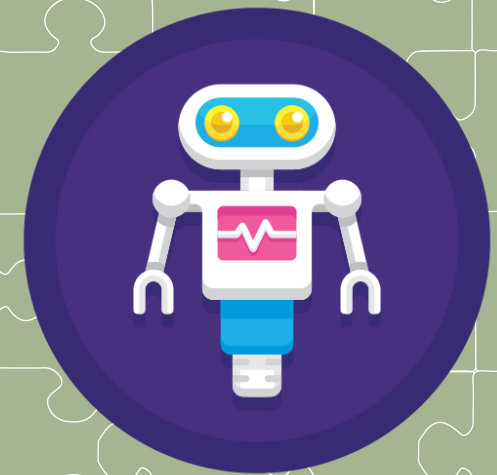
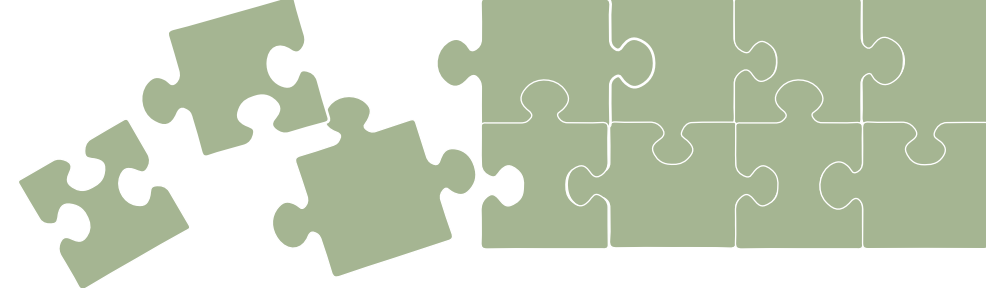


# **Інформаційна система прогнозування артеріальної гіпертензії на основі Telegram-бота**

**Ковалюк Ігор**



# Актуальність проблеми



## Ukraine Hypertension Fact Sheet

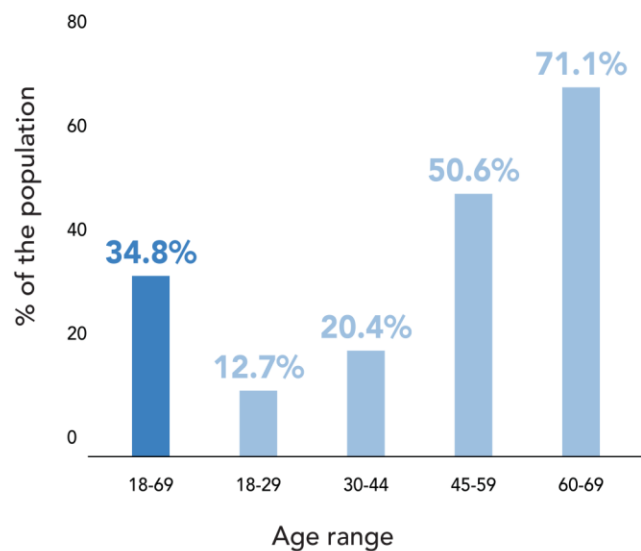


BURDEN OF HYPERTENSION (2019)<sup>1,2</sup>

2019 TOTAL POPULATION: 43 993 643

### 34.8%

**(10.8 million)**  
of the adult population  
have hypertension

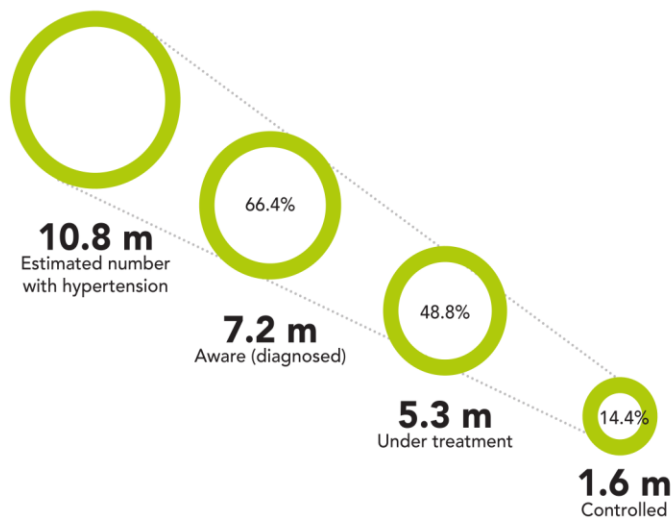


Of the

### 10.8 m → 9.2 m

people with  
hypertension

do not have it  
under control



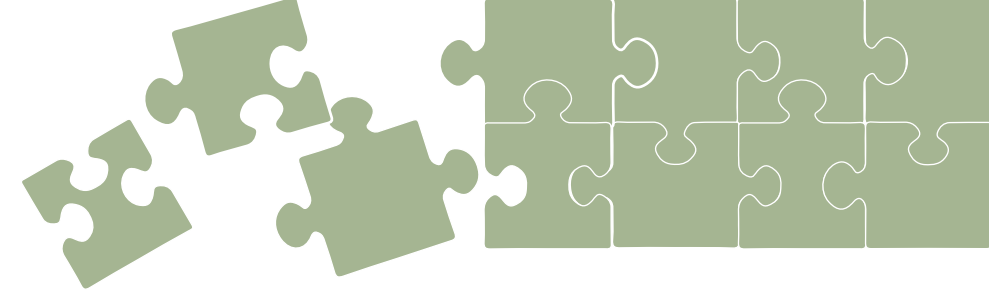
## Світ

За даними Всесвітньої організації охорони здоров'я у світі понад 1,4 мільярди дорослих віком 30–79 років мають підвищений артеріальний тиск.

## Україна

За даними Всесвітньої організації охорони здоров'я за 2019 рік в Україні на артеріальну гіпертензію страждають близько 10-12 мільйонів людей - приблизно кожен третій дорослий.

# Послідовність реалізації



Проект передбачає комплексне програмне рішення – розробку системи штучного інтелекту, яка використовує модель прогнозування на основі нейронної мережі, надає інтерфейс користувача для введення даних і є доступною для використання в інтернеті

## Модель

Обрання, створення та навчання нейронної мережі



## Розгортання

Розгортання системи на сервері для її безперервної роботи

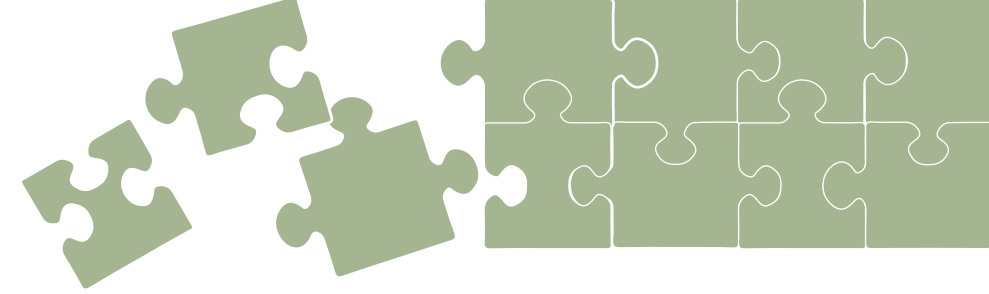


## Система

Інтеграція моделі прогнозування в інформаційну систему



# Постановка задачі дослідження



## Датасет

Для навчання нейронної мережі потрібний набір даних – датасет. В своїй роботі я використовував “Cardiovascular Disease Dataset”, який доступний за посиланням:

<https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset/data>

Обраний датасет містить наступні вхідні фактори: **вік, зріст, вага, стать, систолічний артеріальний тиск, діастолічний артеріальний тиск, холестерин, глюкоза, куріння, вживання алкоголю, фізична активність**. На виході ми отримуємо один з 2 класів - наявність або відсутність артеріальної гіпертензії.

```
Shape: (70000, 13)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   id               70000 non-null  int64
1   age              70000 non-null  int64
2   gender           70000 non-null  int64
3   height           70000 non-null  int64
4   weight           70000 non-null  float64
5   ap_hi            70000 non-null  int64
6   ap_lo            70000 non-null  int64
7   cholesterol      70000 non-null  int64
8   gluc             70000 non-null  int64
9   smoke            70000 non-null  int64
10  alco             70000 non-null  int64
11  active           70000 non-null  int64
12  cardio           70000 non-null  int64
dtypes: float64(1), int64(12)
```

# Навчання нейронної мережі

## Підготовка даних

Розподіл даних на тестову й тренувальну вибірки.  
Масштабування даних

## Навчання НМ

Визначення функції втрат і оптимізатора.  
Процедура навчання

## Збереження

Збереження моделі та скейлера



## Створення моделі

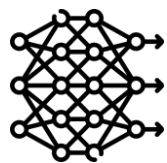
Створюємо модель нейронної мережі - багат шаровий персептрон.

## Оцінка якості моделі

Графік тренування НМ та матриця невідповідностей (confusion matrix)

# Дослідження архітектур мережі

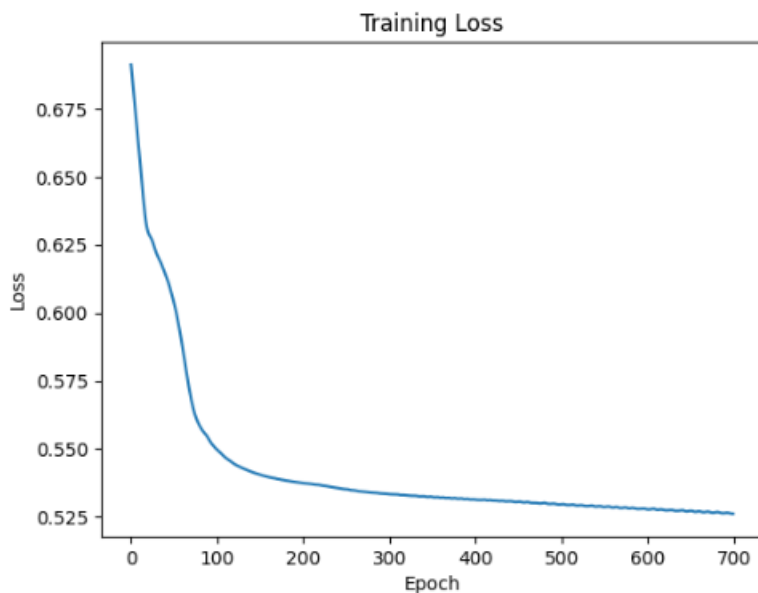
№	Архітектура НМ	Епохи	Accuracy	recall	Confusion Matrix
1	11/64/32/2	500	0,7367857142857142	0.7697/0.7036	[5402, 1616] [2069, 4913]
2	11/128/64/32/2	500	0.7345714285714285	0.7789/0.6904	[5443, 1545] [2171, 4841]
3	11/128/64/32/2	1000	0.7288571428571429	0.8028/0.6552	[5610, 1378] [2418, 4594]
4	11/128/64/2	500	0.7367857142857143	0.7726/0.7011	[5399, 1589] [2096, 4916]
5	11/64/32/16/2	700	0.738	0.7722/0.7039	[5365, 1592] [2076, 4967]
6	11/32/16/8/2	450	0.7395714285714285	0.7816/0.6977	[5462, 1526] [2120, 4892]



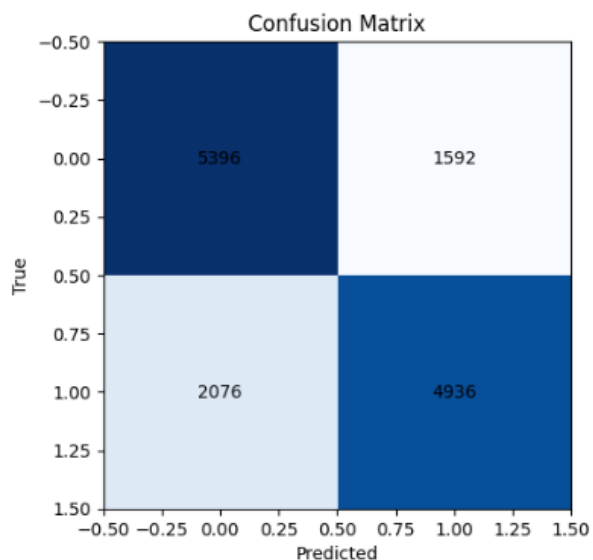
Для підвищення точності НМ було протестовано різні архітектури, а саме змінювалася кількість прихованих шарів та нейронів на них. Також було виконано навчання мережі за різної кількості епох

# Результати навчання

## Графік навчання



## Confusion matrix



## Кількісні показники

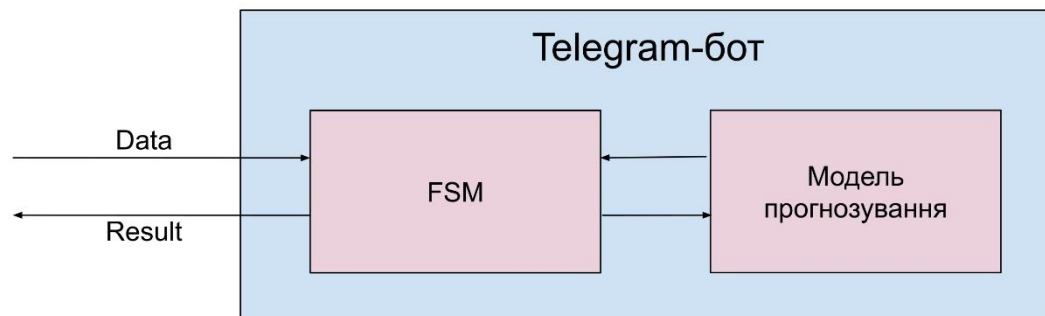
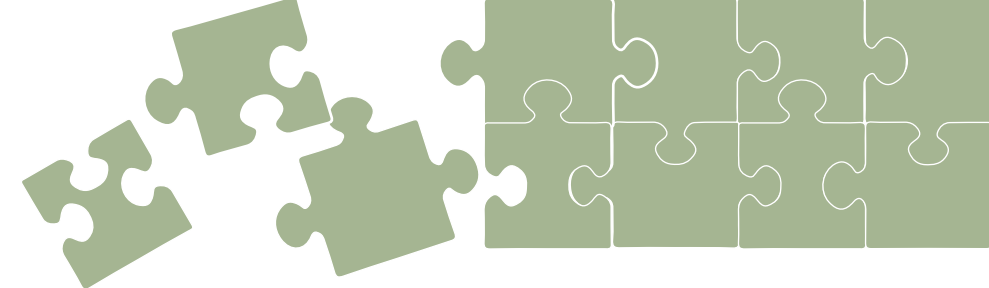
```
Accuracy: 0.738
      precision    recall  f1-score   support
0     0.7222     0.7722     0.7463     6988
1     0.7561     0.7039     0.7291     7012

 accuracy          0.7380     14000
 macro avg         0.7391     0.7381     0.7377     14000
 weighted avg      0.7392     0.7380     0.7377     14000
```

## Коефіцієнт істинно позитивних результатів (TPR)

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

# Створення Telegram-бота



## Реалізація **Aiogram**

Асинхронність, розширена підтримка типізації, middleware, система маршрутизації, машина станів

## Інтерактивність **Finite State Machine**

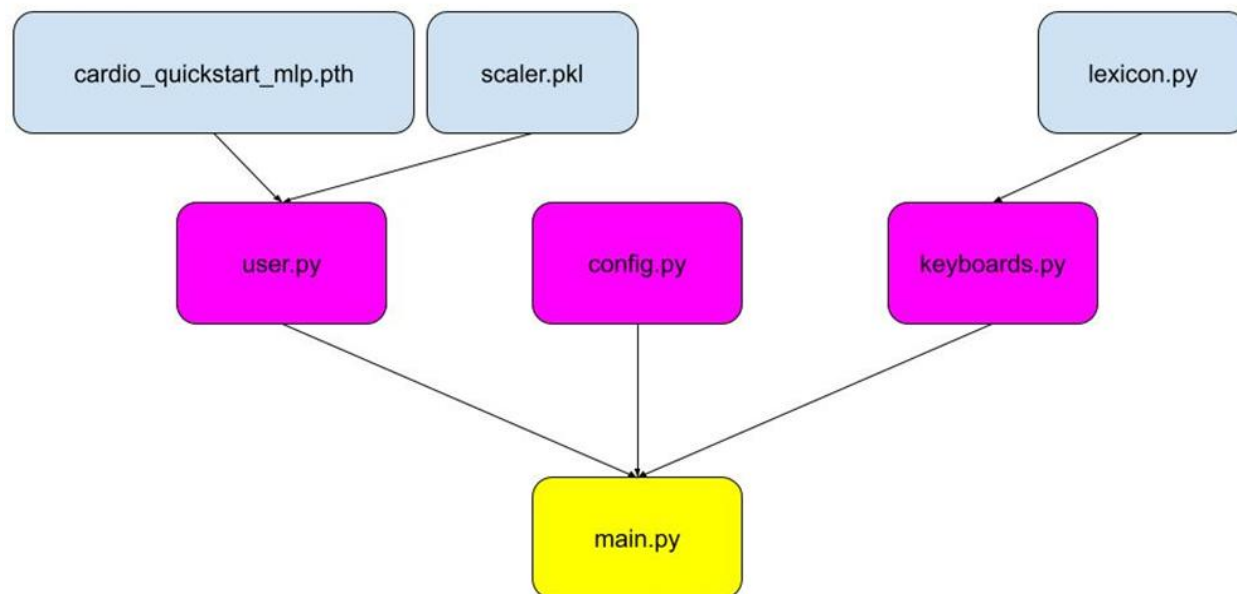
Система здатна динамічно реагувати на запити, змінюючи свій стан. Це зроблено для моделювання складної поведінки у завданнях, де важлива послідовність дій.

## Інтеграція **моделі прогнозування**

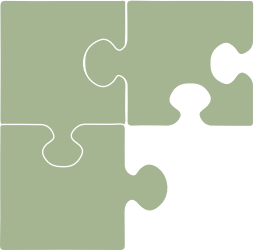
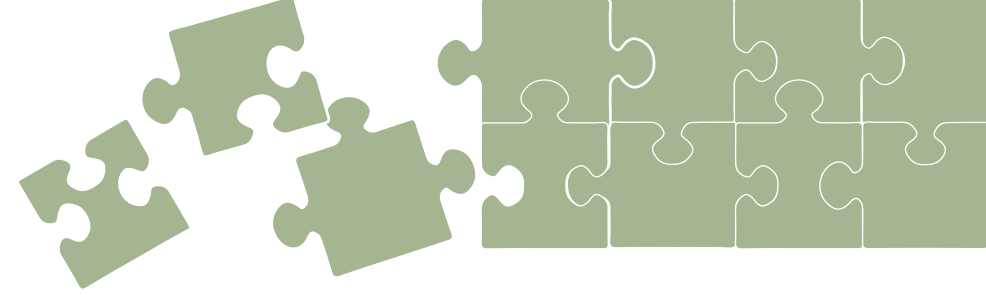
Інтеграція раніше збереженої моделі та її налаштувань до інтерактивного інтерфейсу (бота)

## Структура **окремі модулі**

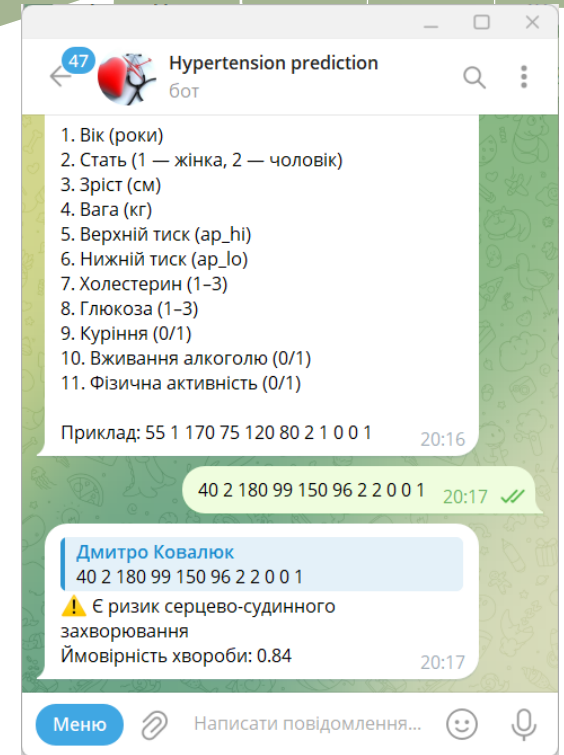
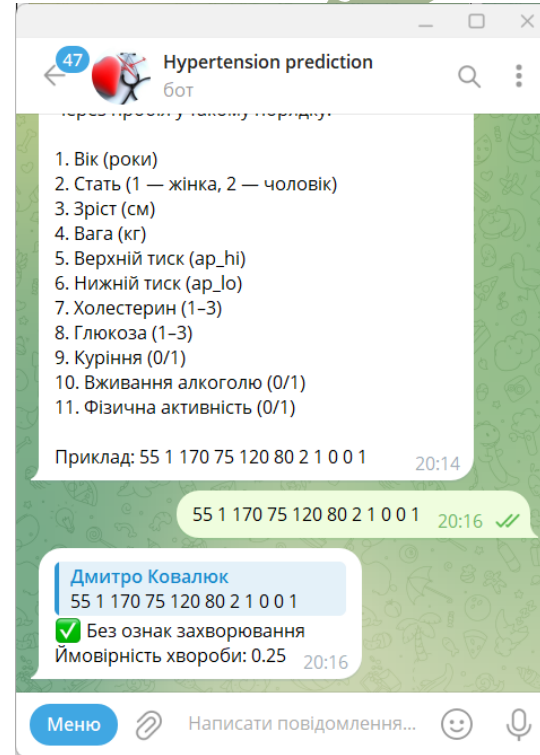
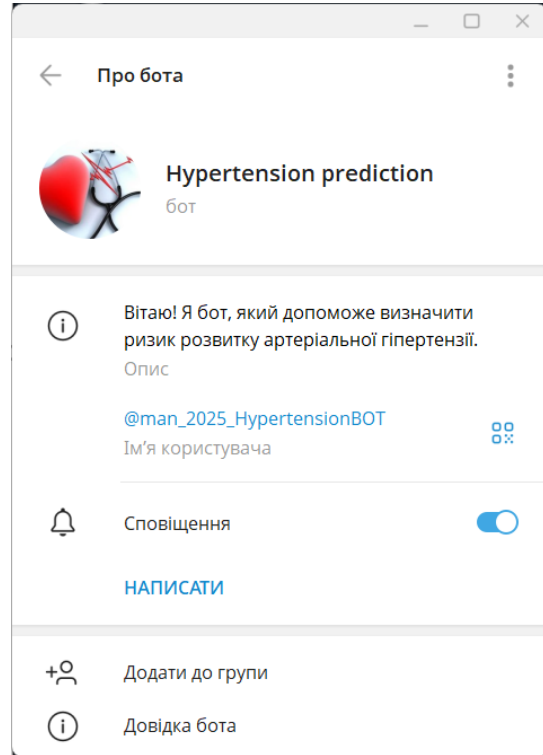
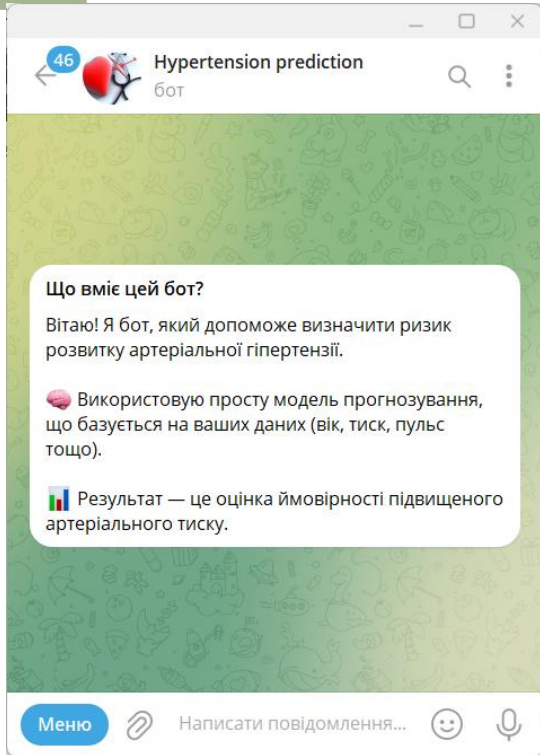
Модульна структура дозволяє легко змінювати проект та структурує взаємодії в ньому, де кожен модуль відповідає за своє завдання



# Переваги AWS та Git



# Результати роботи

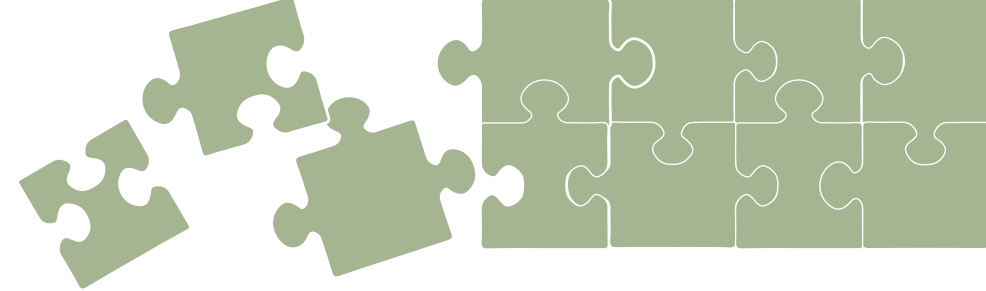


## Алгоритм роботи

Бот отримує на вхід від користувача такі дані як вік, стать, зріст, вага, рівень артеріального тиску, частота серцевих скорочень, наявність шкідливих звичок і рівень фізичної активності.

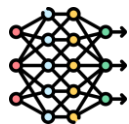
Після введення параметрів бот здійснює прогнозування за допомогою нейронної мережі та повертає користувачу оцінку ймовірності наявності або розвитку артеріальної гіпертензії.

# Висновки



01

## ML-моделі



Знайомство з нейронними мережами для розв'язання задачі класифікації та їх реалізація на PyTorch.

02

## Системи ШІ



Інтеграція моделей ШІ в програмні засоби для інтерактивної взаємодії з користувачами

03

## Розгортання ПЗ



Використання хмарних сервісів для хостингу бота. Проблеми публікації та оновлення ПЗ

04

## Технології розробки ПЗ



Використання системи контролю версій Git, робота за інкрементною моделлю розробки ПЗ