

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler

# Завантаження даних з локального файлу
file_path = "mushroom.data" # Вкажіть шлях до файлу на вашому комп'ютері
column_names = [
    "class", "cap-shape", "cap-surface", "cap-color", "bruises", "odor", "gill-attachment", "gill-spacing",
    "gill-size", "gill-color", "stalk-shape", "stalk-root", "stalk-surface-above-ring",
    "stalk-surface-below-ring", "stalk-color-above-ring", "stalk-color-below-ring", "veil-type",
    "veil-color", "ring-number", "ring-type", "spore-print-color", "population", "habitat"
]

# Завантаження даних з правильним заголовком
data = pd.read_csv(file_path, names=column_names)

# Перевірка перших кількох рядків даних
print(data.head())

# Кодуємо категоріальні змінні за допомогою LabelEncoder
encoder = LabelEncoder()
encoded_data = data.apply(encoder.fit_transform)

# Розподіл на ознаки та мітки
X = encoded_data.drop('class', axis=1)
y = encoded_data['class']

# Нормалізація ознак
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Розподіл на навчальну та тестову вибірки (80% навчання, 20% тестування)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Перевірка розмірів вибірок
print(f"Train shape: {X_train.shape}, Test shape: {X_test.shape}")

```

```

-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-35-1c24ba8d4cd9> in <cell line: 0>()
    14
    15 # Завантаження даних з правильним заголовком
--> 16 data = pd.read_csv(file_path, names=column_names)
    17
    18 # Перевірка перших кількох рядків даних

----- 4 frames -----
/usr/local/lib/python3.11/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression, memory_map,
is_text, errors, storage_options)
    871         if ioargs.encoding and "b" not in ioargs.mode:
    872             # Encoding
--> 873             handle = open(
    874                 handle,
    875                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory: 'mushroom.data'

```

```

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

# Створення архітектури нейронної мережі
model = Sequential([
    Dense(128, input_dim=X_train.shape[1], activation='relu'),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(1, activation='softmax') # Використовуємо softmax для багатокласової класифікації
])

# Скомпілюємо модель
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Огляд архітектури моделі
model.summary()

from tensorflow.keras.callbacks import EarlyStopping

# Ініціалізація раннього зупинення

```

```
early_stopping = EarlyStopping(monitor='val_loss', patience=5)

# Навчання моделі
history = model.fit(X_train, y_train, epochs=100, batch_size=32,
                    validation_split=0.2, callbacks=[early_stopping])

# Оцінка моделі на тестовій вибірці
test_loss, test_accuracy = model.evaluate(X_test, y_test)
print(f"Test accuracy: {test_accuracy:.4f}")

import gradio as gr
import numpy as np

# Функція для класифікації
def classify_mushroom(features):
    # Потрібно зробити передбачення для наданих характеристик гриба
    features_scaled = scaler.transform([features]) # Нормалізуємо введені дані
    prediction = model.predict(features_scaled)
    return np.argmax(prediction)

# Створення інтерфейсу
input_features = [gr.inputs.Slider(minimum=0, maximum=1, default=0, label=col) for col in column_names[1:]]
interface = gr.Interface(fn=classify_mushroom, inputs=input_features, outputs="text")

# Запуск інтерфейсу
interface.launch()
```