import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import accuracy\_score

from keras.models import Sequential

from keras.layers import Dense, Dropout

from keras.utils import to\_categorical

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/letter-recognition/letter-recognition.data"

names = ['letter', 'x-box', 'y-box', 'width', 'height', 'onpix', 'x-bar', 'y-bar', 'x2bar', 'y2bar', 'xybar', 'x2ybr', 'xy2br', 'x-ege', 'xegvy', 'y-ege', 'yegvx']

dataset = pd.read\_csv(url, names=names)

dataset.head(3)

# Split features and target variable

X = dataset.drop('letter', axis=1)

y = dataset['letter']

# Encode target variable

label\_encoder = LabelEncoder()

y = label\_encoder.fit\_transform(y)

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Convert labels to one-hot encoding

y\_train = to\_categorical(y\_train)

y\_test = to\_categorical(y\_test)

# Define the model

model = Sequential()

model.add(Dense(128, input\_dim=X\_train.shape[1], activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(64, activation='relu'))

model.add(Dense(26, activation='softmax')) # 26 classes for letters A-Z

# Compile the model

model.compile(loss='categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model

model.fit(X\_train, y\_train, epochs=20, batch\_size=32, validation\_data=(X\_test, y\_test))

# Evaluate the model

\_, accuracy = model.evaluate(X\_test, y\_test)

print('Accuracy:', accuracy)