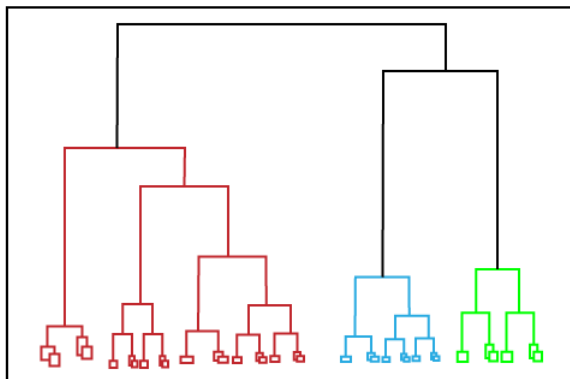


Рисунок 1.6 – Метод ієрархічної кластеризація

Ми розглянули основні методи машинного навчання такі, як контрольоване та неконтрольоване навчання, а також детальніше познайомились з їх алгоритмами та методами. Але є ще інші методи машиного навчання, які не так широко застосовується, але про них також варто знати – напівконтрольоване навчання (навчання з частковим залученням вчителя) та навчання з підкріпленням.

Напівконтрольоване навчання використовується для тих же додатків, що і навчання з учителем. Але для навчання використовуються дані з міткою та без мітки - зазвичай невеликий обсяг даних з міткою та з великим об'ємом даних без мітки (оскільки непомічені дані дешевші і вимагають менше зусиль для збору). Цей тип навчання можна використовувати з такими методами, як класифікація, регресія і прогнозування. Напівконтрольоване навчання корисно, коли витрати, пов'язані з маркуванням, занадто високі, щоб забезпечити повністю маркований процес навчання. Ранні приклади цього включають визначення особи людини на веб-камеру.

Навчання з підкріпленням часто використовується в робототехніці, іграх і навігації. При навчанні з підкріпленням алгоритм методом проб і помилок виявляє, які дії приносять найбільшу вигоду. Цей тип навчання складається з трьох основних компонентів: агента (того, хто навчається або приймає рішення),

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

середовища (всього, з чим агент взаємодіє) і дій (того, що агент може робити). Мета полягає в тому, щоб агент вибирав дії, які максимізують очікувану нагороду протягом заданого проміжку часу. Агент досягне мети набагато швидше, якщо буде дотримуватися правильної політики.[12] Отже, мета навчання з підкріпленням - вивчити кращу політику.

1.3 Аналіз існуючих методів класифікації

Оскільки далі ми будемо працювати з методом контрольованого машинного навчання, а саме з методом класифікації, то розглянемо його детальніше. Тут буде представлений огляд різних методів класифікації, які зазвичай використовуються в методах інтелектуального аналізу даних за різними принципами, а також розглянемо переваги та недоліки кожного алгоритму.

Класифікація - це метод, який класифікує дані по певній кількості класів, до яких присвоюються мітки. Основна мета класифікації - ідентифікувати клас для запуску нових даних шляхом аналізу навчального набору, переглядаючи відповідні межі. Загалом, прогнозування цільового класу і описаний вище процес називається класифікацією [17].

Наприклад, керівництво лікарні записує ім'я пацієнта, адреса, вік, попередню історію здоров'я пацієнта, щоб поставити діагноз і це допомагає класифікувати пацієнтів. Дані можна розділити на дві фази: фаза навчання і фаза оцінки. Фаза навчання моделює підхід на основі навчальних даних, тоді як фаза оцінки передбачає результати для заданих даних. Також метод застосовується для знаходження спаму електронної пошти, прогнозуванні банківських кредитів, розпізнаванні мови, аналізі настроїв. Метод включає математичну функцію f зі входом X і виходом Y .

Класифікація може виконуватися як для структурованих, так і для неструктурованих даних. Її можна розділити на (рис.1.7):

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

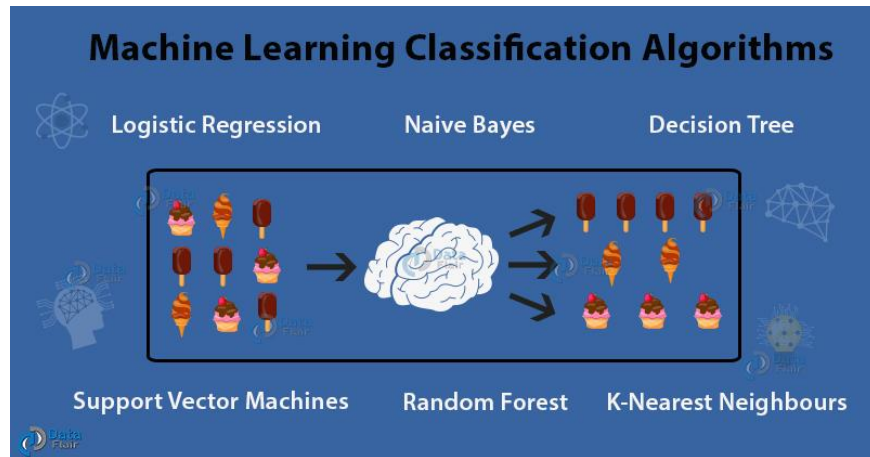


Рисунок 1.7 – Алгоритми класифікації

1. Наївний байесівський метод
2. Класифікатор дерева рішень
3. Метод опорних векторів
4. Випадковий ліс
5. Метод k-найближчих сусідів
6. Логістична регресія

Наївний байесівський метод, заснований на теоремі Байеса, відноситься до статистичних класифікацій, і вимагає невеликої кількості навчальних даних для оцінки параметрів, також відомих як ймовірнісні класифікатори. Він вважається найшвидшим класифікатором, добре масштабується і обробляє як дискретні, так і неперервні дані. Цей алгоритм використовується для прогнозування в реальному часі. Існують різні типи наївних класифікаторів: поліноміальний наївний байесівський класифікатор, наївний байесівський класифікатор Бернуллі, наївний гаусівський класифікатор[19].

Суть класифікатора заснована на теоремі Байеса і визначається за формулою(1.1):

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)} \quad (1.1)$$

Використовуючи теорему Байеса, ми можемо знайти ймовірність того, що відбудеться А, за умови, що В сталося. Тут В - показання, а А - припущення. Зроблене тут припущення полягає в тому, що предиктори(параметр

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

прогнозування) діляться на незалежні ознаки. Тобто наявність однієї особливості не впливає на іншу. Звідси його називають наївним.

Переваги: вимагає значно меншої обчислювальної потужності, використовується в прогнозуванні задачах декількох класів, і точно працює на великих наборах даних.

Недоліки: якщо категоріальна змінна належить до категорії, що не відстежувалася в навчальному наборі, то модель дасть їй ймовірність 0, що завадить їй робити будь-які прогнози. І у них є функції, незалежні один від одного.

Дерево рішень - це алгоритм, який створює деревоподібну структуру або структуру, подібну блок-схемі, де сортування починається від кореневого вузла до кінцевого вузла, поки не буде досягнута мета [20]. Результати прогнозуються на основі заданої вхідної змінної. Дерево рішень складається з наступних елементів: корінь, безліч вузлів, гілок, листя. Кореневий вузол виконує поділ на основі значення атрибута класу, внутрішній вузол приймає атрибут для подальшої класифікації, гілки створюють правило прийняття рішення для поділу вузлів на листові вузли, нарешті, листові вузли дають нам результат. Час виконання дерева рішень залежить від кількості записів, атрибутів навчальних даних. Якщо дерево рішень занадто довге, важко отримати бажані результати. На рисунку 1.8 показано візуалізація алгоритму для кращого розуміння.

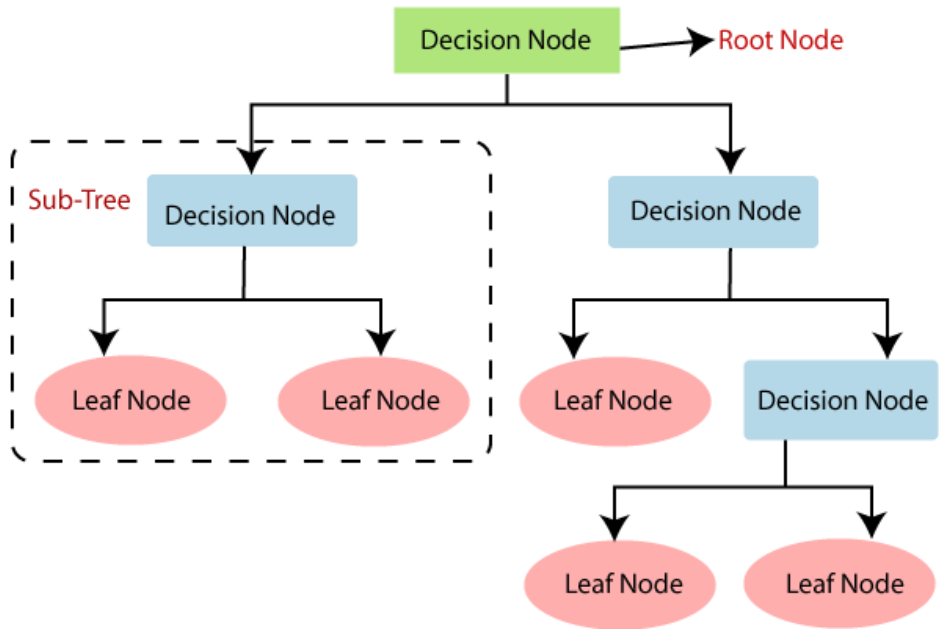


Рисунок 1.8 – Алгоритм дерева прийняття рішень

Переваги: алгоритм враховує кожен можливий результат рішення і відповідно відстежує кожен вузол до виведення. Дерева рішень привласнюють конкретне значення кожної проблеми, рішення і результату. Це зменшує невизначеність і двозначність, а також збільшує ясність. Дерево рішень - один з найпростіших і надійних алгоритмів, оскільки в ньому немає складних формул або структур даних. Для розрахунків потрібно тільки проста статистика і математика.

Недоліки: розмір дерева не піддається контролю, поки для нього не будуть визначені критерії зупинки. Дерево через свою ієрархічну структуру нестабільне.

Далі розглянемо метод опорних векторів (МОВ). Цей алгоритм грає життєво важливу роль в задачах класифікації та найчастіше використовується в контрольованому машинному навчанні [21]. Це важливий інструмент, який задіюють дослідники і фахівці з даними. Цей метод дуже простий, і його процес полягає у пошуку гіперплощини в N-мірному просторі, яка чітко класифікує точки даних. Основна ідея МОВ - знайти максимальну граничну гіперплощину, яка найкращим чином розділить набір даних на класи (рис. 1.9).

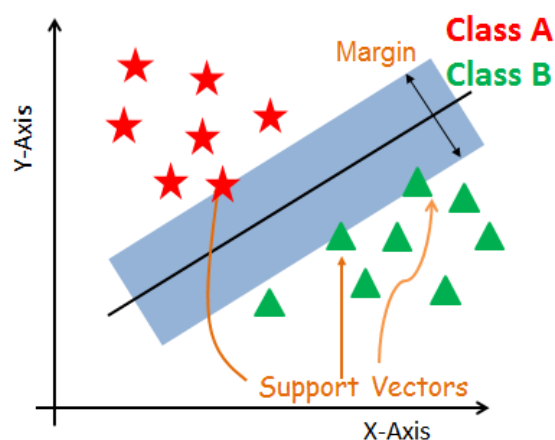


Рисунок 1.9 – Метод опорних векторів

Опорні вектори (Support Vectors) - це точки даних, які знаходяться найближче до гіперплощини. Ці точки будуть краще визначати роздільну лінію при розрахунку полів. Ці моменти більше підходять для побудови класифікатора.

Гіперплощина - це площина прийняття рішень, яка розділяє набір даних, що мають різну приналежність до класів.

Поле (Margin) - це проміжок між двома лініями до найближчих точок класу. Це розраховується як перпендикулярна відстань від лінії до опорних векторів або найближчих точок. Чим більша різниця між класами, тим краще.

МОВ краще будь-якої класифікаційної моделі через їх функції ядра, яка підвищує обчислювальну ефективність.

Метод опорних векторів буває двох видів [21]:

- **Лінійний:** використовується для лінійно відокремлюваних даних, що означає, що якщо набір даних можна класифікувати на два класи за допомогою однієї прямої лінії, то такі дані називаються лінійно відокремлюваними даними, а класифікатор використовується як лінійний класифікатор МОВ.
- **Нелінійний:** використовується для нелінійно відокремлених даних, тобто якщо набір даних неможливо поділити за допомогою прямої лінії, тоді такі дані називаються нелінійними даними, а класифікатор, що використовується, називається нелінійний класифікатор МОВ.

Переваги: менша обчислювальна потужність та ефективна точність і пам'ять. Добре працює в багатовимірному просторі.

Недоліки: обмеження швидкості, ядра та розміру.

Випадковий ліс - це гнучкий, простий у використанні алгоритм машинного навчання, який, навіть без налаштування гіперпараметрів, більшу частину часу дає хороші результати. Це також один з найбільш використовуваних алгоритмів, завдяки своїй простоті та різноманітності. Випадковий ліс будує кілька дерев рішень і об'єднує їх разом, щоб отримати більш точний і стабільний прогноз[22]. Нижче на рисунку 1.10 можемо побачити, як буде виглядати випадковий ліс з двома деревами:

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

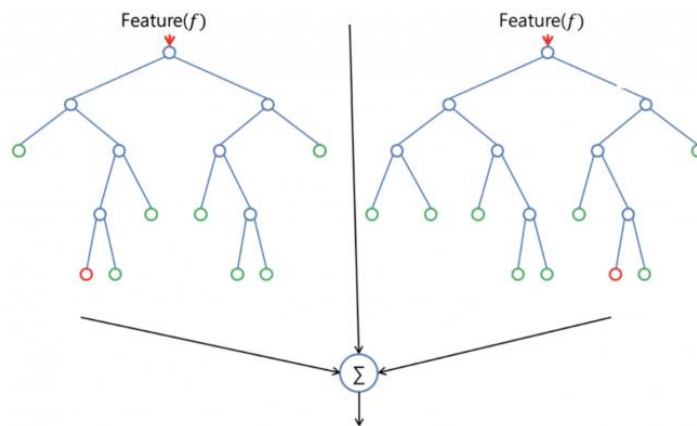


Рисунок 1.10 – Випадковий ліс

Випадковий ліс додає додаткову випадковість моделі під час вирощування дерев. Замість того, щоб шукати найважливішу функцію під час розбиття вузла, вона шукає найкращу функцію серед випадкової підмножини функцій. Це призводить до широкого розмаїття, що, як правило, призводить до кращої моделі.

Випадковий ліс реалізується за допомогою техніки, яка називається беггінг для прийняття рішень. Беггінг використовує паралельне навчання базових класифікаторів. Основна ідея беггінга - усереднити незсунені моделі для створення моделі з низьким розбіжністю в рамках класифікації. Це допомагає досягти кращої точності. Остаточний прогноз робиться на основі середнього з багатьох дерев рішень, тобто часткових прогнозів.

Хоча випадковий ліс являє собою набір дерев рішень, між ними є деякі відмінності.

Якщо введемо навчальний набір даних із функціями та мітками у дерево рішень, він сформулює певний набір правил, які будуть використані для прогнозування. Наприклад, щоб передбачити, чи буде людина натискати на рекламу в Інтернеті, можемо зібрати оголошення, на які користувач натискав у минулому, та деякі функції, що описують його рішення. Якщо помістимо функції та мітки в дерево рішень, воно створить деякі правила, які допоможуть передбачити, чи буде натиснуто рекламу чи ні. Для порівняння, алгоритм випадкового лісу випадковим чином вибирає спостереження та особливості для побудови декількох дерев рішень, а потім усереднює результати.

Інша відмінність полягає в тому, що дерева "глибоких" рішень можуть постраждати від перенавчання. Здебільшого випадковий ліс запобігає цьому, створюючи випадкові підмножини об'єктів та будуючи менші дерева за допомогою цих підмножин. Потім він поєднує піддерева. Важливо зазначити, що це не працює кожного разу, а також робить обчислення повільнішими, залежно від того, скільки дерев будує випадковий ліс [22].

Переваги: не потрібно великої обробки для наборів даних. Дуже проста модель для побудови. Забезпечує більшу точність, допомагає у вирішенні завдань прогнозування. Добре працює при обробці відсутніх значень і автоматично виявляє відхилення.

Недоліки: потрібні високі обчислювальні витрати і великий обсяг пам'яті. Потрібно достатньо часу.

Метод k-найближчих сусідів є дуже простим, зрозумілим, універсальним і одним з кращих алгоритмів машинного навчання [23]. Він є непараметричним та лінійним алгоритмом навчання. Непараметричний означає, що немає ніяких припущень для базового розподілу даних. Іншими словами, структура моделі визначається з набору даних. Це буває дуже корисно на практиці, коли більшість реальних наборів даних не відповідають математичним теоретичним припущенням. Лінійний алгоритм означає, що йому не потрібні будь-які навчальні дані для створення моделі. Всі дані навчання використовуються на етапі тестування. Це прискорює навчання, а етап тестування стає повільнішим і дорожчим. Дорогий етап тестування означає час і пам'ять. У гіршому випадку методу k-найближчих сусідів потрібно більше часу для сканування всіх точок даних, а сканування всіх точок даних вимагатиме більше пам'яті для зберігання навчальних даних.

У цьому методі кількість сусідів є головним вирішальним фактором. K, як правило, непарне число, якщо кількість класів дорівнює 2.

Припустимо, P_1 - це точка, для якої повинна передбачитись мітка. Спочатку знаходимо k, найближчу точку до P_1 , а потім класифікуємо точки більшістю голосів k-сусідів. Кожен об'єкт голосує за свій клас, а клас, який набрав найбільше голосів, приймається за прогноз. Для пошуку найближчих подібних точок

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

знаходимо відстань між точками, використовуючи міри відстані, такі як відстань Евкліда, відстань Хеммінга, відстань Манхеттена та відстань Мінковського [23].

Метод k-найближчих сусідів має такі основні кроки (рис.1.11):

1. Розрахувати відстань
2. Знайти найближчих сусідів
3. Голосувати за мітки

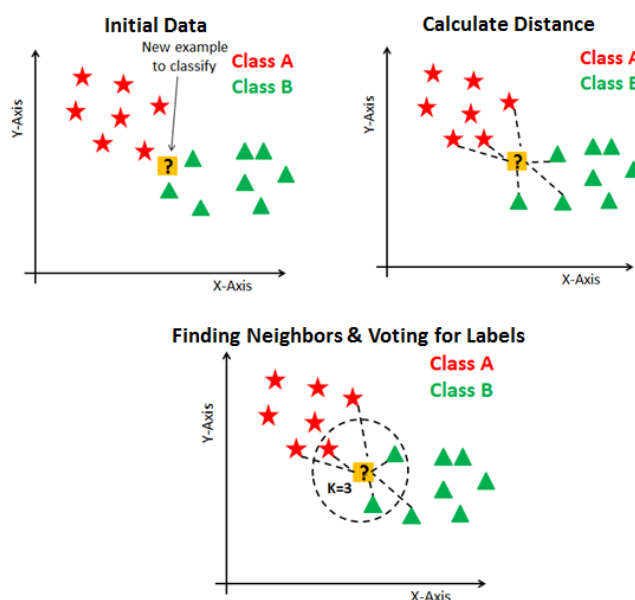


Рисунок 1.11 – Основні кроки в методі k-найближчих сусідів

Переваги: фаза навчання класифікації K-найближчих сусідів виконується набагато швидше в порівнянні з іншими алгоритмами класифікації. Немає необхідності навчати модель для узагальнення, тому метод відомий як простий і заснований на екземплярах алгоритм навчання. Алгоритм корисний у випадку нелінійних даних.

Недоліки: етап тестування класифікації K-найближчих сусідів повільніше і дорожче з точки зору часу і пам'яті. Потрібна велика пам'ять для зберігання всього набору навчальних даних для прогнозування. Об'єкти з високими значеннями будуть мати більшу вагу, ніж об'єкти з низькими значеннями.

Останній алгоритм, який ми розглянемо це логістична регресія. Логістична регресія - один з найбільш часто використовуваних алгоритмів машинного навчання для двійкової класифікації [24]. Це простий алгоритм, який можна використовувати в якості базового показника продуктивності, його легко реалізувати,

і він досить добре справляється з багатьма завданнями. Тому кожен інженер машинного навчання повинен бути знайомий з його концепціями. Концепції логістичної регресії також можуть бути корисними для глибокого навчання при побудові нейронних мереж.

Як і багато інших методів машинного навчання, він запозичений з області статистики і, незважаючи на свою назву, не є алгоритмом для задач регресії, де прогнозується безперервний результат. Замість цього логістична регресія - це метод бінарної класифікації. Він дає вам дискретний двійковий результат від 0 до 1. Простіше кажучи, його результат - це або одне, або інше.

Простим прикладом проблеми логістичної регресії може бути алгоритм, який використовується для виявлення раку, який приймає скринінгове зображення в якості вхідних даних і повинен визначати, чи є у пацієнта рак (1) чи ні (0).

Логістична регресія вимірює взаємозв'язок між залежною змінною (наша мітка, то, що ми хочемо спрогнозувати) і однією або декількома незалежними змінними (нашими характеристиками), шляхом оцінки ймовірностей з використанням її базової логістичної функції.

Потім ці ймовірності повинні бути перетворені в двійкові значення, щоб фактично зробити прогноз. Це завдання логістичної функції, також званої сигмовидною функцією. Сигмоїдальна функція - це S-подібна крива, яка може приймати будь-яке дійсне число і відображати його в значення в діапазоні від 0 до 1, але ніколи не знаходиться точно в цих межах. Ці значення від 0 до 1 потім будуть перетворені в 0 або 1 з використанням класифікатора порогових значень.

На наведеному нижче рисунку показані кроки, логістична регресія проходить, щоб дати вам ваш бажаний результат [24].

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

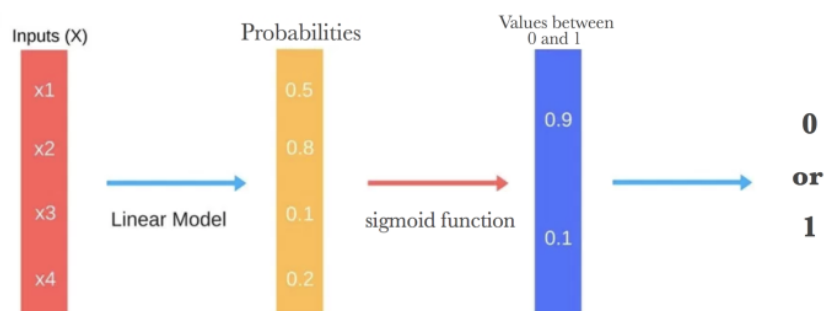


Рисунок 1.12 – Кроки логістичної регресії

Нижче представлено, як виглядає логістична функція (сигмоїдна функція):

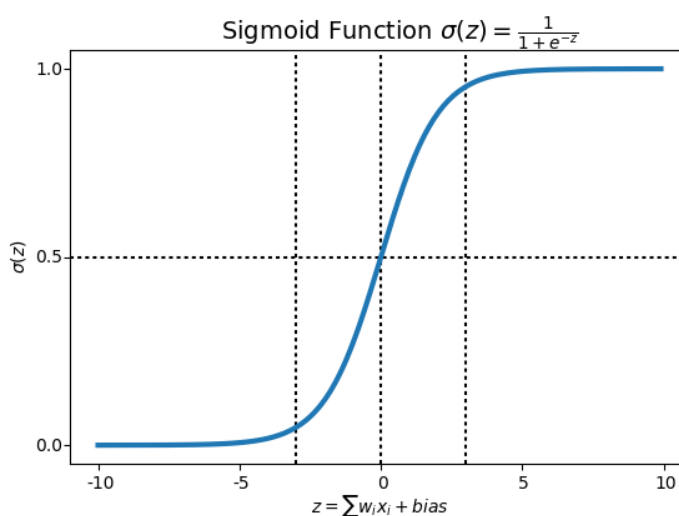


Рисунок 1.13 – Графік сигмоїдної функції

Ми хочемо максимізувати ймовірність того, що випадкова точка даних буде правильно класифікована, як то кажуть оцінкою максимальної правдоподібності. Оцінка максимальної правдоподібності - це загальний підхід до оцінки параметрів в статистичних моделях. Ми можемо максимізувати ймовірність, використовуючи різні методи, наприклад алгоритм оптимізації. Метод Ньютона є таким алгоритмом і може використовуватися для пошуку максимуму (або мінімуму) безлічі різних функцій, включаючи функцію правдоподібності. Замість методу Ньютона також можна використовувати градієнтний спуск.

Переваги: він дуже ефективний, не вимагає занадто великої кількості обчислювальних ресурсів, легко інтерпретується, не вимагає масштабування вхідних функцій, не вимагає будь-якого налаштування, його легко впорядкувати і

виводить добре відкалібровані передбачені ймовірності. Також цей метод є хорошою базою, яку можна використовувати для вимірювання продуктивності інших більш складних алгоритмів.

Недоліки: неможливо вирішувати нелінійні задачі за допомогою логістичної регресії, оскільки її поверхня прийняття рішень лінійна. Високий ступінь залежності від правильного подання набору даних. Це означає, що логістична регресія не є корисним інструментом, якщо ви ще не визначили всі важливі незалежні змінні. Це також алгоритм, який відомий своєю вразливістю до перенавчання.

Ми розглянули можливості різних алгоритмів класифікації, які діють як потужний інструмент у проектуванні функцій, класифікації, і є відмінним ресурсом для машинного навчання. Алгоритми класифікації - це потужні алгоритми, які вирішують складні проблеми.

1.4 Аналіз аналітичних можливостей хмарного середовища Azure

Microsoft Azure - це хмарна платформа, що складається з інфраструктури і служб додатків, з інтегрованими службами даних і розширеною аналітикою, а також засобами та службами розробників, розміщеними в загальнодоступних хмарних центрах обробки даних Microsoft. Клієнти використовують Azure для безлічі різних компонентів і сценаріїв: від базових обчислень, мережних служб і сховищ, мобільних служб і служб веб-додатків до повністю хмарних сценаріїв, таких як Інтернет речей. Таке середовище можна використовувати з технологіями з відкритим кодом, розгортати в якості гібридної хмари або розміщувати в центрі обробки даних клієнта. Azure надає хмарні технології в якості стандартних блоків, допомагаючи компаніям скорочувати витрати, швидко впроваджувати інновації та ефективно управляти системами [25].

Хоча Microsoft Azure має сильних конкурентів таких, як Amazon Web Services та Google Cloud Platform, але, завдяки своїм перевагам виривається вперед і є визнаним лідером як в приватній, так і в публічній хмарі. Microsoft Azure

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

- це єдиний постачальник хмарних обчислень, що надає безпечну узгоджену платформу додатків і інфраструктуру як послугу для команд з різними навичками роботи з хмарою і проектів різного рівня складності з інтегрованими службами даних і аналітичними засобами, що дозволяють отримати інформацію на основі даних в будь-якому місці (на платформах Microsoft і сторонніх постачальників), відкриті платформи і засоби, які надають вибір для інтеграції хмари з локального середовища, а також для розгортання хмарних служб Azure в локальних центрах обробки даних. В рамках довіреної хмари користувачі Microsoft покладаються на Azure заради провідних в області заходів забезпечення безпеки, надійності, відповідності та конфіденційності і великої мережі користувачів, партнерів і процесів для підтримки організацій в хмарі.

Розглянемо деякі хмарні сервіси, які є в Azure Microsoft [26]:

- Azure Databricks - це ключовий інструмент, що допомагає клієнтам масштабувати ІІ і розкривати цінність розрізнених і складних даних. Якщо ви хочете швидко перейти на хмарні сервіси, ми можемо використовувати Azure Databricks для переходу від пропрієтарних і дорогих систем до підвищення операційної ефективності і нових бізнес-моделей.
- Microsoft Azure Synapse робить дані доступними для всіх і руйнує кордони між озерами даних і сховищами даних, пропонуючи широкий спектр аналітичних можливостей. Azure Synapse - це рішення для сховища даних Azure SQL наступного покоління, яке надає безмежну аналітичну службу, яка об'єднує дані вашого підприємства і аналітику великих даних. Synapse дозволяє користувачам бізнес-груп швидше і точніше відповідати на питання про свої дані.
- База даних SQL Azure - це реляційна хмарна база даних, яка дозволяє прискорити розробку додатків і використовувати знайомий механізм бази даних SQL. Можна перенести бази даних SQL-сервера без зміни додатків. Далі підключити свої дані до вбудованого штучного інтелекту та розширених можливостей аналітики, щоб максимально

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

підвищити продуктивність, захистити дані і прискорити розробку додатків. Перевага цього підходу полягає в низьких витратах на обслуговування і такої ж продуктивності, масштабованості і надійності хмари Microsoft Azure в моделях IaaS і PaaS.

- Когнітивні служби Azure використовують алгоритми обробки зображень, складне відображення інформації в поєднанні з семантичним пошуком, обробкою природної мови, перетворенням мови і пошуком для вирішення найскладніших бізнес-завдань. Завдяки найширшим в світі можливостям Microsoft Azure, він є найкращим вибором, який допоможе вирішити ці проблеми.
- Microsoft Azure Stack гібридне хмарне рішення, яке дозволяє запускати сервіси Azure у власному центрі обробки даних організації, а значить, отримати зручні засоби розробки, управління та забезпечення безпеки, аналогічні загальнодоступним хмарним службам Azure. Azure Stack є розширення Azure. Це рішення дозволяє працювати з сервісами Azure в локальних середовищах, а при необхідності - переходити на використання публічного хмари Azure.
- ШІ Azure та МН Microsoft Azure дозволяє фахівцям з обробки даних готувати дані, розробляти експерименти і розгортати моделі в хмарному масштабі. Потім можна використовувати контейнери Docker для експериментів в керованих середовищах і мати можливість використовувати сучасне обладнання для прискорення виконання. Машинне навчання Azure включає три компоненти (Workbench, Experimentation Service і Model Management Service), навчальну бібліотеку, а також інструменти Visual Studio Code для ШІ.

Останню службу розглянемо детальніше.

Платформа ШІ Azure являє собою уніфіковану платформу для машинного навчання з її API і інфраструктурними службами. Отже, далі ознайомимось з основними службами, запропонованими Azure з точки зору рішень для машинного навчання.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Машинне навчання Azure - це основне середовище для управління наборами даних, навчання моделей і розгортання.

Платформа надає студію машинного навчання, веб-середовище та середовище з низьким кодом, для швидкого налаштування операцій і конвеєрів машинного навчання. Як правило, Студія МН Azure має засоби для дослідження даних, попередньої обробки, вибору методів і перевірки результатів моделювання. Студія підтримує близько 100 методів, що відносяться до класифікації (двійковий + багатокласовий), виявлення аномалій, регресії, рекомендацій і аналізу тексту. Варто відзначити, що в платформі є один алгоритм кластеризації (K-means).

Як і Amazon, Azure пропонує інтеграцію з Jupyter для написання і запуску коду в Студії МН. Він також забезпечує виконання середовища ONNX (Open Neural Network Exchange) для прискорення моделей МН у різних ОС, апаратних платформах та фреймворках. Середовище виконання також можна використовувати для забезпечення взаємодії між різними платформами машинного навчання. Azure підтримує такі популярні, як TensorFlow, PyTorch, scikit-learn і інші.

У Студії МН є ряд додаткових функціоналів [27]:

- Конструктор машинного навчання Azure - це графічний користувацький інтерфейс з перетягуванням (drag-and-drop) для студії машинного навчання, який забезпечує доступ до функцій платформи і управління ними. Тут можна змінювати дані, застосовувати методи машинного навчання і розгортати рішення на сервері.
- Автоматизоване машинне навчання - це SDK(Software Development Kit), який забезпечує навчання моделі без коду до моделі з низьким рівнем коду. По суті, автоматизоване машинне навчання доповнює студію машинного навчання високим ступенем автоматизації стандартних завдань і підтримкою дослідження даних, налаштування моделі та розгортання. Azure визначає завдання класифікації, регресії і прогнозування часових рядів стосовно навчання за допомогою інструментів автоматизованого машинного навчання.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

- Пакети SDK для мови Azure ML Python і R повністю інтегровані в ML Studio.
- Підтримка таких фреймворків МН, як PyTorch, TensorFlow та scikit-learn, крім того, Azure пропонує взаємодію між фреймворками за допомогою середовища виконання ONNX.
- Вбудовані модульні конвеєри, що дозволяє команді створити власний конвеєр даних для власного проекту машинного навчання.
- Підтримка проектів маркування даних, включаючи управління даними та командами, прогрес маркування, неповне відстеження маркування та експорт інструментів помічених даних.
- Цільові об'єкти обчислень, які налаштовуються для розгортання моделі, що підтримують різні хмарні служби, такі як служби Azure Kubernetes, екземпляри контейнерів і обчислювальні кластери.
- Інструменти MLOps існують для управління, розгортання і моніторингу моделей в автоматизованих конвеєрах.

І конструктор машинного навчання, і автоматизоване машинне навчання надають недосвідченим користувачам засоби для створення рішень машинного навчання. У свою чергу, студія машинного навчання Azure включає низку функцій, якими можуть використовувати досвідчені фахівці в галузі даних та рішення корпоративного рівня. Але це не обмежує ці засоби, оскільки МН Azure призначений для використання його як єдиної платформи з усіма її можливостями.

Підхід машинного навчання, за допомогою Azure, вимагає деякого навчання. Але в кінцевому підсумку це призводить до глибшого розуміння всіх основних методів в цій галузі. Графічний інтерфейс МН Azure візуалізує кожен крок робочого процесу і підтримує новачків. Можливо, головна перевага використання Azure - це різноманітність доступних алгоритмів.

1.5 Постановка завдання на розробку

Одним з ключових чинників при прогнозуванні майбутньої проблеми є розробка моделі машинного навчання. Модель машинного навчання - це те, що

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

необхідно отримати, комбінуючи дані з алгоритмом машинного навчання. Існує безліч алгоритмів машинного навчання, які були розроблені для вирішення різних задач.

У даному проєкті буде побудована модель класифікації на базі конструктора машинного навчання Azure. Побудова моделі буде включати такі етапи:

- Пошук та імпорт вхідного набору даних. Наші дані будуть завантажені з веб-ресурсу;
- Аналіз вхідних даних. Це процес, в ході якого ми вивчаємо набір даних, щоб ознайомитися з ним. Ми з'ясовуємо форму даних, відсутні значення, тип даних і т.д.;
- Попередня підготовка даних. Завдання підготовки даних включають збір даних, очищення, агрегування, збільшення, маркування, нормалізацію і перетворення, а також будь-які інші дії для структурованих, неструктурованих і напівструктурованих даних;
- Поділ даних на навчання і тестування. Для цілей навчання моделі ми надаємо тільки навчальні дані. Після навчання моделі з використанням цих даних ми порівнюємо якість навченої моделі на основі даних для тестування.
- Розробка та навчання моделі класифікації. На цьому етапі потрібно вибрати і застосувати методики моделі, навчання моделі, встановлення та налаштування гіперпараметрів моделі, перевірка моделі, розробка і тестування ансамблевої моделі, вибір алгоритму і оптимізація моделі;
- Оцінка та аналіз якості моделі. Це включає в себе оцінку метрики моделі, розрахунки матриці неточностей, метрики продуктивності моделі, вимірювання якості моделі і остаточне визначення того, чи може модель досягати встановленої мети;
- Розгортання моделі у вигляді кінцевого продукту. Це потрібно для того, щоб використовувати модель в клієнтських додатках.

		Побережна М.О.			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

РОЗДІЛ 2 РОЗРОБКА ХМАРНОГО РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ МЕДИЧНИХ ДАНИХ НА ОСНОВІ КОНСТРУКТОРА МАШИННОГО НАВЧАННЯ AZURE

2.1 Розробка архітектури рішення з використанням студії машинного навчання Azure

Машинне навчання є основою для більшості рішень штучного інтелекту (ШІ), а створення інтелектуального рішення часто починається з використання машинного навчання для навчання прогнозної моделі за допомогою зібраних даних за минулі періоди.

Навчання і розгортання ефективної моделі машинного навчання вимагає багато часу, зусиль і ресурсів. Машинне навчання Azure - це хмарна служба, яка допомагає спростити деякі завдання і скоротити час, що витрачається на підготовку даних, навчання моделі і розгортання прогнозної служби.

Існує чотири основні кроки для створення додатка (або моделі) машинного навчання.

Крок 1.Обрати та підготувати набір даних для навчання.

Навчальні дані - це набір даних, які модель машинного навчання отримає для вирішення проблем, для яких вона призначена. У деяких випадках навчальні дані позначені як - «теги», щоб визначити функції і класифікації, які модель повинна буде ідентифікувати. Інші дані не позначені, і моделі необхідно буде самостійно вибирати ці особливості та призначити класифікації.

У будь-якому випадку дані для навчання повинні бути правильно підготовлені - рандомізовані, усунуті дублювання і перевірені на наявність дисбалансу або зсувів, які можуть вплинути на навчання. Їх також слід розділити на дві підмножини: навчальна підмножина, яка буде використана для навчання додатку та підмножина оцінки, що використовується для тестування та вдосконалення.

Крок 2. Обрати алгоритм для роботи з навчальним набором даних

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

Знову ж таки, алгоритм - це набір етапів статистичної обробки. Тип алгоритму залежить від типу (з міткою або без мітки) і кількості даних в навчальному наборі даних, а також від типу задачі, яку потрібно вирішити. Вибір правильного алгоритму може здатися складним завданням, оскільки існують десятки контрольованих і неконтрольованих алгоритмів машинного навчання, і кожен використовує свій підхід до навчання[29].

Для нашої моделі буде використовуватись алгоритми контрольованого навчання, а саме класифікація.

Класифікація - це приклад контрольованого машинного навчання, в ході якого модель навчається за допомогою використання даних, що включають в себе як компоненти, так і відомі значення для мітки. Таким чином, модель навчається виконувати підгонку комбінацій компонентів до мітки. Після завершення навчання можна використовувати навчену модель для прогнозування міток для нових компонентів, для яких мітка невідома.

Класифікація - це форма машинного навчання, яка використовується для прогнозування категорії або класу, до якого належить елемент. Наприклад, клініка в процесі проведення здоров'я може використовувати характеристики пацієнта (такі як вік, вага, кров'яний тиск і т. д.), щоб передбачити ризик діабету. У цьому випадку характеристики пацієнта є компонентами (ознаками), а мітка - класифікацією 0 або 1, яка вказує на відсутність або наявність діабету[30].

Крок 3. Навчання алгоритму створення моделі.

Навчання алгоритму - це ітеративний процес: він включає в себе прогін змінних через алгоритм, порівняння вихідних даних з результатами, які він повинен був отримати, коригування значень і зміщень в алгоритмі, які можуть дати більш точний результат. Також повторний запуск змінних до тих пір, поки алгоритм в більшості випадків не поверне правильний результат. Отриманий в результаті навчений точний алгоритм являє собою модель машинного навчання – це важлива відмінність, яку слід зазначити, тому що «алгоритм» і «модель» не можуть бути взаємозамінними, навіть в машинному навчанні.

Крок 4. Використання і поліпшення моделі.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

Останнім кроком є використання моделі з новими даними і, в кращому випадку, підвищення її точності і ефективності з плином часу. Звідки будуть надходити нові дані, буде залежати від розв'язуваної проблеми. Наприклад, модель машинного навчання, розроблена для виявлення спаму, буде приймати повідомлення електронної пошти, тоді як модель машинного навчання, яка управляє роботом-пилососом, буде приймати дані, отримані в результаті реальної взаємодії з переміщеною меблями або новими об'єктами в кімнаті.

Дана модель буде розроблятися за допомогою конструктора. Конструктор – це засіб перетягування, котрий дозволяє створювати моделі машинного навчання без єдиного рядка коду. Конструктор Машинного навчання Azure дозволяє візуально з'єднувати набори даних і модулі на інтерактивному полотні для створення моделей машинного навчання.

Основними етапами створення моделі за допомогою конструктора є такі:

- створення робочої області Azure;
- створення конвеєру;
- імпортування даних;
- підготовка даних;
- навчання моделі машинного навчання;
- оцінка моделі машинного навчання;
- розгортання моделі у вигляді кінцевого продукту прогнозування в реальному часі.

Почнемо з першого етапу – створення робочої області Azure. Робоча область - це ресурс верхнього рівня для Машинне навчання Azure, що надає централізоване місце для роботи з усіма артефактами, створюваними при використанні Машинне навчання Azure. Робоча область зберігає історію всіх навчальних запусків, включаючи журнали, метрики, вихідні дані і моментальні знімки сценаріїв.

Для навчання та розгортання моделей за допомогою конструктора Машинного навчання Azure потрібні обчислення, на основі яких виконується процес навчання і тестування навченої моделі після її розгортання.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

Цільові об'єкти обчислень - це хмарні ресурси, на основі яких можна виконувати процеси навчання моделі і дослідження даних.

У студії машинного навчання Azure перейдемо на сторінку Обчислення (розділ Управління). Саме тут здійснюється управління цільовими об'єктами обчислень для обробки і аналізу даних. Можна створити чотири типи обчислювальних ресурсів.

- Обчислювальні екземпляри: робочі станції розробки, які фахівці з обробки та аналізу даних можуть використовувати для роботи з даними і моделями.
- Обчислювальні кластери: масштабовані кластери віртуальних машин для обробки коду експерименту за запитом.
- Кластери виведення: цільові об'єкти розгортання для прогнозних служб, які використовують навчені моделі.
- Підключення обчислення: посилання на існуючі обчислювальні ресурси Azure, такі як віртуальні машини або кластери Azure Databricks.

На вкладці Обчислювальні екземпляри додамо новий екземпляр обчислень. Він буде використовуватися в якості робочої станції для тестування моделі.

Під час створення обчислювального екземпляру перейдемо на вкладку Обчислювальні кластери і додамо новий обчислювальний кластер. Він буде використовуватися для навчання моделі машинного навчання.

Далі потрібно створити конвеєр машинного навчання. Він використовується для створення робочих процесів і управління ними, які об'єднують етапи машинного навчання. Наприклад, конвеєр може включати етапи підготовки даних, навчання та розгортання моделі, а також виведення і оцінки. Кожен етап може охоплювати кілька кроків, кожен з яких може автоматично виконуватися в різних цільових об'єктах обчислення.

Кроки конвеєра можна використовувати повторно або виконувати без повтору попередніх кроків, якщо вихідні дані таких кроків не повинні змінюватися. Наприклад, ви можете повторно навчити модель без дорогих етапів

		Побережна М.О.			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

підготовки даних, якщо самі дані не змінювалися. Крім того, конвеєри дозволяють фахівцям з обробки та аналізу даних паралельно працювати над різними частинами робочого процесу машинного навчання.

Спочатку у студії машинного навчання Azure для нашої робочої області відкриваємо сторінку Конструктор і вибираємо +, щоб створити новий конвеєр. В області Параметри змінимо ім'я конвеєра за замовчуванням (Конвеєр-створений- дата **) на Навчання моделі даних про хвороби серця(рис.2.1). Після надання ім'я конвеєру необхідно вказати цільовий об'єкт обчислень, в якому буде виконуватися конвеєр. В області Параметри натискаємо Вибрати цільовий об'єкт обчислень і встановлюємо цільовий об'єкт обчислень.

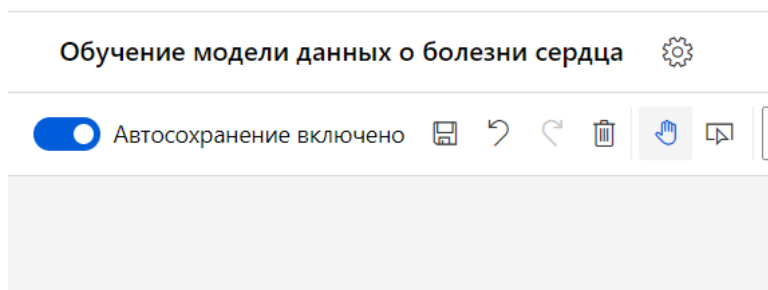


Рисунок 2.1 – Створення та перейменування конвеєра

Конвеєр виконується з цільовим об'єктом обчислень, який є обчислювальним ресурсом, підключеним до нашої робочої області. Після створення цільового об'єкта обчислень його можна повторно використовувати при наступних запусках.

Для всього конвеєра можна задати Цільовий об'єкт обчислень за замовчуванням, що дозволить кожному модулю використовувати один і той же цільовий об'єкт обчислень за замовчуванням. Однак, цільові об'єкти обчислень можна задати конкретно для кожного модуля.

1. Поруч з ім'ям конвеєра обираємо значок Шестерінки у верхній частині полотна, щоб відкрити область Параметри.
2. В області Параметри праворуч від полотна натискаємо Виберіть цільовий об'єкт обчислень.
3. Вказуємо ім'я обчислювального ресурсу.
4. Натискаємо Зберегти.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

Після виконаних дій, наше середовище готове для роботи з конструктором.

2.2 Аналіз вхідних даних та їх попередня обробка

Для навчання моделі класифікації необхідний набір даних, що включає історичні данні (характеристики сутності, для якої необхідно створити прогноз) і відомі значення міток (показник класу, який модель повинна навчитися прогнозувати).

Створюємо набір даних з веб-файлів, який буде називатись «heart-disease», які містять в собі інформацію про пацієнтів, а саме їх дані про аналіз крові та інші показники.

Після завантаження даних, у лівій частині конструктора розгорнемо розділ Набори даних і перетягнемо на полотно набір даних heart-disease (рис.2.2), створений вище.

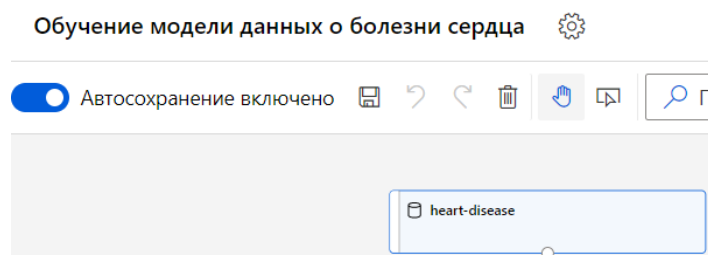


Рисунок 2.2 – Набір даних heart-disease

Переглянемо дані, натиснувши на полотні набір даних та обравши меню Вивода набору даних пункт Візуалізація (рис.2.3).

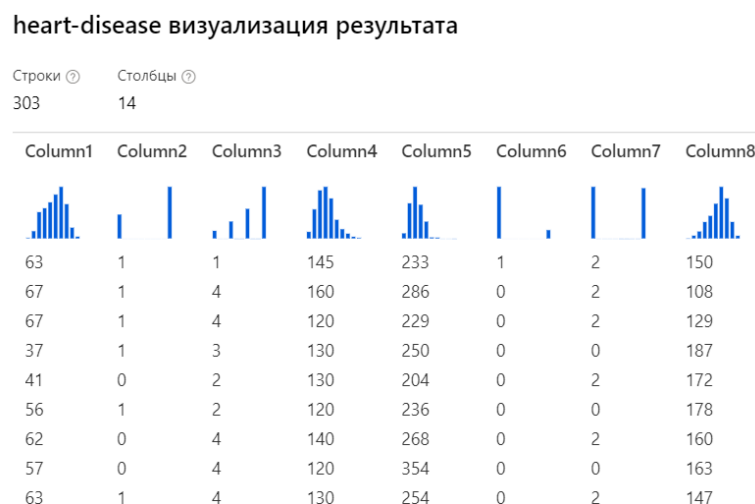


Рисунок 2.3 – Перегляд набору даних

Наші дані, які були імпортовані з веб-файлу, не мають назв стовпців. Щоб це змінити, додамо модуль Edit Metadata (Редагування метаданих) в конвеєр і підключимо до нашого набору даних. У модулів конвеєра є вхідні і вихідні порти, які представляють собою маленькі кола. Вхідні порти завжди розташовані вгорі, а вихідні - внизу. Щоб створити потік даних через експеримент, підключимо вихідний порт одного модуля до вхідного порту іншого, як показано на рисунку 2.4

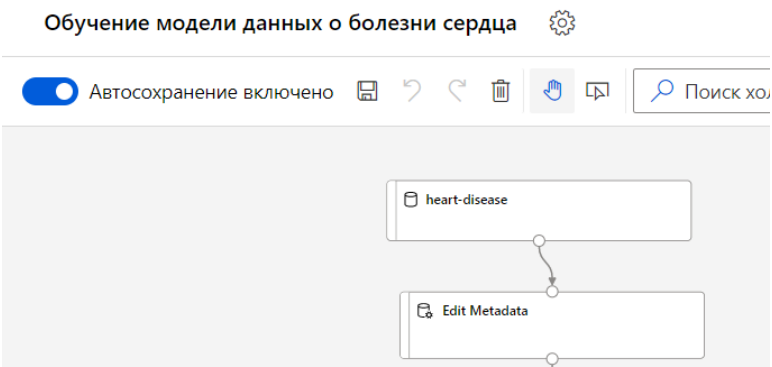


Рисунок 2.4 – Підключення модулю Редагування матаданих

Далі обираємо стовпці, з якими хочемо працювати та змінюємо назви. Після виконаних змін, запускаємо конвеєр як експеримент. Для цього створюємо новий експеримент heart-disease-training та натискаємо Відправити. Дочекавшись завершення роботи, обираємо модуль Edit Metadata (Редагування метаданих) та натискаємо Візуалізація даних для змінених даних. Результат бачимо на рисунку 2.5.

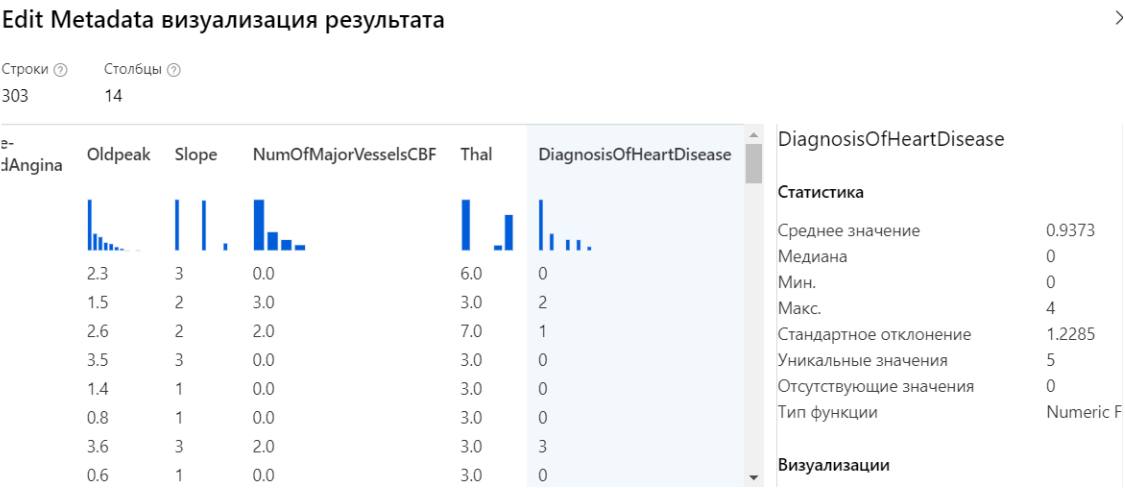


Рисунок 2.5 – Змінений набір даних

Тепер проведемо аналіз вхідних даних. Данні представлені в табличному форматі, де кожен стовпець представляє компоненти, які будуть використовуватися для прогнозування мітки. Правий крайній стовпець «DiagnosisOfHeartDisease» містить в собі значення: 0, 1, 2, 3, 4. Ці значення представляють дві можливих мітки для класу, яку буде прогнозувати модель, 0 означає, що у пацієнта не виявлено серцевих захворювань, а значення 1, 2, 3, 4 – що воно наявне. Набір даних являє собою таблицю, що має 303 рядків і 14 стовпців. Розглянемо кожен стовпець:

Стовпець «Age» : Вік пацієнта.

Стовпець «Sex» : Стать пацієнта (1 – чоловік, 2 – жінка).

Стовпець «ChestPainType» : Тип болю в грудях (1 – типова стенокардія, 2 – атипова стенокардія, 3 – неангінозний біль, 4 – симптоми відсутні).

Стовпець «RestingBloodPressure» : Артеріальний тиск у стані спокою.

Стовпець «Cholesterol» : Показник холестерину.

Стовпець «BloodSugar» : Цукор в крові натще > 120 мг/дл (1 – істинно, 0 – хибно).

Стовпець «ElectrocardiographicRes» : Результати електрокардіографії в стані спокою (0 – нормальне , 1 – має аномалії хвилі ST-T, 2 – показ імовірної або певної гіпертрофії лівого шлуночка за критеріями Естеса).

Стовпець «MaxHeartRate» : Досягнутий максимальний пульс.

Стовпець «Exercise-inducedAngina» : Стенокардія, спричинена фізичними вправами (1 – так, 0 – ні).

Стовпець «Oldpeak» : Депресія ST, спричинена фізичними вправами відносно відпочинку.

Стовпець «Slope» : Нахил піку вправи сегмента ST (1 – підвищення, 2 – плоский, 3 – спад).

Стовпець «NumOfMajorVesselsCBF» : Кількість основних судин (0-3), пофарбованих флюороскопією.

Стовпець «Thal» : Результати сканування товщини серця (3 – нормальний, 6 – виправлений дефект, 7 – оборотний дефект).

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

Стовпець «DiagnosisOfHeartDisease» : Результат діагностики серцевих захворювань (0 – відсутнє, 1,2,3,4 – наявнє).

Перш ніж приступити до навчання моделі, необхідно застосувати ще деякі попередні перетворення до даних. Після того як ми змінили назву стовпців, застосуємо повторно модуль Edit Metadata (Редагування метаданих), але з іншою метою. Потрібно обрати стовпці, які будемо розглядати як категорії. Для цього модуль перетягнемо на полотно, підключимо до попереднього та змінимо параметр на Категоріальний для вибраних стовпців, як бачимо на рис. 2.6. В цьому випадку, відзначаючи стовпець як категоріальний, ми гарантуємо, що значення будуть використовуватися тільки для групування даних, а не в числових обчисленнях.

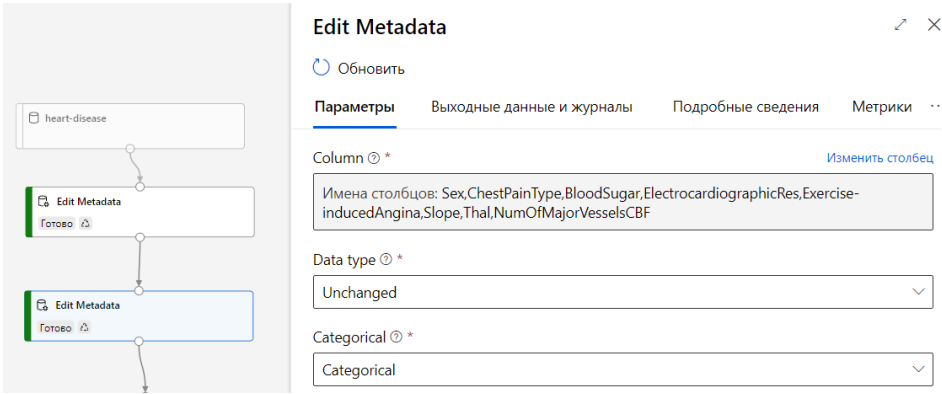


Рисунок 2.6 – Застосування змін для вибраних стовпців

Оскільки у нас багато категоріальних змінних, то зробимо з них фіктивні змінні, тобто перетворимо стовпці, що містять значення категорії, в послідовність стовпців двійкових індикаторів. Це легко зробити за допомогою модуля Convert to Indicator Values (Перетворити в значення індикатора) (рис.2.7). Це допоможе нам краще зрозуміти значення конкретної категоріальної змінної.

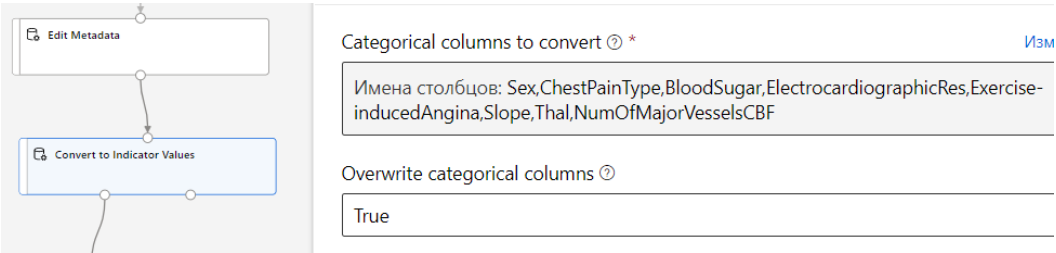


Рис. 2.7. Вибір категоріальних стовпців

Запускаємо конвеєр, використовуючи раніше створений експеримент. Дочекавшись завершення, перевіряємо очищений набір даних. Для цього натискаємо на модуль і справа на панелі обираємо пункт Візуалізація. Для того, щоб краще зрозуміти роботу модуля, переглянемо його результат виконання (рис.2.8).

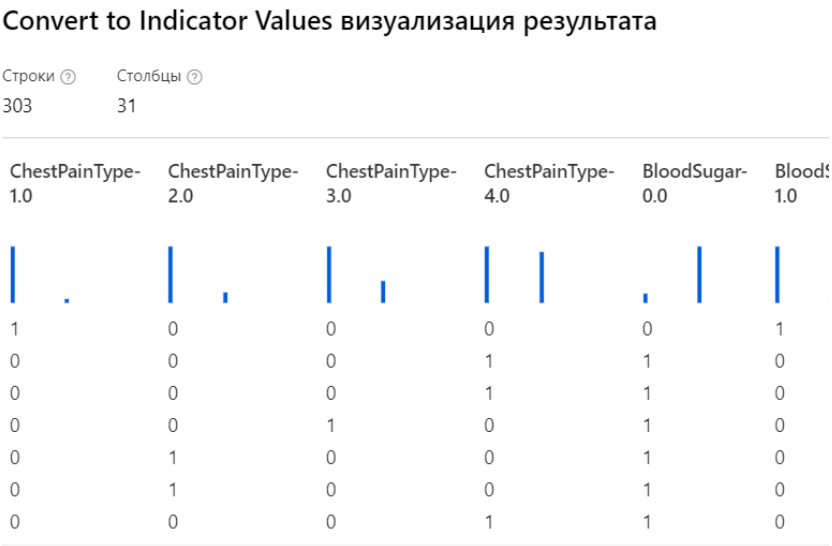


Рисунок 2.8 – Результат виконання модуля Convert to Indicator Values

Як бачимо, стовпець «ChestPainType» поділений на 4 стовпці по різним типам болю в грудях. За допомогою цього модулю, ми покращуємо якість прогнозуючої моделі.

Переглянувши отримані дані, було виявлено лишні стовпці (рис.2.9), позбудемось їх за допомогою модуля Select Columns in Dataset (Вибрати стовпці в модулі набору даних). Тобто оберемо ті стовпці з якими наша модель буде працювати надалі.

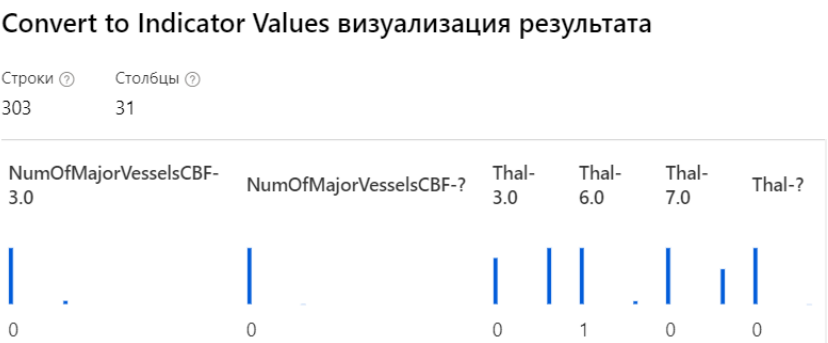


Рисунок 2.9 – Результат виконання модуля Convert to Indicator Values

Виконаємо останній крок підготовки даних. Для цього використаємо модуль Execute R Script (Виконання R скрипта) (рис.2.10).

З допомогою цього скрипту ми змінимо значення в стовпці «DiagnosisOfHeartDisease». Тобто наявність захворювання буде мати мітку 1, замість 1 – 4, а відсутність – 0.

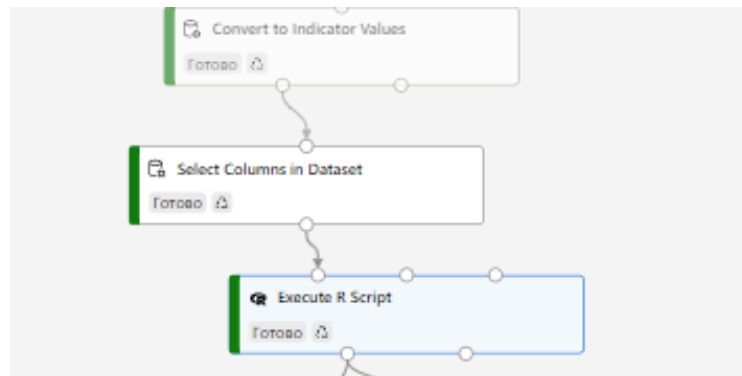


Рисунок 2.10 – Модуль Execute R Script

Лістинг розробленого коду подано на рис.2.11:

```
# Param<dataframe1>: a R DataFrame
# Param<dataframe2>: a R DataFrame
azureml_main <- function(dataframe1, dataframe2){
  print("R script run.")
  # Map 1-based optional input ports to variables
  dataset1 <- dataframe1 # class: data.frame

  #num: diagnosis of heart disease (angiographic disease status)
  #      -- Value 0: < 50% diameter narrowing
  #      -- Value 1: > 50% diameter narrowing
  dataset1$DiagnosisOfHeartDisease[dataset1$DiagnosisOfHeartDisease > 0] <- 1
  data.set = dataset1
  # Select data.frame to be sent to the output Dataset port

  # Return datasets as a Named List
  return(list(dataset1=data.set, dataset2=dataframe2))
}
```

Рисунок 2.11 – Лістинг коду модуля Execute R Script

Результат виконання скрипта бачимо на рисунку 2.12:

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

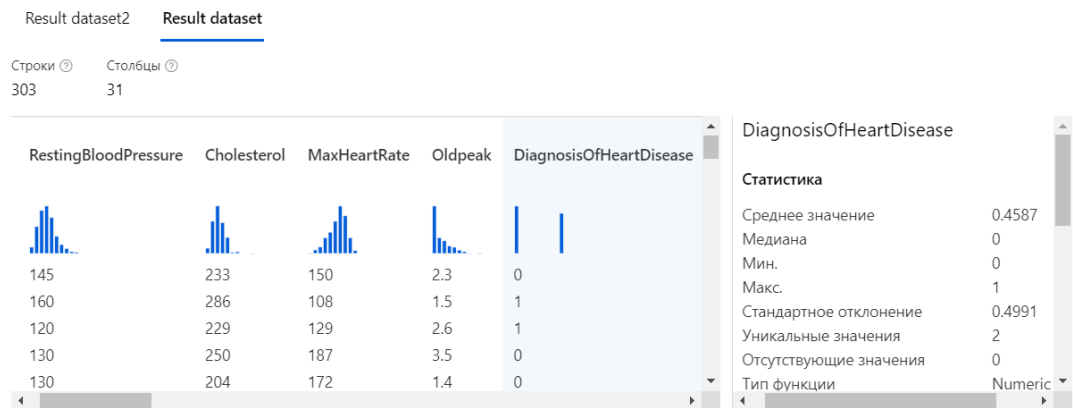


Рисунок 2.12 – Результат виконання модуля Execute R Script

2.3 Розробка та навчання моделі для класифікації

Дані готові для навчання і тестування прогнозової моделі. Ми вже ознайомились з різними методами контрольованого і неконтрольованого машинного навчання. Для нашої моделі був обраний метод класифікації, оскільки на виході будемо отримувати дві мітки наявності і відсутності серцевих захворювань. Дана модель використовує інструмент – дерево прийняття рішень. Це засіб підтримки прийняття рішень застосовується в машинному навчанні, аналізі даних і статистиці. В даному випадку допомагає розв’язати задачу класифікації, коли цільова змінна є дискретною.

Для навчання моделі, передаємо їй вже підготовлений набір даних. У ході роботи модель аналізує дані, шукає закономірності і прогнозує результат. Далі тестуємо її на коректність спрогнозованих даних.

Оскільки наша модель буде навчатись двома способами, умовно поділимо її на дві частини. Та почнемо розглядати перший спосіб.

Дані потрібно розділити на два набори, один буде використовуватись для навчання, а інший – для перевірки. Вибираємо і перетягуємо в область конструктора Split Data (Поділ даних) та підключаємо до останнього модулю Execute R Script. Для налаштування параметрів, натискаємо на модуль. Збоку з'явиться

панель, де можна задати потрібні нам властивості. Параметру Fraction of rows in the first output dataset (Частка рядків в першому вихідному наборі даних) задаємо число 0.8. Це означає, що 80 відсотків даних будуть використанні для навчання, а 20 відсотків для тестування. Для параметру Stratified split (Стратифікований розподіл) обираємо значення True, щоб два набори містили репрезентативну вибірку значень і вказуємо значення для поля Stratification key column (Стовпець ключа стратифікації) стовпець DiagnosisOfHeartDisease. За допомогою стратифікованої вибірки дані діляться таким чином, що кожен вихідний набір даних отримує приблизно однаковий відсоток кожного цільового значення. В даному випадку цільовим значенням буде мітка серцевого захворювання, тобто в кожному наборі буде збалансована кількість позитивних і негативних міток. Запускаємо конверсер як експеримент. Тепер дані будуть передаватись наступним модулям для навчання.

Наступним кроком буде вибір алгоритму класифікації, базуючись на попередньому аналізі, наша модель використовує модуль двокласового збільшеного дерева прийняття рішень. Цей модуль використовується для створення моделі машинного навчання на основі алгоритму підвищеного дерева прийняття рішень. Збільшене дерево прийняття рішень - це метод навчання ансамблів в якому друге дерево виправляє помилки першого дерева, а третє дерево виправляє помилки першого і другого дерев і т. д. Прогнози засновані на повному ансамблі дерев, які роблять прогноз[32].

Параметри налаштування алгоритму показано на рис. 2.13.

Оскільки для параметра Create trainer mode (Створити режим вчителя) обраний варіант Parameter Range (Діапазон параметрів), підключимо набір даних з тегами і навчимо модель за допомогою модуля Tune Model Hyperparameters (Налаштування гіперпараметрів моделі). Мета полягає у визначенні оптимальних гіперпараметрів для моделі машинного навчання. Модуль створює та тестує кілька моделей, використовуючи різні комбінації параметрів. Він порівнює метрики по всім моделям, щоб отримати оптимальні параметри.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

Рисунок 2.13 – Параметри налаштування алгоритму Two-Class Boosted Decision Tree

Терміни параметр та гіперпараметр можуть заплутати. Параметри моделі - це те, що встановлюється на правій панелі модуля. Фактично, цей модуль виконує підбір значень параметра за заданим значенням для нього. Він вивчає оптимальний набір гіперпараметрів, який може бути різним для кожного конкретного дерева рішень, набору даних або методу регресії. Процес пошуку оптимальної конфігурації іноді називають налаштуванням[33].

Налаштуємо параметри модуля на панелі властивостей. Для Specify parameter sweeping mode (Вкажіть режим розгортки параметрів) обираємо Random sweep (Випадковий підбір значень). При виборі цього параметра модуль буде випадковим чином підбирати значення параметрів в діапазоні, заданому системою. Необхідно вказати максимальну кількість запусків, яке повинен виконати модуль. Цей параметр використовується, якщо необхідно підвищити продуктивність моделі, використовуючи вибрані метрики, при цьому, заощаджуючи обчислювальні ресурси. А для Metric for measuring performance for classification (Метрика для вимірювання ефективності для класифікації) буде обрано таку метрику, як правильність (акуратність) це відношення кількості істинних результатів до загальної кількості випадків. Тобто ця метрика буде використовуватись для підбору кращої навченої моделі.

Переносимо цей модуль на полотно та підключаємо його до модуля Split Data (Поділ даних) та до алгоритму Two-Class Boosted Decision Tree (Двокласове збільшуваче дерево прийняття рішень), результат бачимо на рис. 2.14.

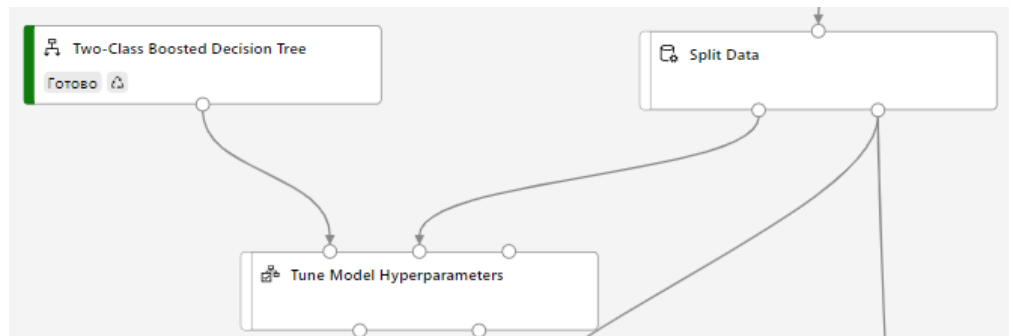


Рисунок 2.14 – Результат підключення модуля Tune Model Hyperparameters

Перед тим як оцінювати модель, використаємо ще один модуль Permutation Feature Importance (Важливість функції перестановки). Це алгоритм, який обчислює оцінки важливості для кожної зі змінних ознак набору даних. Міри важливості визначаються шляхом обчислення чутливості моделі до випадкових перестановок значень ознак. Іншими словами, оцінка важливості кількісно визначає внесок певної ознаки в прогнозовану ефективність моделі з точки зору того, наскільки відхиляється обрана метрика оцінки після перестановки значень цієї функції[34]. Переміщуємо цей модуль на полотно та підключаємо до нього один порт до Tune Model Hyperparameters та інший до Split Data.

Для оцінки нашої моделі буде використано модуль Score Model (Оцінка моделі). Перетягуємо його на полотно конструктора та область вихідних даних модуля Tune Model Hyperparameters підключаємо до вхідних даних модуля Score Model (ліворуч) та результативний набір даних модуля Split Data до області вхідних даних Score Model (праворуч).

Тепер права частина моделі виглядає наступним чином (рис.2.15):

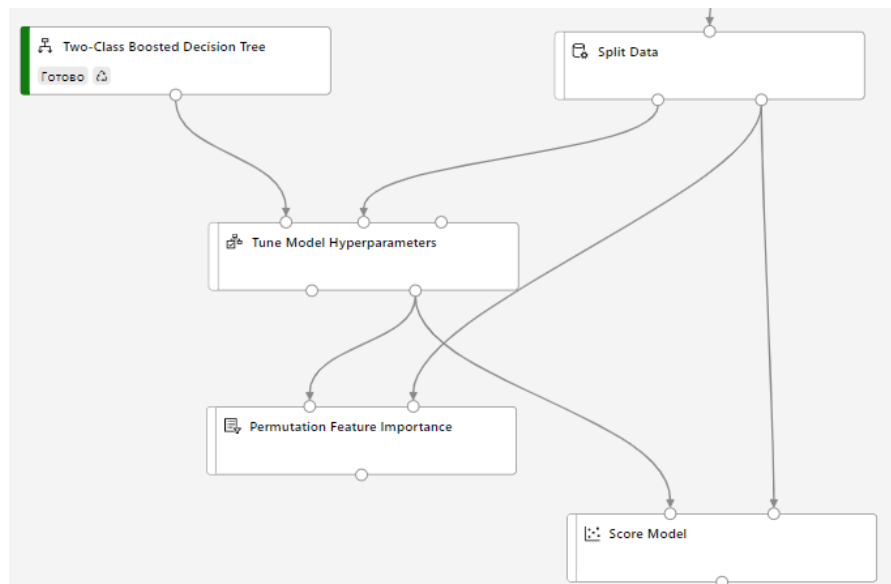


Рисунок 2.15 – Підключення усіх потрібних модулів для навчання та оцінки

Другий спосіб навчання моделі буде використовувати також модуль двокласового збільшуваного дерева прийняття рішень, але з іншими параметрами. Аналогічно до першого способу, підключимо та налаштуємо модуль Split Data (Поділ даних). Далі перетягуємо на полотно конструктора Two-Class Boosted Decision Tree (Двокласове збільшуване дерево прийняття рішень) та переглянемо параметри алгоритму (рис.2.16).

Two-Class Boosted Decision Tree

Обновить

Параметры

Выходные данные и журналы

Подробные сведения

Метрики

...

Create trainer mode *

SingleParameter

Maximum number of leaves per tree *

20

Minimum number of samples per leaf node *

10

Learning rate *

0.2

Number of trees constructed *

100

Random number seed

2

Рисунок 2.16 – Параметри налаштування алгоритму Two-Class Boosted Decision Tree

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Для параметра Create trainer mode (Створити режим вчителя) обраний варіант SingleParameter (Одиночний параметр) і далі був застосований ряд інших аргументів. Навчати цю модель будемо за допомогою модуля Train Model (Навчання моделі), оскільки для алгоритму обраний одиничний параметр і він не потребує начання за допомогою гіперпараметрів. Також підключимо модуль Permutation Feature Importance (Важливість функції перестановки) для оцінки кожної функції моделі. Тепер ліва частина моделі виглядатиме наступним чином (рис.2.17):

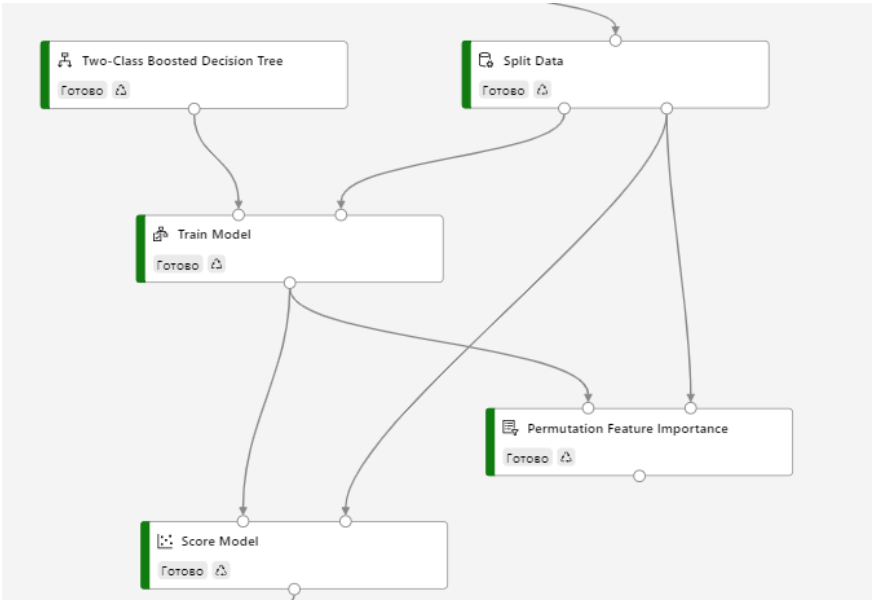


Рисунок 2.17 – Підключення усіх потрібних модулів для навчання та оцінки

2.4 Оцінка та аналіз якості моделі

Модель готова, тому запускаємо конвеєр навчання. Після виконання перейдемо до етапу оцінки моделі. Для початку переглянемо результат модуля Score Model на прикладі лівої моделі. Натискаємо на нього та на панелі властивостей обираємо - Візуалізація.

Прокрутмо екран вправо і побачимо, що поруч зі стовпцем DiagnosisOfHeartDisease (який містить відомі значення істинності для мітки) відображається новий стовпець з ім'ям Scored Labels (Мітки з оцінкою), де

містяться спрогнозовані значення міток, а також стовпець Scored Probabilities (Вірогідність з оцінкою), де міститься значення ймовірності в діапазоні від 0 до 1. Воно вказує ймовірність позитивного прогнозу, і таким чином ймовірність вище 0,5 дає спрогнозовану мітку 1 – (є хвороба), а ймовірності в діапазоні від 0 до 0,5 дають спрогнозовану мітку – 0 (немає хвороби). Результат можемо побачити на рисунку 2.18:

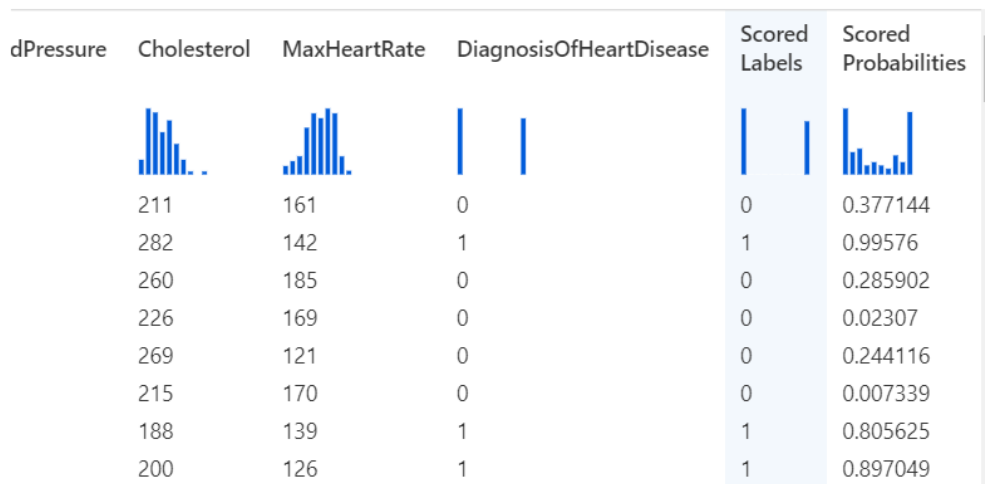


Рисунок 2.18 – Спрогнозовані значення міток

Для кращого представлення моделі було використано модуль Permutation Feature Importance. Враховуючи навчену модель, тестовий набір даних та метрику оцінки, модуль «Важливість функції перестановки» створює випадкову перестановку стовпчика функцій (перетасовує значення цього стовпця) і оцінює продуктивність моделі введення для зміненого набору даних. Враховуючи навчену модель, тестовий набір даних та метрику оцінки, модуль важливості перестановки функції створює випадкову перестановку стовпця ознак (перемішує значення цього стовпця) та оцінює ефективність вхідної моделі для зміненого набору даних. Це робиться ітеративно для кожного зі стовпців даних. Потім модуль повертає список змінних функцій та їх відповідні оцінки важливості. Оцінка важливості визначається як зниження продуктивності після перетасовки значень функцій. З лівої моделі (рис.2.19) ми бачимо, що «NumOfMajorVesselsCBF», «ChestPainType» та «Oldpeak» мають відносно високий показник важливості характеристик.

Permutation Feature Importance визуализация результата

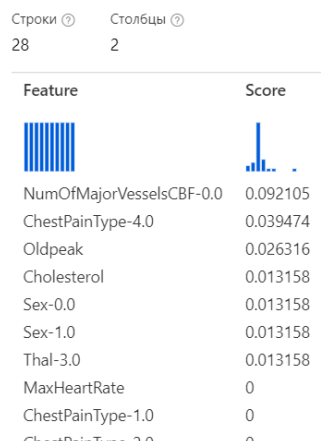


Рисунок 2.19 – Візуалізація результату модуля Permutation Feature Importance (ліва модель)

Коли ми перетворили категоріальні значення в фіктивні, ми дійсно можемо отримати більше інформації, подивившись на оцінки важливості характеристик правильної моделі. Тут ми також знаходимо «Oldpeak» з найвищим балом, поступаючись місцем «ChestPainType-4.0», що означає безсимптомний біль у грудях. Ця змінна типу болю в грудях виникла не при використанні тільки категоріальних змінних, так що іноді потрібно дійсно допомогти перетворити їх, щоб отримати більше інформації.

Якщо порівнювати з правою моделю, то результати трохи відрізняються. Для правої моделі (рис.2.20) «NumOfMajorVesselsCBF», «ChestPainType» також мають високі значення, але «Oldpeak» має значання 0 і це означає, що функція має низький вплив на дану модель.

Дані перевірки, які ми залишили для оцінки моделі, включають відомі значення для мітки. Таким чином, щоб перевірити модель, можна порівняти істинні значення для мітки зі значеннями міток, які були спрогнозовані при оцінці перевіркового набору даних. Грунтуючись на цьому порівнянні, можна обчислити різні метрики, які описують працездатність і продуктивність моделі. Модуль Score Model показує тільки спрогнозовані мітки, але не дає можливість детально проаналізувати модель.

Permutation Feature Importance візуалізація результату

Строки ?	Столбцы ?
28	2
Feature	Score
	
ChestPainType-4.0	0.039474
NumOfMajorVesselsCBF-0.0	0.039474
Cholesterol	0.013158
Sex-0.0	0.013158
Sex-1.0	0.013158
BloodSugar-1.0	0.013158
Exercise-inducedAngina-1.0	0.013158
NumOfMajorVesselsCBF-2.0	0.013158
Thal-3.0	0.013158

Рисунок 2.20 – Візуалізація результату модуля Permutation Feature Importance(права модель)

Для цього використаємо модуль Evaluate Model(Аналіз моделі). Підключимо область виведення модуля Оцінка моделі до області введення модуля Аналіз моделі (рис.2.21).

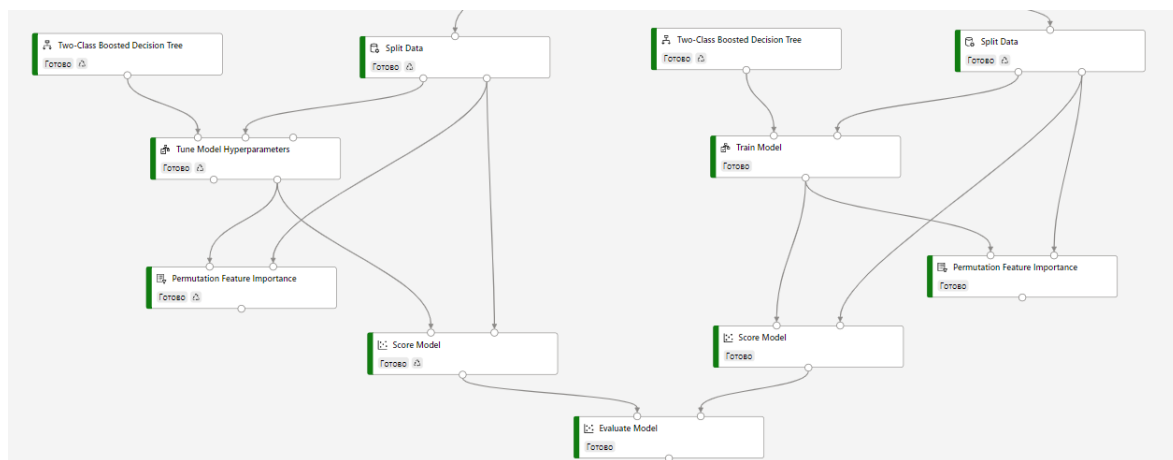


Рисунок 2.21 – Підключення модуля Evaluate Model

Після закінчення виконання експеремнту, натискаємо на модуля Evaluate Model та обираємо пункт Візуалізація. Для оцінки моделі студія машинного навчання використовує такі метрики оцінки, як акуратність, точність, повнота, оцінка F1, матриця неточностей, ROC-крива та AUC-площина (рис.2.22-23).

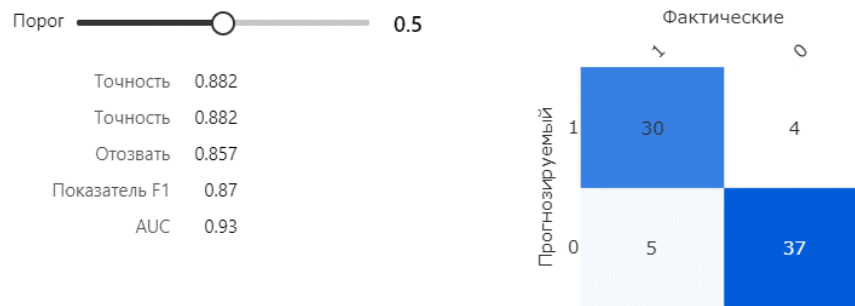


Рисунок 2.23 – Метрики для лівої моделі



Рисунок 2.24 – Метрики для правої моделі

Розглянемо кожну більш детально.

Акуратність (Accuracy). Пропорція точних прогнозів по відношенню до загальної кількості прогнозів, тобто це ймовірність того, що клас буде передбачений правильно (3.1).

Формула виглядає так:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

Істинно-позитивні результати (TP) – це випадки, коли модель правильно передбачила позитивний клас.

Істинно-негативні (TN) – це випадки, коли модель правильно передбачила негативний клас.

Хибно-позитивні (FP) – це випадки, коли модель невірно передбачила позитивний клас.

Хибно-негативні (FN) – це випадки, коли модель невірно передбачила негативний клас.

Хоча акуратність є швидким та інформативним індикатором продуктивності моделі, ми не можемо покладатися виключно на неї. Це пов'язано з тим, що вона приховує наявність зсуву в моделі, що є звичайним явищем, якщо набір даних незбалансований, тобто негативних моментів значно більше, ніж позитивних, або навпаки.

Точність (Precision). Частка позитивних варіантів, які згідно з прогнозами будуть вірні(3.2).

Формула виглядає так:

$$Precision = \frac{TP}{TP + FP} \quad (3.2)$$

Повнота (Recall). Частка варіантів, класифікованих як позитивні, які насправді виявилися позитивними(3.3).

Формула виглядає так:

$$Recall = \frac{TP}{TP + FN} \quad (3.3)$$

Точність і повнота можуть бути візуалізовані на графіку з точністю по осі Y і повнотою по осі X (рис.2.25).

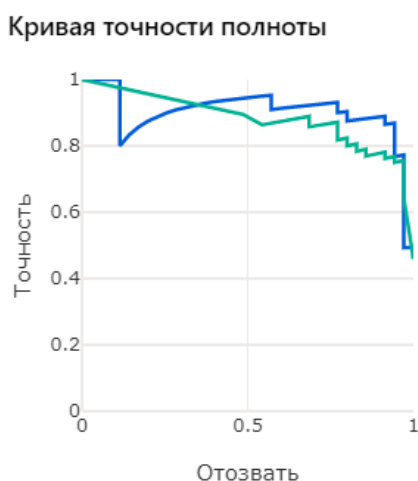


Рисунок 2.25 – Графік точності повноти

Оцінка F1(F-score). Загальна метрика, яка об'єднує в собі інформацію про повноту і точність алгоритму(3.4).

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

Формула виглядає так:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3.4)$$

Матриця неточностей. Це матриця розміром N на N, де N – кількість класів яка представляє собою табличне представлення прогнозованих і фактичних значень для кожного можливого класу(35).

Матриця неточностей, одна з наважливіших речей, на яку потрібно дивитися при оцінці моделі класифікації. Вона містить велику кількість ідей для такої невеликої таблиці. Це матриця, яка візуалізує кількість фактичних екземплярів класу в порівнянні з прогнозованими екземплярами класу. Це дозволяє нам швидко побачити кількість правильних і неправильних прогнозів для кожної категорії.

Над списком метрик є повзунок Порог. Як відомо, модель класифікації прогнозує лише ймовірність для кожного можливого класу. При використанні такої моделі бінарної класифікації прогнозована ймовірність для прогнозу позитивного результату (тобто, наявності серцевої хвороби) становить від 0 до 1. За замовчуванням прогнозована ймовірність хвороби понад 0,5 призводить до прогнозу класу зі значенням 1, а прогноз, що не досягає цього порога, означає, що існує підвищена ймовірність того, що у пацієнта немає серцевої хвороби, тобто прогнозований клас матиме значення 0. Якщо спробувати змістити повзунок, то побачимо як зміниться матриця неточностей (рис.2.26).

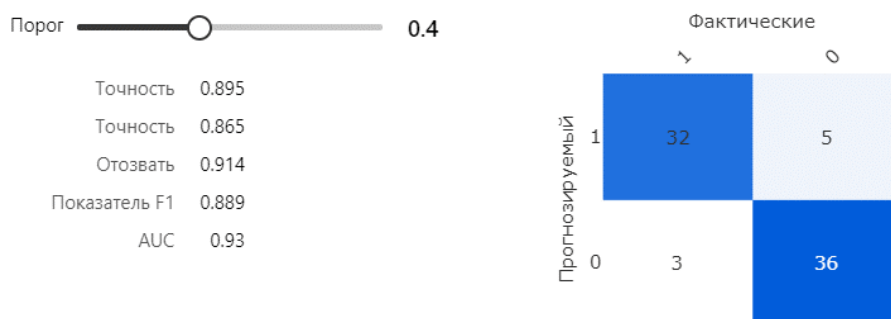


Рисунок 2.26 – Зміна порогу

При зміні порогу, точність моделі збільшується.

Для наочної оцінки якості алгоритму застосовується ROC-крива. Крива будується на площині, визначеної осями повноти по вертикалі і частоти помилково позитивних результатів по горизонталі (рис.2.26).

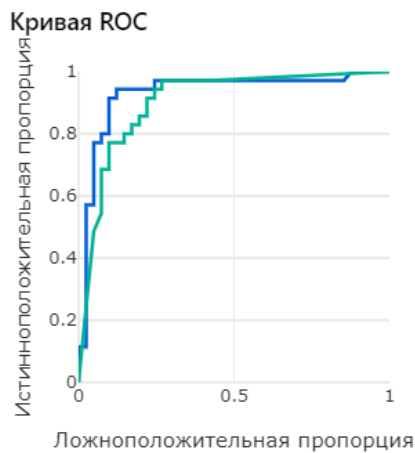


Рисунок 2.26 – Графік ROC-крива

Графік ROC допомагає прийняти рішення про те, де встановити поріг класифікації, щоб максимізувати істинно позитивний рівень або мінімізувати псевдопозитивний показник, що в кінцевому підсумку є бізнес-рішенням.

Цей графік найбільш корисний, коли набір даних збалансований, тобто однакова кількість позитивних і негативних результатів. Ідеальний графік повинен розташовуватися близько до верхнього лівого кута діаграми.

За кривою ROC можна розрахувати ще один важливий показник оцінки: площа під кривою (AUC). AUC буквально вимірює площу під кривою ROC. У нашому випадку значення AUC становить 0,93 лівої моделі та 0,89 правої, отже, 93% та 89% площі графіка знаходиться нижче кривої. Чим більше площа під кривою ROC, тим ближче значення AUC до 1 і тим краще модель розділяє класи. Модель, яка працює не краще, ніж випадковий випадок, матиме AUC 0,5. AUC - хороший загальний огляд передбачуваної сили класифікатора, особливо коли набір даних незбалансований.

Проаналізувавши дані метрики, можна сказати, що алгоритм Two-Class Boosted Decision Tree з діапазоном параметрів і використанням гіперпараметрів, має вищу оцінку, ніж алгоритм з одиночними параметрами.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		60

РОЗДІЛ 3 ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ТА ЕФЕКТИВНОСТІ РІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ

3.1 Розгортання моделі у вигляді кінцевого продукту

Після створення і запуску конвеєра для навчання моделі необхідний другий конвеєр, який виконує ті ж перетворення для нових даних, а потім використовує навчену модель для прогнозування значень міток на основі його функцій. Він стане основою для прогнозної служби, яку можна опублікувати для використання в додатках. Але перед тим, як розгорнути модель у вигляді кінцевого продукту, потрібно зробити декілька кроків.

Перший крок – створення конвеєру виводу в режимі реального часу. Для цього зверху над полотном у списку Створення конвеєра виведення вибираємо Конвеєр виведення в режимі реального часу. Через кілька секунд відкриється нова версія конвеєра з ім'ям Навчання моделі даних про хвороби серця - вивід в реальному часі. Змінюємо ім'я на Прогнозування хвороб серця та проаналізуємо новий конвеєр (рис.3.1).

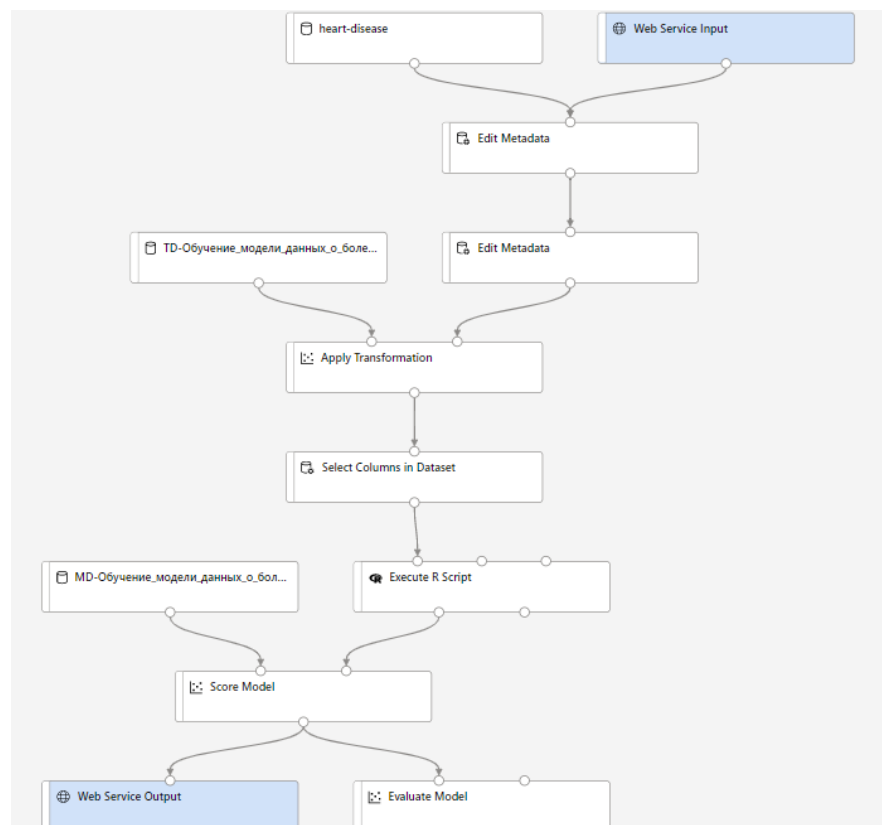


Рисунок 3.1 – Конвеєр в режимі реального часу

Конвеєр містить вхідні дані веб-служби для відправлених нових даних, а також вихідні дані веб-служби для повернення результатів. Деякі перетворення і навчальні дії були вміщені в цьому конвеєрі, тому статистика з навчальних даних буде використовуватися для нормалізації нових значень даних, а навчена модель буде використовуватися для оцінки нових даних.

При використанні конвеєра виведення передбачається, що нові дані відповідають схемі вихідних даних навчання, тобто набір даних heart-disease з конвеєра навчання включений. Однак ці вхідні дані містять мітку DiagnosisOfHeartDisease, яка не включається інтуїтивно в дані нового пацієнта, для якого ще не сформований прогноз, щодо хвороби серця. Видаляємо цей модуль і замінимо набір даних heart-disease на Enter Data Manually (Введіть дані вручну), де наведемо такі дані в форматі CSV (рис. 3.2), в тому числі значення компонентів без міток для спостережень за трьома новими пацієнтами:

Enter Data Manually

Data format ⓘ *

CSV

Has header ⓘ *

True

Data ⓘ *

```
1 Age,Sex,ChestPainType,RestingBloodPressure,Cholesterol,BloodSugar,ElectrocardiographicRes,MaxHeartRate
2 55.0,0.0,4.0,128.0,205.0,0.0,1.0,130.0,1.0,2.0,2.0,1.0,7.0
3 35.0,1.0,2.0,122.0,192.0,0.0,0.0,174.0,0.0,0.0,1.0,0.0,3.0
4 61.0,1.0,4.0,148.0,203.0,0.0,0.0,161.0,0.0,0.0,1.0,1.0,7.0
```

Рисунок 3.2 – Введення нових даних вручну

Вставимо модуль Execute Python Script (Виконання сценарію Python) перед областю вихідних даних веб-служби, щоб повертати тільки вік пацієнта, прогнозоване значення мітки і ймовірність (рис.3.3).

```
import pandas as pd
```

```
def azureml_main(dataframe1 = None, dataframe2 = None):
```

```
    scored_results = dataframe1[['Age', 'Scored Labels', 'Scored Probabilities']]
    scored_results.rename(columns={'Scored Labels':'DiagnosisOfHeartDisease',
                                   'Scored Probabilities':'Probability'},
                           inplace=True)
    return scored_results
```

Рисунок 3.3 – Лістинг коду модуля Execute Python Script

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

Видаляємо ще непотрібні модулі, такі як Edit Metadata (для змін назв стовпців) та Execute R Script. Тепер модель виглядає наступним чином (рис.3.4):

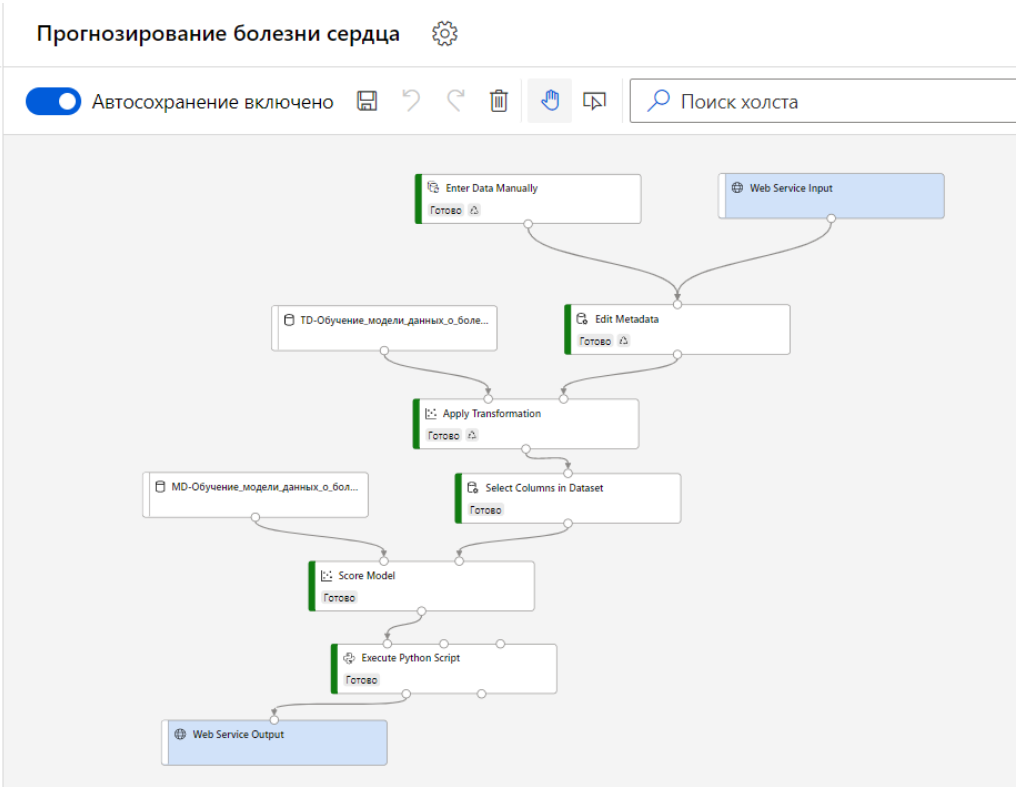


Рисунок 3.4 – Конвеєр в режимі реального часу

Після завершення роботи конвеєра вибираємо модуль Execute Python Script і в області параметрів, на вкладці Вихідні дані і журнали, візуалізують Результуючий набір даних, щоб переглянути спрогнозовані мітки і ймовірності в області вихідних даних для спостережень за трьома пацієнтами (рис.3.5).

Age	DiagnosisOfHeartDisease	Probability
55	1	0.967615
35	0	0.015533
61	1	0.840014

Рисунок 3.5 – Вірно спрогнозовані мітки для трьох пацієнтів

Після створення і тестування конвеєра виведення в режимі реального часу можна опублікувати його як службу і використовувати в клієнтських додатках.

Наступним кроком буде розгортання служби. Тут ми розгорнемо веб-службу в створеному раніше екземплярі контейнера Azure (ACI). Цей тип обчислень створюється динамічно і він корисний для розробки і тестування. Повертаємось до конвеєра виведення в режимі реального часу. У правому верхньому кутку вибираємо Розгорнути, після чого розгорнемо нову кінцеву точку для прогнозування в реальному часі (рис.3.6).

Настройка конечной точки для прогнозирования в реальном времени

☒ Развертывание новой конечной точки для прогнозирования в реальном времени
 ☐ Замена существующей конечной точки для прогнозирования в реальном времени

Имя *

predict-heart-disease

Описание

Классификация диагноза "болезнь сердца"

Тип вычисления *

Экземпляр контейнера Azure

[Дополнительно](#)

[Развернуть](#)
[Отмена](#)

Рисунок 3.6 – Налаштування розгортання кінцевої точки

Тепер можна протестувати розгорнуту службу з клієнтського додатку. В цьому випадку код в нижче використовується для імітації клієнтського додатку.

На сторінці Кінцеві точки відкриваємо кінцеву точку в реальному часі predict-heart-disease. Перейдемо на вкладку Використання і переглянемо дані, які необхідні для розгортання кінцевої служби з клієнтського додатку (рис.3.7).

predict-heart-disease

[Подробности](#)
[Тест](#)
[Использование](#)
[Журналы развертывания](#)

Сведения о базовом потреблении

Конечная точка REST

http://12468fea-6f40-4c5c-9a9b-c48676946140.westeurope.azurecontainer.io/score

Проверка подлинности

Первичный ключ
8c1zYSeuW3GxRUwvF1wlskcEr6rqNnqE [Создать повторно](#)

Вторичный ключ
TqTPrcov3f8CXY8imyXR9jzK7q7cumfh [Создать повторно](#)

Рис. 3.7. Кінцева точка в реальному часі predict-heart-disease

		Побережна М.О.			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

На вкладці діалогового вікна обираємо пункт Записні книжки і в розділі Мої файли створюємо новий файл «Test-Heart-Disease»

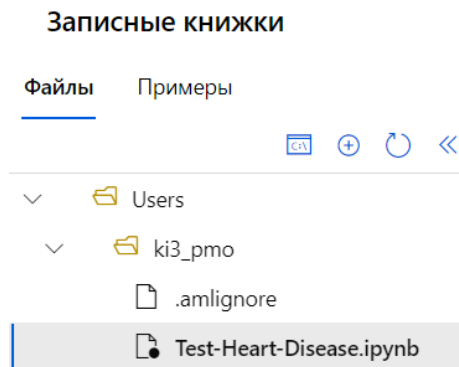


Рисунок 3.8 – Створений файл «Test-Heart-Disease»

Вставляємо код у файл та змінюємо потрібні нам ключі. Далі вводимо дані пацієнта для якого потрібно спрогнозувати мітку та запускаємо код. Повний код файлу наведений в додатку А.

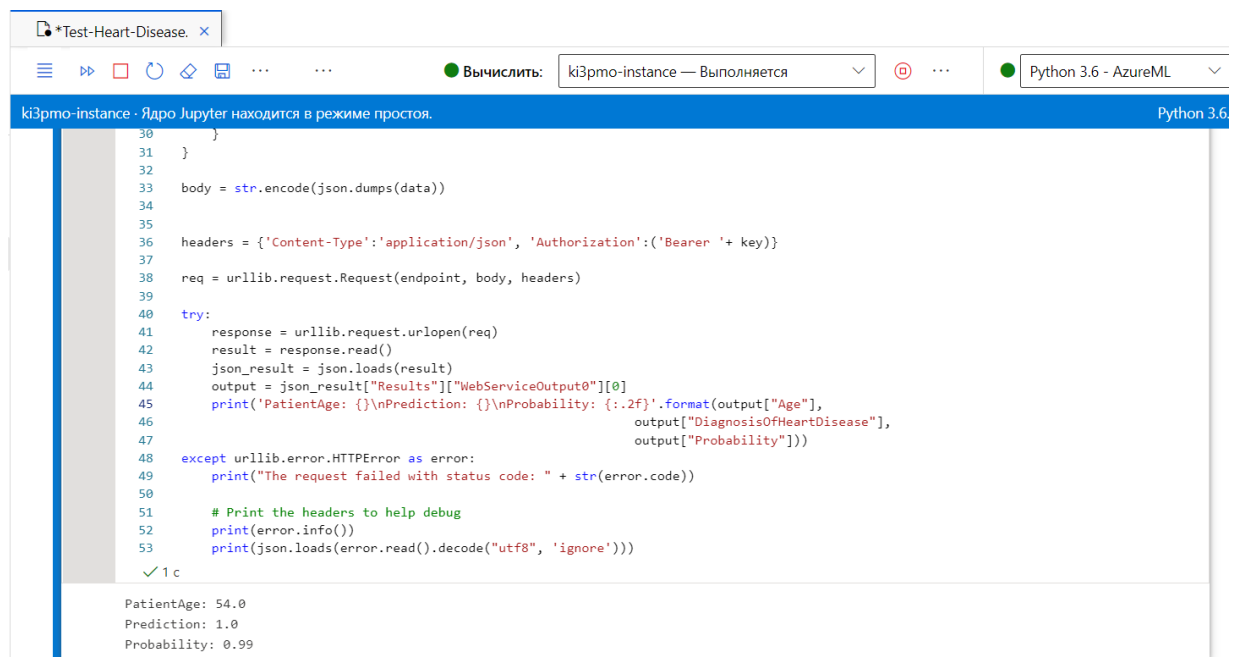


Рисунок 3.9 – Правильно спрогнозована мітка для пацієнта

3.2 Дослідження та порівняння алгоритмів прогнозування студії машинного навчання Azure

Наша розроблена модель з використанням алгоритму Двокласове збільшуваче дерево прийняття рішень дає хороші результати. Однак розглянемо ще інші

		Побережна М.О.			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

алгоритми класифікації та порівняємо їх ефективність. Дослідивши роботу різних алгоритмів, можна покращити власну модель класифікації.

Створюємо конвеєр Порівняння алгоритмів класифікації, як це було описано в другому розділі. Обираємо вже створений кластер, який буде використовуватись для навчання моделі. Коли повністю готове для конструювання моделі, нам потрібно виконати наступних декілька кроків.

1. Завантаження даних та їх попередня обробка.

Перетягуємо модуль, раніше створеного набору даних heart-disease на полотно. Проводимо підготовку даних. Оскільки дані не мають назв стовпців, під'єднаємо модуль Edit Metadata(Редагування метаданих) та створимо назви. Далі використаємо такий же модуль, але з іншими параметрами, а саме відзначимо вибрані стовпці як категоріальні. Щоб краще оцінити модель, перетворимо категоріальні стовпці у фіктивні, використовуючи модуль модуля Convert to Indicator Values. Заберемо лишні стовпці за допомогою модуля Select Columns in Dataset. Також нам знадобиться модуль Execute R Script. За допомогою цього скрипту ми змінимо значення в стовпці «DiagnosisOfHeartDisease» на бінарні, тобто 0 та 1.

2. Розробка та навчання моделі.

Для навчання моделі і порівняння продуктивності буде використано два двокласових алгоритми: логістична регресія та метод опорних векторів. Для алгоритмів будемо застосовувати одиничні параметри за замовчуванням.

Також необхідно поділити дані, використовуючи модуль Split Data (Поділ даних). Набір даних для навчання буде містити 75% вхідних даних, а для тестування – 25%.

Алгоритм та навчальний набір даних підключаємо до модуля Train Model (Навчання моделі), який буде навчати модель.

3. Оцінка, аналіз та порівняння якості моделі.

Після завершення навчання, переходимо до етапу оцінки. Для цього використовуємо модуль Score Model. В ньому ми можемо побачити спрогнозовані мітки та вірогідність з оцінкою. Для глибшого аналізу та порівняння

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

застосовується модуль Evaluate Model, куди передаються дані для створення показників оцінки. Тепер, коли всі модулі підключені, модель виглядає наступним чином (рис.3.10):

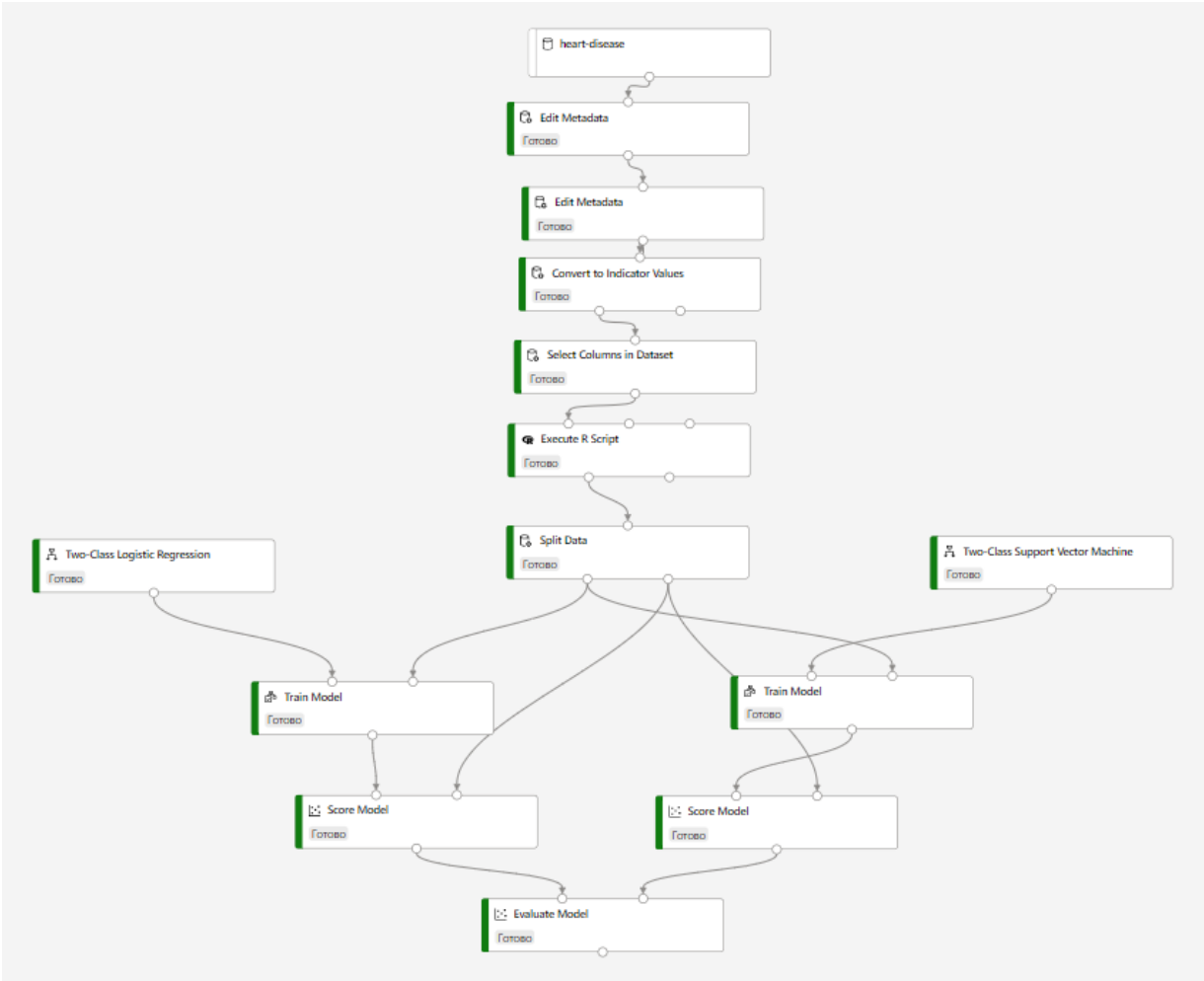


Рисунок 3.10 – Модель для порівняння алгоритмів

Щоб порівняти роботу двох алгоритмів, натискаємо на модуль Evaluate Model та на панелі вхідних та вихідних даних обираємо пункт Візуалізація. Оцінимо ефективність моделей за допомогою метрик. На рисунку 3.11 бачимо, що графіки практично однакові.

Evaluate Model визуализация результата

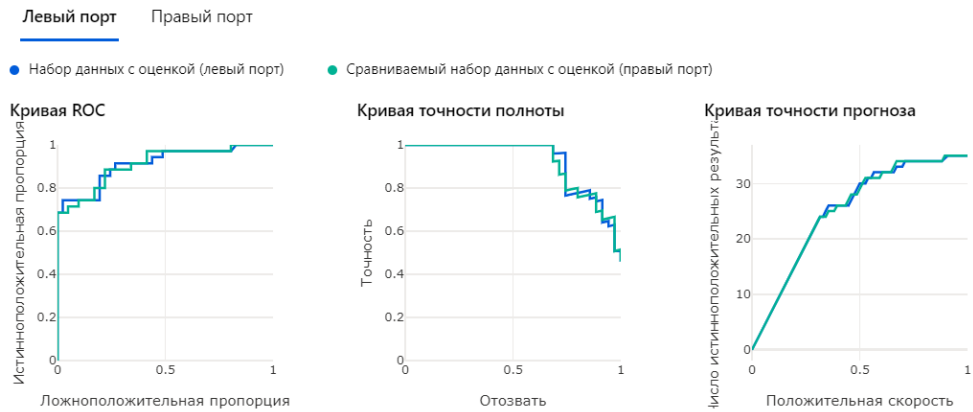


Рисунок 3.11 – Порівняння графіків

Щоб вирішити, яка модель працює краще, порівняємо метрики точності та розглянемо матриці неточностей.

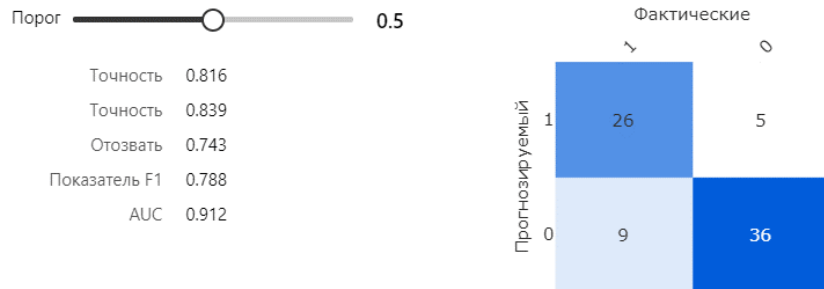


Рис. 3.12. Метрики моделі з алгоритмом логістична регресія

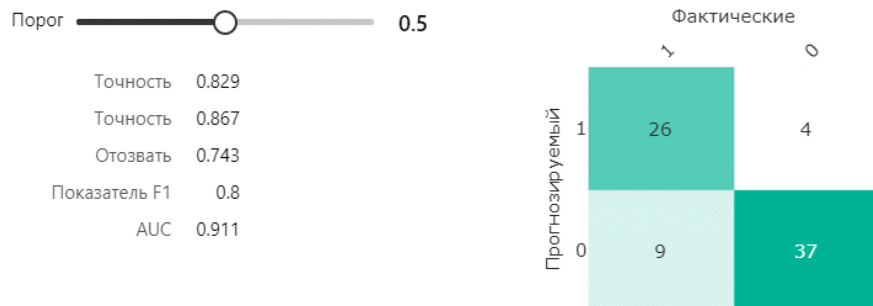


Рисунок 3.13 – Метрики моделі з алгоритмом метод опорних векторів

Проаналізувавши метрики на рисунках 3.12-13, можна зробити висновок, що алгоритм метод опорних векторів має вищі результати, чим логістична регресія. Тому модель побудована з використанням цього алгоритму матиме вищу точність.

3.3 Рекомендації по застосуванню розробленої інформаційної технології

У кваліфікаційній бакалаврській роботі розроблено технологію, яка вирішує задачі класифікації медичних даних. Після вдалого розгортання та тестування моделі в режимі реального часу, можна застосовувати нашу модель машинного навчання у сфері медичної діагностики.

Машинне навчання в інформатиці здоров'я дозволяє набагато швидше аналізувати генетичні мутації і допомагає діагностувати стани, які можуть привести до захворювань. Використовуючи типи розширеної аналітики, ми можемо надати лікарям більш точну інформацію при наданні допомоги пацієнтам. Процес діагностики може пришвидшитись, якщо лікарям надавати інформацію про ризик інсульту пацієнта, ішемічної хвороби серця або ниркової недостатності на основі останніх 50 свідчень артеріального тиску, результатів лабораторних аналізів, раси, статі і дані останніх клінічних випробувань.

Кардіологія - одна з галузей медицини, що є найбільш цікавою для застосування розробленої технології. Тому її впровадження у цю сферу є актуальним. Розроблена технологія може надавати більше інформації клініцистам, щоб вони могли приймати більш обґрунтовані рішення щодо діагнозів пацієнта і варіантів лікування, розуміючи при цьому можливі результати і вартість кожного з них. Технологія може позитивно вплинути на прогнозування і діагностику серцево-судинних захворювань за рахунок розробки алгоритмів, які можуть моделювати представлення даних набагато швидше і ефективніше, ніж це може зробити лікар. А також цінність полягає в її здатності обробляти величезні набори даних, що виходять за рамки людських можливостей, а потім надійно перетворювати аналіз цих даних в клінічні дані, які допомагають лікарям у плануванні та наданні допомоги, що в кінцевому підсумку призводить до кращих результатів і зниження витрат на лікування, а також турботи і підвищення задоволеності пацієнтів.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		69

Наприклад, в даний час лікар, який хоче спрогнозувати повторну госпіталізацію пацієнта із застійною серцевою недостатністю, повинен переглянути великий, але неструктурований набір даних електронної історії хвороби, який включає такі змінні, як коди виставлення рахунків Міжнародної класифікації хвороб, рецепти на ліки, лабораторні показники, фізіологічні вимірювання, візуалізаційні дослідження. Такий набір даних надзвичайно ускладнює вирішення питань, які змінні повинні бути включені в прогнозну модель і які методи слід застосовувати в самій моделі. Тому машинне навчання стає тут у нагоді.

Ми використовуємо простий приклад, щоб продемонструвати теорію і практику машинного навчання клініцистам і дослідникам-медикам. Принципи, які ми демонструємо у нашій технології, можуть бути легко застосовані до інших складних завдань, включаючи обробку природної мови, для розуміння неструктурованих текстових даних, і розпізнавання зображень. Зокрема, МН може бути використане для додатків візуалізації серця, таких як автоматичне обчислення балів, диференціація прогностичних фенотипів, кількісна оцінка функції серця і сегментація серця. Ці інструменти також продемонстрували здатність виконувати раннє і точне виявлення аномалій при електрокардіографічних дослідженнях. Алгоритми машинного навчання також можуть сприяти оцінці ризику серцево-судинних захворювань в різних умовах і виконувати прогнозування серцево-судинних подій. Ще один цікавий напрям досліджень в цій області - геномна оцінка серцево-судинних захворювань.

Таким чином, МН може допомогти в більш ранній діагностиці захворювання, розробці індивідуалізованих методів лікування і виявленні прогностичних характеристик при різних патологічних станах, що призведе до точної кардіології.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У кваліфікаційній бакалаврській роботі було вирішено актуальне завдання по розробці хмарного рішення задачі класифікації медичних даних на основі конструктора машинного навчання Azure.

Обґрунтування актуальності теми дало змогу оцінити внесок машинного навчання у різні галузі людства, зокрема у сферу охорони здоров'я.

Після проведення аналізу МН, був обраний метод контрольованого машинного навчання, а саме метод класифікації, для вирішення поставленої задачі кваліфікаційної роботи. Аналіз переваг та недоліків кожного методу класифікації дав змогу обрати кращий алгоритм навчання нашої моделі, зокрема алгоритм дерева прийняття рішень.

З метою вирішення проблеми прогнозування серцево-судинних захворювань було обрано хмарну платформу Microsoft Azure з сервісом Azure Machine Learning. Azure Studio має засоби для дослідження даних, попередньої обробки, вибору методів і перевірки результатів моделювання. Конструктор машинного навчання Azure забезпечує доступ до функцій платформи і управління ними.

Ознайомившись з середовищем розробки, було виділено етапи розробки моделі в конструкторі машинного навчання Azure: створення робочої області Azure, створення конвеєру, імпортування даних, підготовка даних, навчання та оцінка моделі машинного навчання, розгортання моделі у вигляді кінцевого продукту прогнозування в реальному часі.

Готова модель прогнозування серцево-судинних захворювань включає модулі: набір даних, два модулі редагування даних, вибір стовпців для навчання, виконання R скрипта, поділ даних, двокласове збільшуване дерево прийняття рішень з одиничним параметром та діапазоном параметрів, налаштування гіперпараметрів моделі, навчання моделі, оцінка та аналіз моделі.

Дана модель розроблена за допомогою алгоритму контрольованого машинного навчання, а саме класифікації, оскільки результат виконання моделі є мітка 0 – не виявлено хворобу, 1 – виявлено. Етап навчання і тестування моделі

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

використовує набір даних «heart-disease»(серцеві захворювання). Оцінка і аналіз моделі показало кращі результати прогнозування з використанням алгоритму збільшуваче дерево прийняття рішень з діапазоном параметрів та гіперпараметрами. Тому ця модель була обрана для розгортання моделі в режимі реального часу. Оскільки модель працює тільки в Студії машинного навчання, розгортання дає змогу протестувати модель як додаток для інших користувачів. Тестування показало працездатність моделі та готовність застосування у реальних додатках.

Також було проведено дослідження та порівняння моделі з використанням двокласових алгоритмів: логістична регресія та метод опорних векторів. Результат дослідження показав, що модель з алгоритмом логістичної регресії має вищі показники метрик, ніж метод опорних векторів.

Розглянуто доцільність використання запропонованої моделі у сфері охорони здоров'я. Основними перевагами є обробка великої кількості даних, миттєве прогнозування та здійснення аналізу даних в режимі реального часу. Використання нашої моделі прогнозування серцево-судинних захворювань дає змогу завчасно виявити хворобу, що допоможе лікарям негайно приймати рішення щодо лікування пацієнта.

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Флах. П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных/ пер. с англ. А. А. Слинкина. - М.: ДМК Пресс, 2015. - 400 с.
2. Мерков А. Б. Распознавание образов. Введение в методы статистического обучения. 2011. 256 с.
3. Top 10 real-life examples of Machine Learning. URL: <https://bigdata-madesimple.com/top-10-real-life-examples-of-machine-learning/>
4. Schmidt, J., Marques, M.R.G., Botti, S., Miguel A. L. Marques. Recent advances and applications of machine learning in solid-state materials science. npj Comput Mater 5, 2019. №83 . URL: <https://www.nature.com/articles/s41524-019-0221-0>
5. Daffodil Software. 9 Applications of Machine Learning from Day-to-Day Life, 2017. URL: <https://medium.com/app-affairs/9-applications-of-machine-learning-from-day-to-day-life-112a47a429d0>
6. Machine Learning: What it is and why it matters. URL: https://www.sas.com/en_us/insights/analytics/machine-learning.html
7. Jenni A. M. Sidey-Gibbons & Chris J. Sidey-Gibbons. Machine learning in medicine: a practical introduction. BMC Medical Research Methodology 19, 2019. №64. URL: <https://bmcmmedresmethodol.biomedcentral.com/articles/10.1186/s12874-019-0681-4>
8. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017. №542. p.115–8. URL: <https://doi.org/10.1038/nature21056>
9. 15 examples of machine learning in healthcare that are revolutionizing medicine, 2020. URL: <https://builtin.com/artificial-intelligence/machine-learning-healthcare>

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				73
Змн.	Арк.	№ докум.	Підпис	Дата		

10. Серцево-судинні захворювання — головна причина смерті українців. висновки з дослідження глобального тягаря хвороб у 2019 році. // Центр громадського здоров'я МОЗ України, 2021. URL: <https://phc.org.ua/news/sercevo-sudinni-zakhvoryuvannya-golovna-prichina-smerti-ukrainsiv-visnovki-z-doslidzhennya>
11. Machine Learning Methods. URL: <https://www.educba.com/machine-learning-methods/>
12. What are some popular machine learning methods? URL: https://www.sas.com/en_us/insights/analytics/machine-learning.html
13. Regression vs. Classification in Machine Learning. URL: <https://www.javatpoint.com/regression-vs-classification-in-machine-learning>
14. Regression Analysis in Machine learning. URL: <https://www.javatpoint.com/regression-analysis-in-machine-learning>
15. 8 Clustering Algorithms in Machine Learning that All Data Scientists Should Know, 2020. URL: <https://www.freecodecamp.org/news/8-clustering-algorithms-in-machine-learning-that-all-data-scientists-should-know/>
16. Clustering in Machine Learning URL: <https://www.javatpoint.com/clustering-in-machine-learning>
17. Classification Algorithms. URL: <https://www.educba.com/classification-algorithms/?source=leftnav>
18. Кравченко С.М, Гришкун Є.О., Власенко О.В.. Методи класифікації машинного навчання з використанням бібліотеки scikit-learn // Науково-технічна інформація. 2020 . №3. с.121-122. URL: http://www.tech.vernadskyjournals.in.ua/journals/2020/3_2020/part_1/21.pdf
19. Rohith Gandhi. Naive Bayes Classifier, 2018. URL: <https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>
20. Decision Tree Classification in Python. URL: <https://www.data-camp.com/community/tutorials/decision-tree-classification-python>

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		74

21. Avinash Navlani. Support Vector Machines with Scikit-learn, 2019.
URL: <https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python>
22. Niklas Donges. A complete guide to the random forest algorithm, 2019.
URL: <https://builtin.com/data-science/random-forest-algorithm#how>
23. Avinash Navlani. KNN Classification using Scikit-learn, 2018. URL: <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>
24. Ayush Pant. Introduction to Logistic Regression, 2019. URL: <https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148>
25. Технические возможности обеспечения безопасности Azure. URL: <https://docs.microsoft.com/ru-ru/azure/security/fundamentals/technical-capabilities>
26. Exploit the ever-expanding power of Azure cloud services. URL: <https://www.avanade.com/en/technologies/azure>
27. Comparing Machine Learning as a Service: Amazon, Microsoft Azure, Google Cloud AI, IBM Watson. URL: <https://www.altexsoft.com/blog/datascience/comparing-machine-learning-as-a-service-amazon-microsoft-azure-google-cloud-ai-ibm-watson/>
28. Хенрик Бринк, Джозеф Ричардс, Марк Феверолф. Машинное обучение: СПб.: Питер, 2017. с. 44-51.
29. How machine learning works. URL: <https://www.ibm.com/en/cloud/learn/machine-learning>
30. Создание прогнозных моделей без кода в Машинном обучении Azure. URL: <https://docs.microsoft.com/ru-ru/learn/modules/create-classification-model-azure-machine-learning-designer/introduction>
31. Создание вычислительных ресурсов. URL: <https://docs.microsoft.com/ru-ru/learn/modules/create-classification-model-azure-machine-learning-designer/create-compute>

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

32. Модуль двухклассового увеличивающегося дерева принятия решений URL: <https://docs.microsoft.com/ru-ru/azure/machine-learning/algorithm-module-reference/two-class-boosted-decision-tree>

33. Настройка гиперпараметров модели. URL: <https://docs.microsoft.com/ru-ru/azure/machine-learning/algorithm-module-reference/tune-model-hyperparameters>

34. Permutation Feature Importance. URL: <https://docs.microsoft.com/ru-ru/archive/blogs/machinelearning/permutation-feature-importance>

35. Evaluating Models in Azure Machine Learning (Part 1: Classification). URL: <https://adatis.co.uk/evaluating-models-in-azure-machine-learning-part-1-classification/>

36. Пулеко І.В., Побережна М.О. Хмарне рішення задачі класифікації медичних даних на основі конструктора машинного навчання Azure. Тези доповідей XII Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології – 2021 (ІКТ-2021)», м. Житомир, 01 - 03 квітня 2021 р. – Житомир: Житомирська політехніка, 2021. с. 83-84.

		Побережна М.О.			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				76
Змн.	Арк.	№ докум.	Підпис	Дата		



ДОДАТКИ

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				77
Змн.	Арк.	№ докум.	Підпис	Дата		

```

endpoint = 'http://12468fea-6f40-4c5c-9a9b-
c48676946140.westeurope.azurecontainer.io/score' #Replace with your endpoint
key = '8c1zYSeuW3GxRUwvF1wlskEr6rqNnqE' #Replace with your key

import urllib.request
import json
import os

data = {
    "Inputs": {
        "WebServiceInput0":
            [
                {
                    'Age': 54,
                    'Sex': 1,
                    'ChestPainType': 4,
                    'RestingBloodPressure': 124,
                    'Cholesterol': 266,
                    'BloodSugar': 0,
                    'ElectrocardiographicRes': 2,
                    'MaxHeartRate': 109,
                    'Exercise-inducedAngina': 1,
                    'Oldpeak': 2.2,
                    'Slope': 2,
                    'NumOfMajorVesselsCBF': 1.0,
                    'Thal': 7.0,
                },
            ],
    },
    "GlobalParameters": {
    }
}

body = str.encode(json.dumps(data))

headers = {'Content-Type': 'application/json', 'Authorization': ('Bearer ' + key)}

req = urllib.request.Request(endpoint, body, headers)

try:
    response = urllib.request.urlopen(req)
    result = response.read()
    json_result = json.loads(result)
    output = json_result["Results"]["WebServiceOutput0"][0]
    print('PatientAge: {}\nPrediction: {}\nProbability: {:.2f}'.format(output["Age"],
        output["DiagnosisOfHeartDis
ease"],
        output["Probability"])))
except urllib.error.HTTPError as error:
    print("The request failed with status code: " + str(error.code))

    # Print the headers to help debug
    print(error.info())
    print(json.loads(error.read().decode("utf8", 'ignore')))
```

		Побережна М.О			КІ.КР.Б. – 123 – 21 – ПЗ	Арк.
		Пулеко І.В.				78
Змн.	Арк.	№ докум.	Підпис	Дата		