```
Лабороторна робота 2.1 студента К. Д. Долматова

Варіант 3: Pima Indians Diabetes

Опис: Прогнозування наявності діабету у пацієнтів на основі медичних
характеристик. Особливості набору даних: 8 ознак, два класи (наявніс
відсутність діабету)
```

Лабороторна робота 2.1 студента К. Д. Долматова

Варіант 3: Pima Indians Diabetes

Опис: Прогнозування наявності діабету у пацієнтів на основі медичних характеристик. Особливості набору даних: 8 ознак, два класи (наявність/відсутність діабету)

```
!pip install gradio
```

```
Collecting gradio
  Downloading gradio-5.26.0-py3-none-any.whl.metadata (16 kB)
Collecting aiofiles<25.0,>=22.0 (from gradio)
  Downloading aiofiles-24.1.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.9.0)
Collecting fastapi<1.0,>=0.115.2 (from gradio)
  Downloading fastapi-0.115.12-py3-none-any.whl.metadata (27 kB)
Collecting ffmpy (from gradio)
  Downloading ffmpy-0.5.0-py3-none-any.whl.metadata (3.0 kB)
Collecting gradio-client==1.9.0 (from gradio)
  Downloading gradio_client-1.9.0-py3-none-any.whl.metadata (7.1 kB)
Collecting groovy~=0.1 (from gradio)
  Downloading groovy-0.1.2-py3-none-any.whl.metadata (6.1 kB)
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.28.1)
Requirement already satisfied: huggingface-hub>=0.28.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.30.2)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.1.6)
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.0.2)
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.0.2)
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.10.16)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from gradio) (24.2)
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.2.2)
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (11.1.0)
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.11.3)
Collecting pydub (from gradio)
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting python-multipart>=0.0.18 (from gradio)
  Downloading python_multipart-0.0.20-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (6.0.2)
Collecting ruff>=0.9.3 (from gradio)
  Downloading ruff-0.11.7-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (25 kB)
Collecting safehttpx<0.2.0,>=0.1.6 (from gradio)
  Downloading safehttpx-0.1.6-py3-none-any.whl.metadata (4.2 kB)
Collecting semantic-version~=2.0 (from gradio)
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl.metadata (9.7 kB)
Collecting starlette<1.0,>=0.40.0 (from gradio)
  Downloading starlette-0.46.2-py3-none-any.whl.metadata (6.2 kB)
Collecting tomlkit<0.14.0,>=0.12.0 (from gradio)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.15.2)
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.13.2)
Collecting uvicorn>=0.14.0 (from gradio)
  Downloading uvicorn-0.34.2-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.9.0->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.9.0->gradi
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (2025.1.31)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (1.0.8)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.24.1->grad
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (3.18.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (4.6
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio
Requirement already satisfied: pydantic-core==2.33.1 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio)
```

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

# Бібліотеки для машинного навчання
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, auc

# TensorFlow для створення нейронної мережі
```

```
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

# Gradio для інтерактивного інтерфейсу
import gradio as gr
```

```
# Завантаження набору даних
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']
data = pd.read_csv(url, names=columns)

# Перевірка даних
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
# Розділення на вхідні дані та мітки
X = data.drop('Outcome', axis=1)
y = data['Outcome']

# Розділення на тренувальну і тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Нормалізація ознак
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
# Створення моделі
model = Sequential([
    Dense(16, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(8, activation='relu'),
    Dense(1, activation='sigmoid')
])

# Компіляція моделі
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Навчання моделі
history = model.fit(X_train, y_train, epochs=50, batch_size=16, validation_split=0.2, verbose=1)
```

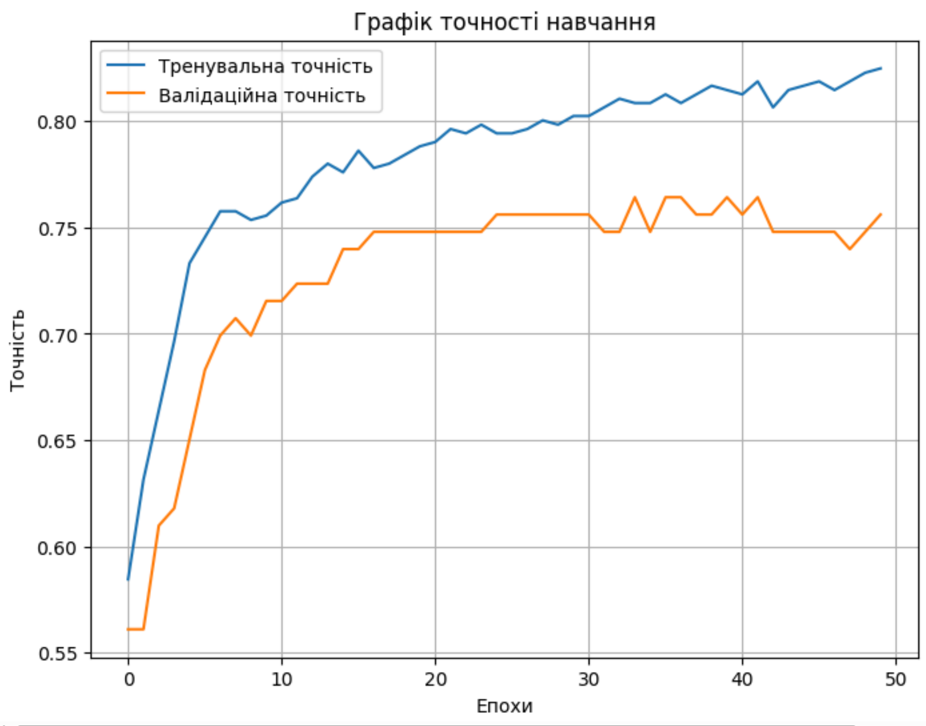
```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` a
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 4s 43ms/step - accuracy: 0.5522 - loss: 0.7090 - val_accuracy: 0.5610 - val_loss: 0.7224
Epoch 2/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 2s 14ms/step - accuracy: 0.6291 - loss: 0.6614 - val_accuracy: 0.5610 - val_loss: 0.6865
Epoch 3/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - accuracy: 0.6713 - loss: 0.6243 - val_accuracy: 0.6098 - val_loss: 0.6515
Epoch 4/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7134 - loss: 0.5920 - val_accuracy: 0.6179 - val_loss: 0.6204
Epoch 5/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.7054 - loss: 0.5750 - val_accuracy: 0.6504 - val_loss: 0.5925
Epoch 6/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.7430 - loss: 0.5405 - val_accuracy: 0.6829 - val_loss: 0.5709
Epoch 7/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - accuracy: 0.7386 - loss: 0.5424 - val_accuracy: 0.6992 - val_loss: 0.5519
Epoch 8/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7861 - loss: 0.4833 - val_accuracy: 0.7073 - val_loss: 0.5388
Epoch 9/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 13ms/step - accuracy: 0.7660 - loss: 0.4693 - val_accuracy: 0.6992 - val_loss: 0.5261
Epoch 10/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 13ms/step - accuracy: 0.7753 - loss: 0.4656 - val_accuracy: 0.7154 - val_loss: 0.5183
Epoch 11/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - accuracy: 0.7376 - loss: 0.4937 - val_accuracy: 0.7154 - val_loss: 0.5096
Epoch 12/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - accuracy: 0.7738 - loss: 0.4559 - val_accuracy: 0.7236 - val_loss: 0.5049
Epoch 13/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - accuracy: 0.7663 - loss: 0.4880 - val_accuracy: 0.7236 - val_loss: 0.4999
```

```
Epoch 14/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - accuracy: 0.7922 - loss: 0.4344 - val_accuracy: 0.7236 - val_loss: 0.4964
Epoch 15/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - accuracy: 0.7928 - loss: 0.4654 - val_accuracy: 0.7398 - val_loss: 0.4925
Epoch 16/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 2s 31ms/step - accuracy: 0.8018 - loss: 0.4359 - val_accuracy: 0.7398 - val_loss: 0.4908
Epoch 17/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 28ms/step - accuracy: 0.7953 - loss: 0.4369 - val_accuracy: 0.7480 - val_loss: 0.4863
Epoch 18/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 27ms/step - accuracy: 0.7904 - loss: 0.4355 - val_accuracy: 0.7480 - val_loss: 0.4871
Epoch 19/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - accuracy: 0.7731 - loss: 0.4419 - val_accuracy: 0.7480 - val_loss: 0.4836
Epoch 20/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - accuracy: 0.8010 - loss: 0.4291 - val_accuracy: 0.7480 - val_loss: 0.4821
Epoch 21/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - accuracy: 0.8189 - loss: 0.4104 - val_accuracy: 0.7480 - val_loss: 0.4816
Epoch 22/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - accuracy: 0.7734 - loss: 0.4524 - val_accuracy: 0.7480 - val_loss: 0.4808
Epoch 23/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - accuracy: 0.7754 - loss: 0.4412 - val_accuracy: 0.7480 - val_loss: 0.4807
Epoch 24/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - accuracy: 0.7832 - loss: 0.4385 - val_accuracy: 0.7480 - val_loss: 0.4811
Epoch 25/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - accuracy: 0.7934 - loss: 0.4155 - val_accuracy: 0.7561 - val_loss: 0.4798
Epoch 26/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - accuracy: 0.7880 - loss: 0.4213 - val_accuracy: 0.7561 - val_loss: 0.4819
Epoch 27/50
31/31 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - accuracy: 0.7714 - loss: 0.4583 - val_accuracy: 0.7561 - val_loss: 0.4788
Epoch 28/50
```

```python
# Оцінка на тестових даних
test_loss, test_accuracy = model.evaluate(X_test, y_test)
print(f"Точність на тестових даних: {test_accuracy:.2f}")
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7463 - loss: 0.5522
Точність на тестових даних: 0.75
```

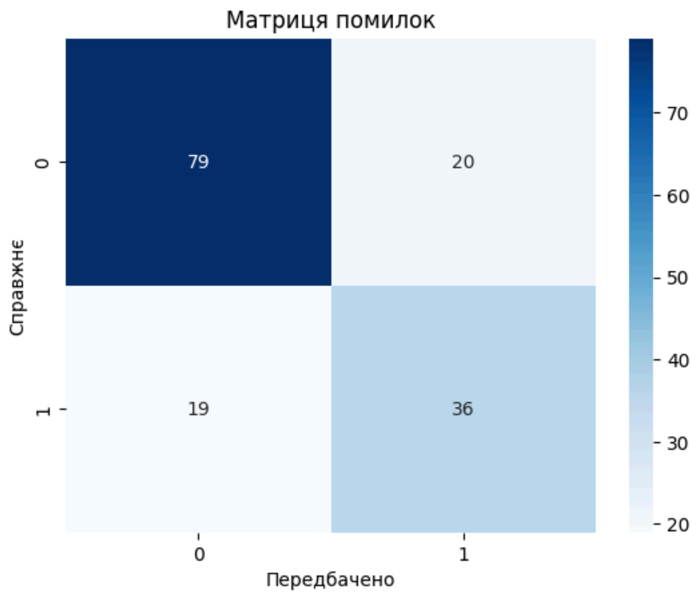
```python
# Візуалізація точності на тренувальних та валідаційних даних
plt.figure(figsize=(8, 6))
plt.plot(history.history['accuracy'], label='Тренувальна точність')
plt.plot(history.history['val_accuracy'], label='Валідаційна точність')
plt.xlabel('Епохи')
plt.ylabel('Точність')
plt.title('Графік точності навчання')
plt.legend()
plt.grid(True)
plt.show()
```



```python
# Передбачення
y_pred_prob = model.predict(X_test)
y_pred = (y_pred_prob > 0.5).astype(int)
```

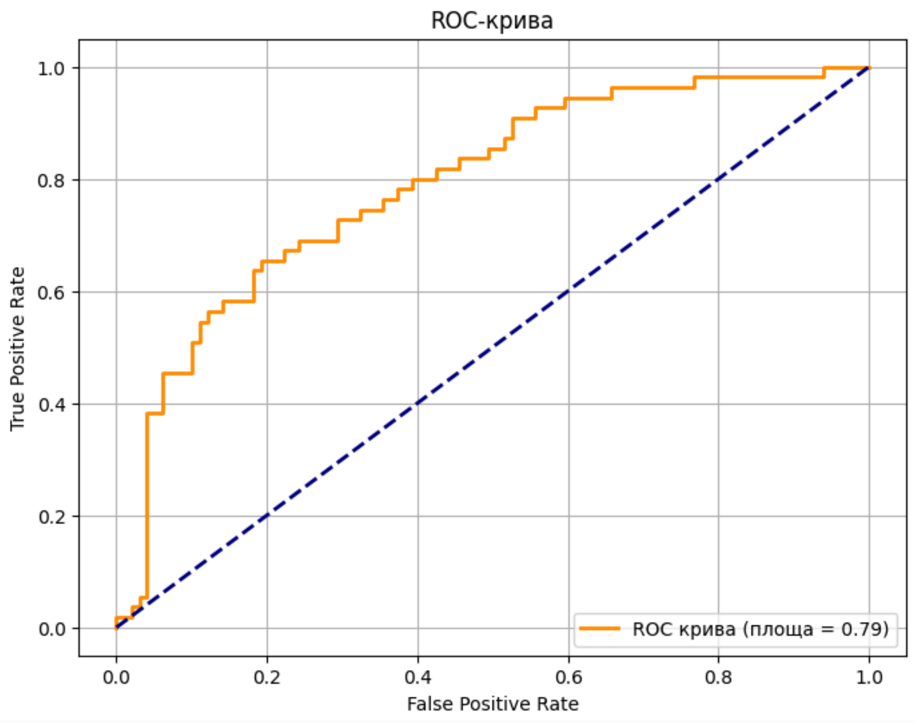
```
# Матриця помилок
conf_mat = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Передбачено')
plt.ylabel('Справжнє')
plt.title('Матриця помилок')
plt.show()
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step
```



```
# Класифікаційний звіт
print(classification_report(y_test, y_pred))

# Побудова ROC-кривої
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC крива (площа = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC-крива')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

```
              precision    recall  f1-score   support

           0       0.81      0.80      0.80        99
           1       0.64      0.65      0.65        55

    accuracy                           0.75       154
   macro avg       0.72      0.73      0.73       154
weighted avg       0.75      0.75      0.75       154
```



```
def predict_diabetes(pregnancies, glucose, blood_pressure, skin_thickness, insulin, bmi, dpf, age):
    input_data = np.array([[pregnancies, glucose, blood_pressure, skin_thickness, insulin, bmi, dpf, age]])
    input_data = scaler.transform(input_data)
    prediction = model.predict(input_data)
    return "Діабет" if prediction[0][0] > 0.5 else "Немає діабету"

iface = gr.Interface(
    fn=predict_diabetes,
    inputs=[
        gr.Number(label="Кількість вагітностей"),
        gr.Number(label="Рівень глюкози"),
        gr.Number(label="Тиск крові"),
        gr.Number(label="Товщина шкіри"),
        gr.Number(label="Рівень інсуліну"),
        gr.Number(label="Індекс маси тіла (BMI)"),
        gr.Number(label="Діабетичний родовідний функціональний індекс (DPF)"),
        gr.Number(label="Вік")
    ],
    outputs="text",
    title="Прогнозування наявності діабету",
    description="Введіть медичні показники для прогнозу наявності діабету."
)

iface.launch()
```

```
It looks like you are running Gradio on a hosted a Jupyter notebook. For the Gradio app to work, sharing must be enabled. Automatica

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://d2e2237f60361a42cb.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working
```