**2. АНАЛІЗ МАТЕРІАЛІВ ТА МЕТОДІВ**

Спочатку необхідно підготувати дані і збір датасету, цей етап є найважливішим. Датасет – це основа для будь-якого аналізу даних. Він містить інформацію, яка використовується для роботи з моделями машинного навчання. Тому наявність пропущених значень у датасеті може значно ускладнити аналіз та викликати невизначеність у результатах. Будуть використовуватися існуючі датасети на сайті Kaggle, який містить різні великі набори даних з пропущеними значеннями.

Буде проаналізовано різні підходи для обробки пропущених значень та визначення невизначеності у великих наборах даних, і різні математичні моделі для передбачення цільових стовпців, такі як: дерево рішень, KNN, логістична регресія, Наївний Баєс, підсилення градієнту. В наступному розділі детальніше проаналізуємо датасети, які були використанні у цій курсовій роботі.

Вибір датасетів має великий вплив на метод обробки пропущених значень. І для цього було вибрано пару датасетів, які є суттєво відрізняються один від одного, вони також містять великі набори даних і мають NaN значення.

* 1. **Опис набору даних**

Вибір правильного датасету відіграє важливу роль для обробки пропущених значень та визначення невизначеності у великих наборах даних, оскільки:

1. Датасет відіграє велику роль у обробці NaN значень та визначенні невизначеності, оскільки він є основою для аналізу даних та впливає на точність та достовірність результатів.
2. Дані в датасеті повинні бути якісними і достовірними, він повинен бути без помилок або неточностей, та відображати реальну ситуацію.

Через правильний датасет ми отримаємо найкращу обробку NaN значень, та змогу перевірки найкращих методів для обробки NaN значень.

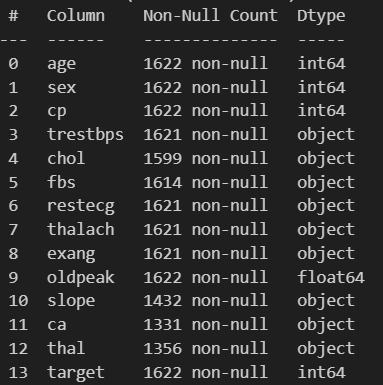
* + 1. **Heart Attack Dataset**

Набір даних складається з трьох менших датасетів. Усі вони знаходиться на платформі Kaggle у відкритому доступі. Цей набір містить дані про стан здоров’я людини, і як ці показники впливають на здоров’я людини.

|  |  |  |  |
| --- | --- | --- | --- |
| Статистика датасету | | Тип змінних | |
| Кількість стовпців | 14 | Категорійні | 9 |
| Кількість змінних | 1622 | Числові | 5 |

**Таблиця 1** Статистика датасету

У таблиці 1 показується скільки у наборі даних є категоріальних і числових стовпців. Разом 3 датасети містять 1622 рядків даних і 14 стовпців, з яких 9 стовпців є категоріальними, а 5 числовими.



**Рисунок 1** Інформація про датасет

Рисунок 1 дає інформацію про тип стовпців, імена стовпців і їх кількість і кількість пропущених значень у кожному стовпці. Кожен стовпець пояснює характеристику людини: її вік, стать, тип болю в грудях, артеріальний тиск, рівень цукру в крові, стани серця, кількість великих судин.

Інформація про значення стовпців у наборі даних пояснено у таблиці 2:

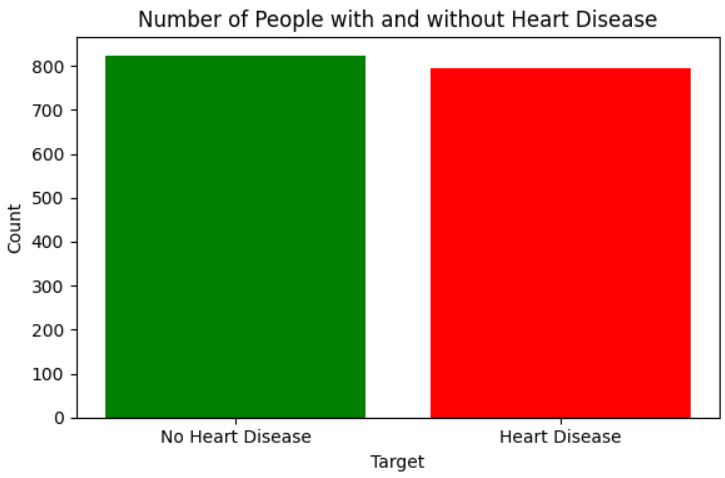
|  |  |
| --- | --- |
| **Ім’я колонки** | **Пояснення значення** |
| Age | Вік |
| Sex | Стать |
| Cp | Тип болю в грудях:  1 – типова стенокардія  2 – атипова стенокардія  3 – неангіальний біль  4 – без симптомний |
| Trbps | артеріальний тиск у стані спокою |
| Chol | холестеральний в мг/дл, отриманий за допомогою датчика ІМТ. |
| Fbs | рівень цукру в крові > 120 мг/дл, де 1 = так; 0 = ні |
| Restecg | Результати електрокардіографії в спокої |
| Thalach | Максимальна досягнута частота серцевих скорочень |
| Exng | стенокардія фізичного навантаження, де 1 = так, 0 = ні |
| Oldpeak | показник зміни електрокардіографічних хвиль після фізичного навантаження |
| Slp | нахил піка СТ-сегменту під час навантаження |
| Caa | кількість великих судин, значення від 1 до 3 |
| Thall | досягнута максимальна частота серцевих скорочень |
| Target | цільова колонка, 0 – немає ймовірності серцевого нападу, 1 = є ймовірність серцевого нападу |

**Таблиця 2** Пояснення характеристик датасету

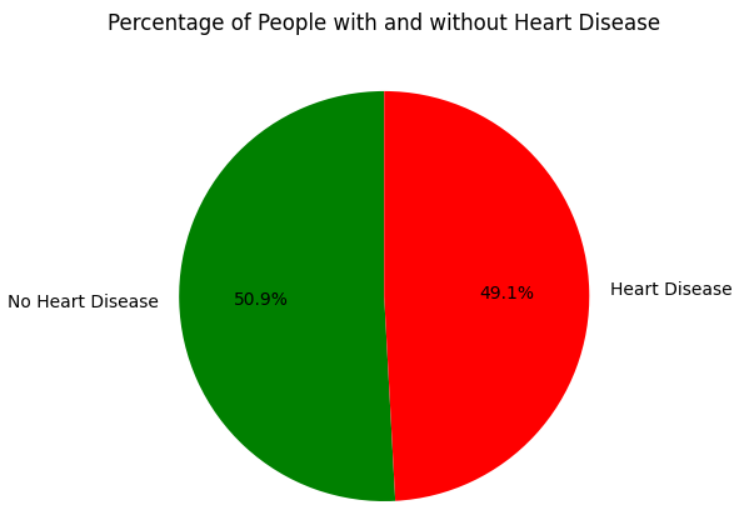
У таблиці 2 показується характеристика стовпців, за що вони відповідають. Датасет є збалансованим, оскільки усі ознаки мають одинаковий вплив на результат.

Цільова колонка цього набору даних є Target, де будемо передбачувати чи можливий у людини інфаркт чи ні. Де 0 – не можливо, а 1 – можливий інфаркт.

У рисунку відображено кількість людей які мали інфаркт і не мали інфаркту.

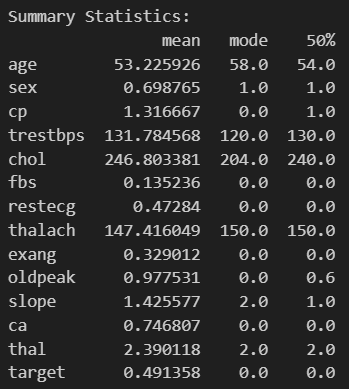


**Рисунок 2** Кількість людей з і без інфаркту, стовпчаста діаграма



**Рисунок 3** Кількість людей з і без інфаркту, кругова діаграма

На рисунку 3 і рисунку 4 показано діаграми, які мають співвідношення людей з інфарктом і без інфаркту. У датасеті є більше людей, які не мають інфаркт. Оскільки їхня частка в датасеті 50.9%. Пацієнти які мають інфаркт у датасеті займають 49.1% всього набору даних.



**Рисунок 4** Опис метрик

На рисунку 4 показано середнє значення, мода і медіана кожного стовпця у датасеті. Як бачимо на Рисунку 4 середній вік людини за середнім значенням досягає 53 роки, за модою 58 років, а за медіаною 54 роки. Також дуже відрізняються такі ствопці один від одного: chol, oldpeak, cp. Стать – переважають 1, тобто чоловіки.

Цей набір даних аналізує фактори ризику серцевого нападу на основі цих медичних даних та допомагає визначити зв'язок між різними параметрами та ймовірністю виникнення серцевого нападу. Також цей датасет не містить пропущені значення.

* + 1. **Covid-19 Dataset**

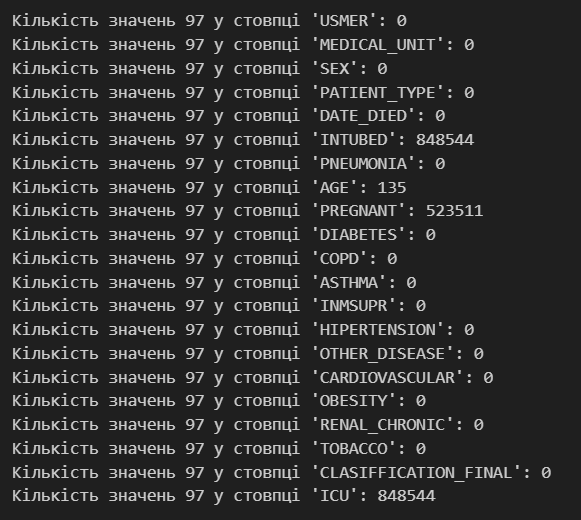
Даний набір даних є у відкритому доступі на сайті Kaggle. Цей датасет містить інформацію про здоров’я людини під час захворювання на коронавірус у 2020 році.

|  |  |  |  |
| --- | --- | --- | --- |
| Статистика датасету | | Тип змінних | |
| Кількість стовпців | 21 | Категорійні | 8 |
| Кількість змінних | 1048575 | Числові | 13 |

**Таблиця 3** Статистика датасету

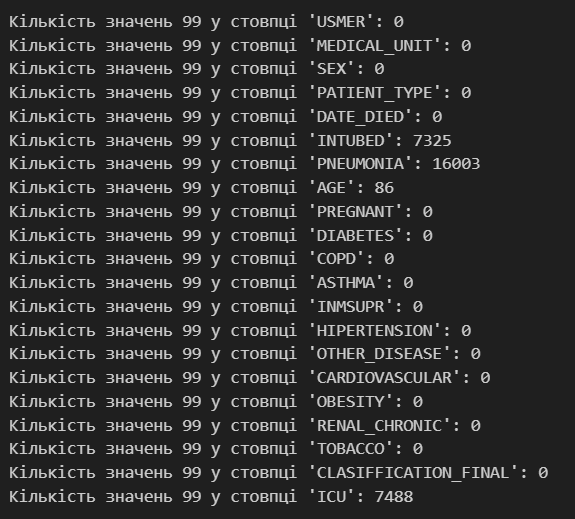
Таблиця 3 дає інформацію скільки у наборі даних є категоріальних і числових стовпців. Набір даних містить 1048575рядків даних і 21 стовпців, з яких 8 стовпців є категоріальними, а 13 числовими. Порівняно з першим датасетом Heart Attack Dataset, він є більш об’ємним в рази, а саме цей датасет більший в 646 разів за попередній. Також даний набір даних має на 7 більше стовпців ніж попередній датасет, але він містить на 1 категоріальний стовпець менше, і на 8 більше числових стовпців.

У датасеті пропущені значення не є NaN, а замінені на числа 97 і 99, а для дати на дату 9999-99-99.



**Рисунок 5** Інформація про датасет

Рисунок 5 демонструє кількість пропущених значень, які замінені на число 97 у кожному стовпці.

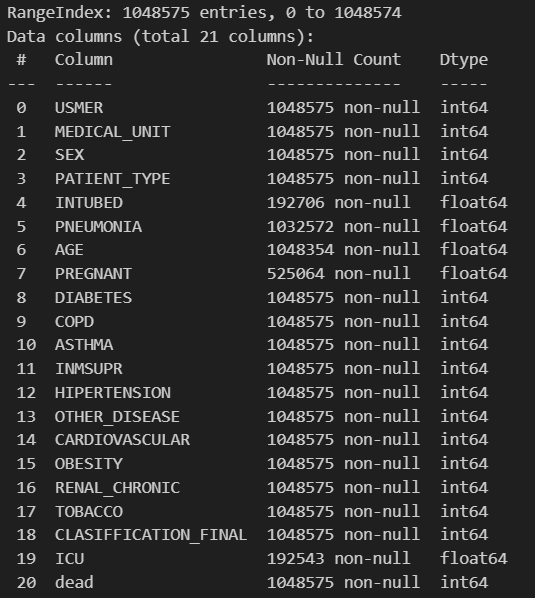


**Рисунок 6** Інформація про датасет

Рисунок 6 демонструє кількість пропущених значень, які замінені на число 99 у кожному стовпці.

Переведемо значення 97, 99 і 9999-99-99 в NaN значення, перевіримо заальну кількість пропущених значень.

Рисунок 7 дає інформацію про стовпці, їх імена і кількість.



**Рисунок 7** Інформація про датасет

Рисунок 7 показано 21 стовпець їх тип і кількість пропущених значень у кожному стовпці. Найбільше пропущених значень є у стовпцях з назвою ‘ICU’, ‘INTUBED’, ‘PREGNANT’. Також є невеликі пропущені значення в колонках з назвою ‘AGE’, ‘PNEUMONIA’.

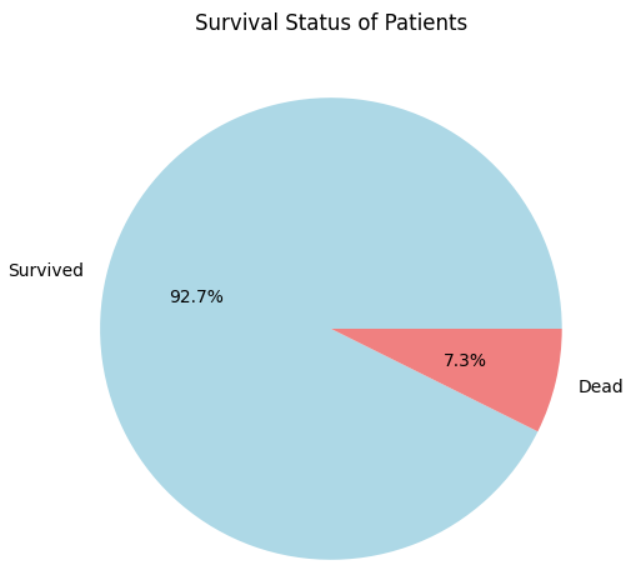
Інформація про значення стовпців у наборі даних пояснено у таблиці 4:

|  |  |
| --- | --- |
| **Ім’я колонки** | **Пояснення значення** |
| Sex | Стать  0 – жінка  1 – чоловік |
| Age | Вік людини |
| Сlassification | Р езультати тесту на COVID. Значення 1-3 означають, що пацієнт був діагностований з COVID різних ступенів важкості. 4 або більше означає, що пацієнт не є носієм COVID або тест недостовірний. |
| Patient type | Тип догляду за пацієнтом в медичному закладі.  1 – виписаний додому  2 – госпіталізований. |
| Pneumonia | Чи має пацієнт запалення повітряних мішків. |
| Pregnancy | Чи вагітний пацієнт. |
| Diabetes | Чи має пацієнт діабет. |
| COPD | Хронічна обструктивна хвороба легень. |
| Asthma | Показник астми. |
| Inmsurp | Чи має пацієнт імунодепресію. |
| Hypertension | Показник гіпертонії. |
| Сardiovascular | Показник, чи має пацієнт захворювання серця чи судин. |
| Renal chronic | Чи має пацієнт хронічну ниркову хворобу. |
| Other disease | Інші захворювання. |
| Obesity | Пацієнт має ожиріння чи ні. |
| Tobacco | Показник тютюну |
| USMER | Показує, чи отримував пацієнт лікування в медичних одиницях першого, другого чи третього рівня. |
| Medical unit | Тип установи Національної медичної системи, яка надавала догляд. |
| Intubed | Чи був пацієнт підключенй до апарату штучної вентиляції легень. |
| Icu | Показує, чи був пацієнт прийнятий до відділення інтенсивної терапії. |
| Date died | Показує чи пацієнт помер. Якщо пацієнт помер, вказує дату смерті; якщо ні, то значення 9999-99-99. |

**Таблиця 4** Пояснення характеристик датасету

Цільова колонка є date died, оскільки ця колонка дає ключове значення, чи людина при певному стані здоров’я смертельно хвора чи може виздоровіти.

У рисунку відображено процентне співвідношення – кількість людей які померли і не померли від коронавірусу.



**Рисунок 8** Кількість людей які померли і не померли від коронавірусу

Рисунок 8, Набір даних містить 92.7% не померлих людей і 7.3% померлих людей. Хоч і різниця між даними є дуже великою, але це не робить датасет не збалансованим, оскільки це відображає реальну ситуацію в світі.



**Рисунок 9** Опис метрик

На рисунку 9 показано середнє значення, мода, медіана кожного стовпця у датасеті. У датасеті сильно відрізняються результати від кожної метрики. У колонці ‘Age’ відрізняється вік між модою і середнім значенням з медіаною, він містить 30 і 40, 41 рік. Також сильно відрізняється ‘MEDICAL\_UNIT’ – медіана відрізняється від моди і середнього значення на 4 позначки, що має великий вплив на результати. У цьому датасеті результати середнього значення, медіани і моди є в більшості одинаковими порівняно з результатами першого датасету. Результати середнього значення є дробовими, оскільки ми сумуємо усі дані, і ділимо на їх кількість у стовпці.

Він містить 1048575 різних випадків, рядків. У цьому датасеті ми визначатимемо, які симптоми коронавірусу призводять до смерті. Ключова цільова колонка є date\_died. Оскільки через цю колонку ми знаємо, як впливають симптоми на людину.

* + 1. **Diabetes Health Indicators Dataset**

Цей датасет містить інформацію про здоров'я та показники хронічних захворювань серед американців, яка була зібрана за допомогою опитування BRFSS2015, проведеного Центром контролю та профілактики захворювань (CDC).

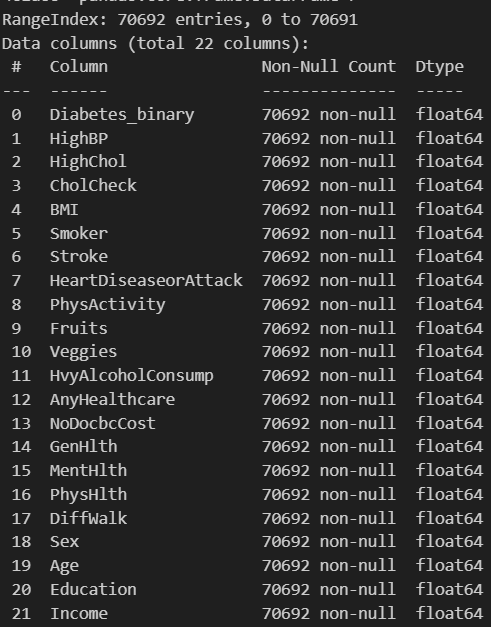
Датасет містить відповіді від 70692 осіб та включає 21 змінну ознаку, яка визначає наявність діабету серед опитаних.

|  |  |  |  |
| --- | --- | --- | --- |
| Статистика датасету | | Тип змінних | |
| Кількість стовпців | 21 | Категорійні | 10 |
| Кількість змінних | 70692 | Числові | 11 |

**Таблиця 5** Статистика датасету

У таблиці 5 вираховано з набору даних скільки стовпців є є категоріальні і числові. Датасет містить відповіді від 70 692 осіб та включає 21 змінну, з яких 10 стовпців є категоріальними, а 11 числовими, які визначають наявність діабету серед опитаних. Даний датасет є більшим за перший датасет який містив 1622 рядки і меншим за другий датасет який містить 1048575 даних про хворих людей на корона вірус. Він містить найбільшу кількість категоріальних стовпців – 10.

Рисунок 10 дає інформацію про тип стовпців, імена стовпців і їх кількість і кількість пропущених значень у кожному стовпці.



**Рисунок 10** Інформація про датасет

На рисунку 10 зображено назву кожного стовпця, його тип і кількість пропущених значень. У цьому датасеті немає пропущених значень, їх вставлятимемо штучно: 5%, 10%, 15 від загального обсягу даних для порівняння результатів взалежності від пропущених значень.

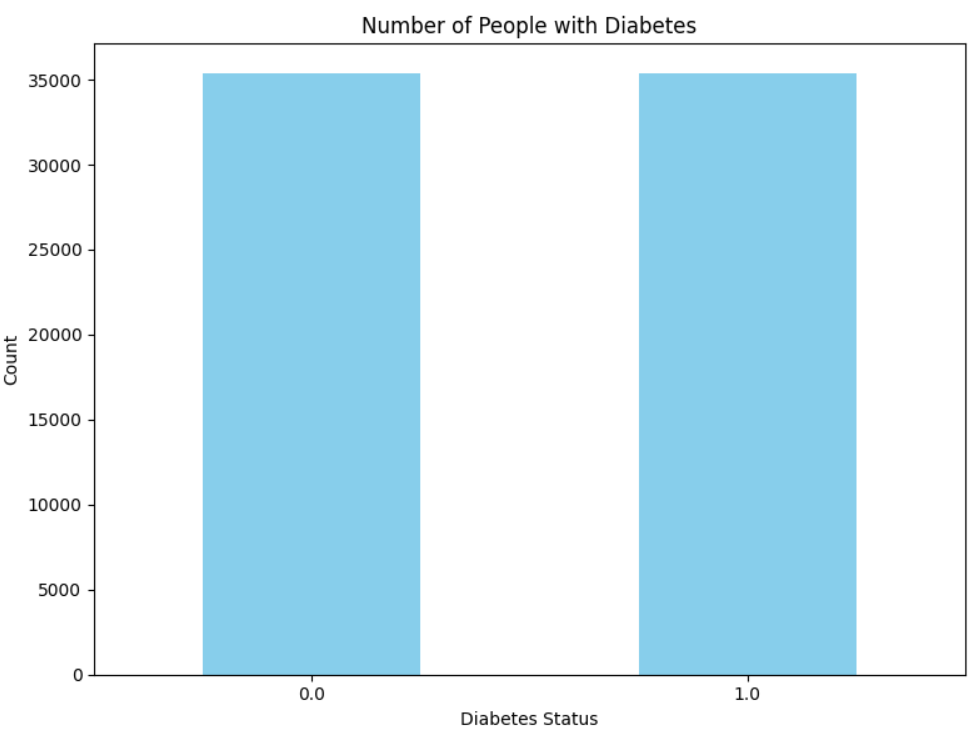
Інформація про значення стовпців у наборі даних пояснено у таблиці 2:

|  |  |
| --- | --- |
| **Ім’я колонки** | **Пояснення значення** |
| Diabetes\_binary | Показник наявності діабету. Має 2 класи: 0 - відсутність діабету або тільки під час вагітності, 1 – діабет. |
| HighBP | Показник наявності високого артеріального тиску. |
| HighChol | Показник наявності високого рівня холестерину. |
| CholCheck | Показник проведення перевірки рівня холестерину. |
| BMI | Індекс маси тіла, який використовується для оцінки ступеня ожиріння або худоби учасників. |
| Smoker | Чи пацієнт курить чи ні. |
| Stroke | Показник наявності інсульту. |
| HeartDiseaseorAttack | Показник наявності серцево-судинних захворювань або серцевого нападу. |
| PhysActivity | Показник фізичної активності. |
| Fruits | Кількість споживаних фруктів. |
| Veggies | Кількість споживаних овочів. |
| AnyHealthcare | Показник доступності будь-якої медичної допомоги. |
| HvyAlcoholConsump | Показник важкого споживання алкоголю. |
| NoDocbcCost | Показник відсутності витрат на медичну допомогу. |
| GenHlth | Загальний стан здоров'я. |
| MentHlth | Психічний стан. |
| PhysHlth | Фізичне здоров’я. |
| DiffWalk | Показник складності ходьби. |
| Sex | Стать 1 – жінка, 2 – чоловік. |
| Age | Вік. |
| Education | Рівень освіти. |
| Income | Дохід пацієнта. |

**Таблиця 6** Пояснення характеристик датасету

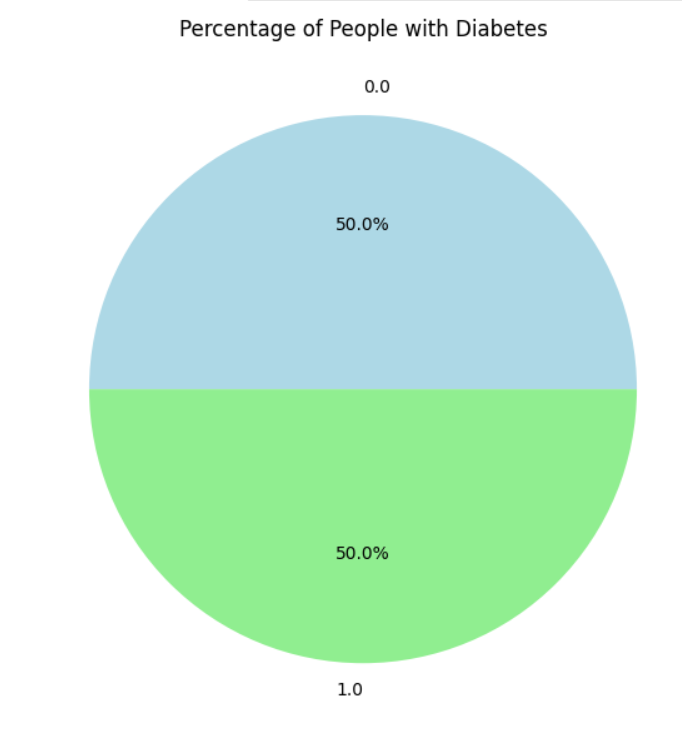
У таблиці 6, цільова колонка є Diabetes\_binary, де будемо передбачувати чи можливий діабет. Де 0 – немає діабету, а 1 – можливий діабет.

У рисунку відображено кількість людей які мали діабет і не мали діабету в датасеті.



**Рисунок 11** Кількість людей з діабетом, стовпчаста діаграма

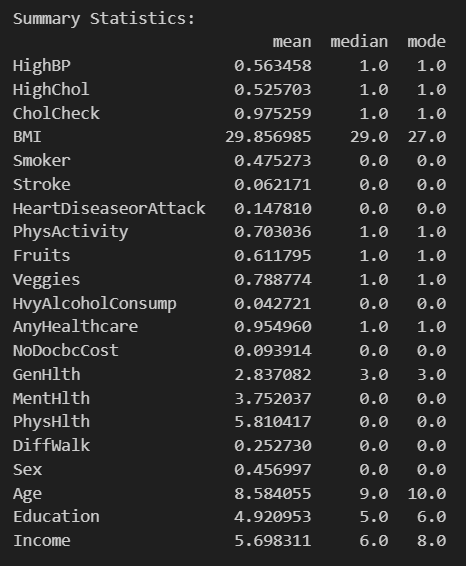
Рисунок 11 показує стовпчасту діаграму, де кількість людей які хворі на діабет, де 0 – немає діабету, а 1 – є діабет. Бачимо, що даний датасет є збалансованим, містить одинакову кількість здорових і хворих людей.



**Рисунок 12** Кількість людей з діабетом, кругова діаграма

Рисунок 12 відображає кругову діаграму, де кількість людей які хворі на діабет, де 0 – немає діабету, а 1 – є діабет. Показано в відсотковому відношенні, де 50% датасету є люди з діабетом, а інші 50% людей без діабету.

На рисунках 11 і 12 видно, що набір даних є збалансованим. Кількість людей які мають діабет в наборі даних є 50% від загальної кількості людей.



**Рисунок 13** Опис метрик

На рисунку 13 показано середнє значення, мода, медіана кожного стовпця у цьому датасеті. Дані одинакові у стовпцях медіани і моди, крім середнього значення.

Основна цільова змінна Diabetes\_binary має 2 класи:

0 – відсутність діабету або тільки під час вагітності,

1 – діабет.

**2.2 Опис та обгрунтування вибраних моделей**

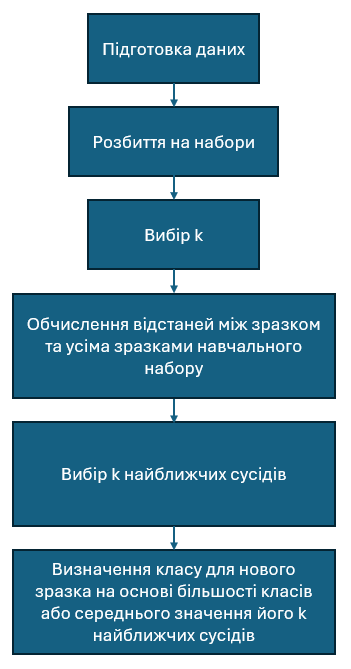
Ми використаємо різні математичні моделі, для того, щоб досягти максимальну ефективність для тренування, передбачування моделей і виявлення найкращої обробки NaN значень. А саме: КНН, логістична регресія, дерева рішень, Наївний Баєс, підсилення градієнта (gradient boosting). Пояснимо різницю між кожним методом машинного навчання:

**2.2.1 KNN**

Алгоритм k-найближчих сусідів (KNN) – це один з найпопулярніших алгоритмів машинного навчання для класифікації та регресії. Він базується на припущенні, що схожі об'єкти зазвичай належать до одного класу або мають схожі значення цільової змінної.

**Послідовність роботи алгоритму KNN**:

1. Підготовка даних.
2. Розбиття даних на навчальний та тестові набори.
3. Вибір значення k (кількість найближчих сусідів).
4. Обчислення відстаней між новим зразком та усіма зразками навчального набору.
5. Вибір k найближчих сусідів.
6. Визначення класу для нового зразка на основі більшості класів (тобто для класифікації) або середнього значення (для регресії) його k найближчих сусідів.



**Рисунок 14** Послідовність алгоритму KNN

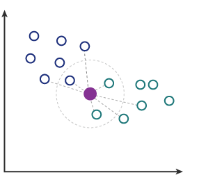
Рисунок 14 показує загальну послідовність алгоритму KNN.

**Переваги алгоритму KNN:**

* Простота реалізації.
* Ефективність у вирішені задач класифікації з невеликим набором атрибутів.
* Можна використовувати для задач регресії і класифікації.
* Не вимагається попереднє навчання моделі
* Адаптивний до змін
* Може моделювати складні нелінійні залежності між ознаками та цільовими змінними.

**Недоліки алгоритму KNN:**

* Обчислювально витратний для великого набору даних.
* КНН дуже чутливий до шуму в даних, що може призвести до неправильної класифікації або прогнозів.



**Рисунок 15** Зображення класифікації точки

Рисунко 15 відображає приклад пошуку класифікації точки серед інших відомих точок.

Отже, KNN - це ефективний алгоритм з простим механізмом прийняття рішення, але його ефективність залежить від обсягу даних, наявності шуму та вибору оптимального значення параметра k.

**2.2.2 Логістична регресія**

Логістична регресія – це метод для аналізу даних, що використовується для прогнозування ймовірностей бінарних подій. Вона є одним з основних алгоритмів класифікації у машинному навчанні, особливо для випадків, коли маємо два можливих результати або класи. Основна ідея логістичної регресії полягає в тому, щоб побудувати математичну модель, яка оцінює ймовірність того, що зразок належить до певного класу. Ця ймовірність виражається у вигляді значень від 0 до 1, де 0 означає неможливість події, а 1 - абсолютна достовірність.

**Послідовність роботи логістичної регресії:**

1. **Підготовка даних**: підготувати і обробити дані для датасету, перевірити вірність даних і наявність NaN значень.
2. **Розділення даних:** розділити дані на навчальний набір та тестовий набір для оцінки ефективності моделі.
3. **Побудова моделі:** використовуючи навчальний набір, побудувати математичну модель логістичної регресії, що відображає зв’язок між залежними та незалежними змінними.
4. **Навчання моделі:** застосувати метод максимальної ймовірності або інші методи оптимізації для навчання моделі на навчальних даних.
5. **Оцінка моделі:** через тестовий набір оцінити точність та ефективність моделі на прогнозування нових зразків.

**Формула алгоритму:**

*P(y = 1|x) =*

де

*P(y = 1|x) –* ймовірність, що вихідний результат *𝑦* дорівнює 1 (позитивний клас), при умові вхідних ознак *𝑥*.

e – число Ейлера

- параметри моделі, які підбираються під час навчання.

*x1*, *x2, …, xn* – вхідні ознаки, які використовуються для прогнозування.

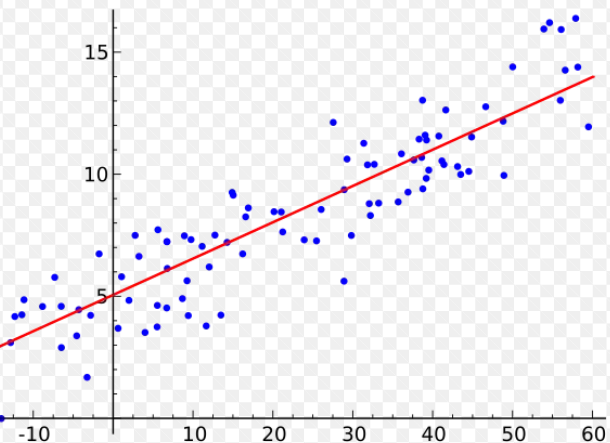
*y* – бінарний результат (категорія), який прогнозується 0 або 1.

**Переваги логістичної регресії:**

* Це простий і потужний алгоритм для бінарної класифікації.
* Логістична регресія може враховувати надлишкові ознаки або мультиколінеарність, що дозволяє уникнути перенавчання та покращити загальну ефективність моделі.
* Логістична регресія дає змогу встановлювати власні пороги для прийняття рішень. Це є великою перевагою, у тих задачах, коли необхідно мати баланс між чутливістю та специфічністю.

**Недоліки логістичної регресії:**

* Вона може бути не ефективною у моделюванні складних нелінійних залежностей в даних, через те що це лінійна модель.
* Аномалії в даних можуть впливати на точність та надійність результатів логістичної регресії.
* Логістична регресія може бути обмежена у вирішенні завдань, де існують сильні нелінійні залежності між ознаками та цільовими змінними.



**Рисунок 16** Зображення прямої для логістичної регресії

На рисунку 16 намальовано пряму, за якою класифікуватимуться точки.

Логістична регресія - це потужний і широко використовуваний алгоритм класифікації, який дозволяє прогнозувати ймовірність настання події на основі вхідних ознак. Однак ефективність цього алгоритму може залежати від характеристик даних та налаштувань моделі.

**2.2.3 Дерева рішень**

Дерево рішень (Decision tree) — це непараметричний контрольований алгоритм навчання, який можна застосовувати для опрацювання як дискретних, так і безперервних даних. Він має ієрархічну структуру дерева, яка складається з кореневого вузла, гілок, внутрішніх вузлів і листових вузлів. Основне завдання алгоритму, це розподілити набір даних на підмножини на основі найважливішого атрибута у цих даних.

**Основні сфери застосування дерев рішень:**

* Класифікація даних
* Регресивний аналіз даних

**Покроковий опис алгоритму обрахунку дерев рішень:**

1. Підготувати дані. Для усіх значень знайти значення Gini(D).
2. Знайти Ginik(D) для кожного атрибута k: Ginik(D) = ∑ (Di / D ) \* Gini(Di)
3. Розбити за мінімальним значенням індексу Gini. Тобто елемент з найменшим значенням Gini, буде коренем нашого дерева, а всі подальші гілки і листки додаватимуться за зростанням значення індексу Gini.

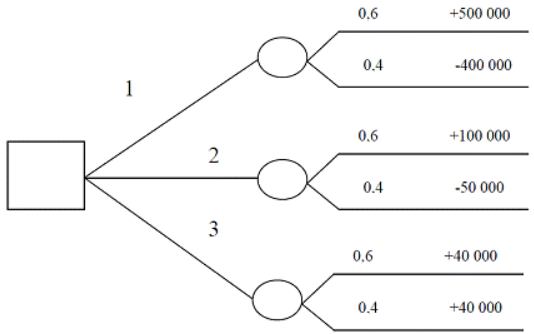
**Переваги алгоритму дерева рішень:**

* Дерева рішень легко інтерпретувати, оскільки вони можуть бути візуалізовані у вигляді деревоподібних структур з правилами прийняття рішень.
* Дерева рішень не потребують нормалізації даних, оскільки вони розглядають кожну ознаку окремо та нечутливі до масштабування даних.
* Дерева рішень можна використовувати як для класифікації, так і для регресійних завдань.
* Дерева рішень можуть ефективно обробляти великі обсяги даних без значних витрат часу на підготовку даних.

**Недоліки алгоритму дерева рішень:**

* Дерева рішень можуть перенавчатися на тренувальних даних і демонструвати високу точність на навчальному наборі, але низьку узагальнюючу здатність на нових даних.
* Дерева рішень виявляють обмежену здатність моделювати складні нелінійні залежності між ознаками та цільовими змінними.
* Невеликі зміни у навчальному наборі можуть призводити до значних змін у структурі та результативності дерева рішень.

Отже, дерева рішень – це простий та зрозумілий спосіб моделювання рішень, який може бути застосований до різних завдань машинного навчання.



**Рисунок 17** Класифікація дерева рішень

НА рисунку 17 показано класифікацію числа, до інших чисел. Де гілка 1 має межі від 100 до 500, гілка 2 має межі від 50 до 100, гілка 3 має межі від 40 до 0.

**2.2.4 Наївний Баєс**

Наївний Баєс - це простий, але потужний алгоритм машинного навчання, який використовує ймовірність для прийняття рішень. Його основна ідея полягає в тому, що він визначає ймовірність того, що певний об'єкт належить до певного класу, враховуючи його атрибути чи характеристики. Наприклад, він може визначати ймовірність того, що лист - це спам або не спам, відповідно до його вмісту та властивостей. Це дає можливість автоматично розподіляти об'єкти в різні категорії, що є корисним для багатьох задач, таких як класифікація документів, виявлення шахрайств, медичні діагнози тощо.

**Покроковий опис алгоритму Наївного Баєса:**

1. Спочатку потрібно підготувати дані, включаючи навчальний набір з об'єктами та відповідними класами чи категоріями, які потрібно передбачити. Обробити дані, щоб вони були коректними і не містили NaN значення.
2. Для кожного класу обчислюють ймовірності віднесення об'єкта до цього класу, використовуючи формулу теореми Баєса. Це включає розрахунок апостеріорної ймовірності P(C|x) - ймовірності класу С при умові, що відомі атрибути x.
3. Наївний Баєс припускає незалежність між атрибутами, тобто він вважає, що значення кожного атрибуту впливає на клас незалежно від інших атрибутів. Це спрощує обчислення ймовірностей, але це припущення не завжди відповідає реальним даним. Навчаємо модель.
4. Після навчання моделі і обчислення ймовірностей для кожного класу відбувається прогнозування класу для нових об'єктів. Об'єкт призначається класу з найвищою апостеріорною ймовірністю.

**Формула**

*P(Ck|x1,x2, …, xn) =*

де:

*P(Ck|x1,x2, …, xn)* – ймовірність належності прикладу *x* до класу *Ck*, при умові, що відомі ознаки *x1,x2, …, xn* .

*P(Ck)* – ймовірність того, що вибраний приклад належить класу *P(Ck)* без урахування будь-яких ознак.

– умовна ймовірність ознак при умові, що приклад належить класу *Ck .*

– ймовірність вектору ознак у загальному випадку, незадежно від класу.

**Переваги алгоритму Наївного Баєса:**

* Простий та ефективний алгоритм, що швидко навчається на великих обсягах даних.
* Особливо ефективний для роботи з категоріальними та текстовими даними, де він може показати високу точність.
* Не потрібно налаштовувати багато гіперпараметрів або враховувати складні взаємозв'язки між ознаками.

**Недоліки алгоритму Наївного Баєса:**

* Наївний Баєс базується на припущенні про незалежність між ознаками, що може бути недостатнім для деяких даних, де взаємозв'язки складніші.
* У деяких складних завданнях, де залежність між ознаками значна, наївний Баєс може показати низьку точність прогнозування.
* Якщо в навчальному наборі виникають нульові ймовірності для деяких комбінацій ознак, алгоритм може давати неправильні прогнози.
* Наївний Баєс може бути вразливим до шуму в даних та впливу незначущих ознак, що може погіршити якість прогнозів.
* Якщо взаємозв'язки між ознаками виходять за межі наївного припущення, алгоритм може втратити точність та неадекватно моделювати дані.

Наївний Баєс працює швидко та ефективно навіть з великими обсягами даних, він відносно простий у реалізації та має низький ризик перенавчання. Він добре працює з великою кількістю атрибутів та може бути використаний для категоріальних та числових даних. Однак, наївний Баєс робить дуже спрощені припущення про незалежність атрибутів, що може бути неідеальним у деяких реальних ситуаціях. Також, якщо в наборі даних є атрибути, які сильно впливають на результат, але вони неправильно припускаються як незалежні, то це може призвести до неадекватних прогнозів.

**2.2.5 Підсилення градієнту**

Підсилення градієнту – використовується для підвищення точності прогнозування моделі шляхом послідовного навчання низькорівневих моделей, кожна з яких коригує попередні помилки, що здійснюється на основі градієнта (зміни) функції втрати. Основна ідея полягає в тому, щоб послідовно створювати слабкі моделі, які компенсують недоліки попередніх моделей.

**Послідовність алгоритму:**

1. **Ініціалізація базової моделі:** початкова модель може бути простою моделлю, наприклад, константним прогнозом або середнім значенням цільової змінної.
2. **Розрахунок помилок:** розраховуємо вектор помилок, який представляє різницю між спостереженими значеннями та прогнозами початкової моделі.
3. **Навчання нової моделі:** нова модель навчається на векторі помилок, тобто спробує покращити прогнози попередньої моделі.
4. **Оновлення прогнозів:** по закінченні навчання нової моделі, прогнози попередньої моделі оновлюються, додаючи до них прогнози нової моделі з вагою (зазвичай, за допомогою швидкості навчання або кроку навчання).
5. **Розрахунок нового вектора помилок:** рахуємо новий вектор помилок, який тепер враховує виправлені прогнози.
6. **Повторення кроків 3-5:** повторюємо цей процес (зазвичай кілька разів) для навчання додаткових моделей та покращення прогнозів.
7. **Закінчення:** після завершення всіх ітерацій, ансамбль моделей об'єднується для утворення кінцевого прогнозу.

**Формула:**

*Fm(x) = Fm-1(x) +ή\* hm(x)*

де:

*Fm(x)* – прогноз моделі на *m* кроці для вхідних значень *x*.

*Fm-1(x)* - прогноз моделі на *m* кроці для вхідних значень *x*.

*ή –* швидкість навчання (learning rate), яка визначає, наскільки швидко модель навчається на кожному кроці.

*hm(x) –* слабка модель (зазвичай дерево рішень), яка навчається на помилках попередніх прогнозів *Fm-1(x).*

**Переваги алгоритму підсилення градієнту:**

* Підсилення градієнту часто дає дуже точні результати, особливо в порівнянні з іншими алгоритмами.
* Цей метод може ефективно моделювати складні взаємозв'язки в даних та робити точні прогнози.
* Може працювати без необхідності передобробки даних.
* Алгоритм може використовувати різні функції втрати в залежності від конкретної задачі.

**Недоліки алгоритму підсилення градієнту:**

* Якщо не налаштовувати гіперпараметри належним чином, підсилення градієнту може перенавчитися.
* Цей метод може вимагати більше часу та ресурсів для навчання порівняно з іншими алгоритмами.
* Підсилення градієнту може бути вразливим до шуму в даних та випадкових варіацій, що може призвести до невірних прогнозів.

Підсилення градієнту є потужним та ефективним алгоритмом для прогнозування в машинному навчанні, здатним до вирішення складних задач та досягнення високої точності. Проте його використання вимагає належного налаштування та обережності, щоб уникнути перенавчання та врахувати його вразливість до шуму.

**2.3 Опис методів обробки NaN значень**

Вибір конкретного методу обробки NaN значень повинен базуватися на характеристиках самого датасету та вимогах конкретної аналітичної задачі.

**Імпутація** – це заповнення пропущених значень на основі інших доступних даних. Наприклад: медіана, середнє значення, мода, заміна на константу, видалення NaN значень, KNN.

**Різновиди імпутації:**

* **Медіана** – це значення, яке ділить впорядкований набір чисел навпіл, так що половина чисел менша за медіану, а половина більша.

Як працює метод: для кожного стовпця з пропущеними значеннями обчислюється медіана з відомих значень, і пропущені значення заповнюються цією медіаною.

Цей метод використовується тоді, коли дані мають викиди або відносно велику варіацію, оскільки медіана менш чутлива до викидів, ніж середнє значення, то вона є ефективним вибором для числових даних з великою кількістю викидів або аномалій.

* **Середнє значення** – всі числа у стовпці додаються і діляться на кількість чисел у наборі стовпця. Цей метод підходить для числових даних, які мають нормальний розподіл і відсутність викидів. Використання середнього значення може бути ефективним для великих наборів даних з невеликою кількістю пропусків.
* **Мода** – це показник центральної тенденції у статистиці, який визначається як найбільш часто зустрічатиме значення в наборі даних. Цей метод добре підходить для категоріальних даних або даних з дискретними значеннями, де можна визначити найбільш часто зустрічаємі значення. Він ефективний, коли потрібно заповнити пропуски у категоріальних змінних або змінних з обмеженим набором значень.
* **Видалення NaN значень** – це видалення пустих значень, цей метод використовують, коли малий набір даних і мала кількість пропусків.
* **Заміна на константу** – це замінювання всіх пропущених значень на певну фіксовану константу.
* **KNN** (метод найближчих сусідів) можна використовувати для імпутації пропущених значень (NaN) у даних. Основна ідея полягає в тому, що для кожного пропущеного значення можна знайти k найближчих сусідів з відомими значеннями цієї ознаки і використати їхні значення для заповнення пропущеного.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Метод імпутації** | **Вплив викидів** | **Тип даних (числові / категоріальні)** | **Розмір набору даних** | **Ефективність** |
| Медіана | Менше впливають | Числові | Будь-який | Залежить від розподілу даних. Ефективний при наявності викидів |
| Середнє значення | Чутливий | Числові | Будь-який | Ефективний для нормально розподілених даних без викидів |
| Мода | Не впливають | Категоріальні | Будь-який | Ефективний для категоріальних даних з невеликою кількістю пропусків |
| Видалення NaN значень | Видаляється | Будь-який | Малий набір даних | Ефективний при малій кількості пропусків |
| Заміна на константу | Не впливають | Числові /  Категоріальні | Будь-який | Ефективний для наборів даних з невеликою кількістю пропусків |
| KNN | Залежить від налаштування методу | Будь-який | Будь-який | Ефективний при великих наборах даних з різноманітними типами даних |

**Таблиця 7** Порівняння методів імпутації

**2.4 Визначення похибки**

Точність – це відношення правильно класифікованих прикладів до загальної кількості прикладів у тестовому наборі даних. Формула для обчислення точності виглядає так:

*Accuracy =*

де *m –* це кількість правильно класифікованих прикладів, а *n –* загальна кількість прикладів.

У цьому розділі порівняли набори даних між собою, з’ясували які колонки і дані містять датасети. Яка кількість пропущених значень і які цільові колонки необхідно передбачити. З’ясували, які методи машинного навчання використовуватимуться для тренування моделі. Визначили методи імпутації за допомогою яких оброброблятимуться пропущені значеня.