!pip install yfinance --quiet

!pip install keras --quiet

!pip install tensorflow --quiet

import yfinance as yf

import numpy as np

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from keras.models import Sequential

from keras.layers import LSTM, Dense

data = yf.download('AAPL', start='2018-01-01', end='2024-12-31')['Close'].dropna()

print("Data downloaded:", data.shape)

scaler = MinMaxScaler()

scaled\_data = scaler.fit\_transform(data.values.reshape(-1, 1))

def create\_sequences(data, step=60):

    X, y = [], []

    for i in range(step, len(data)):

        X.append(data[i-step:i, 0])

        y.append(data[i, 0])

    return np.array(X), np.array(y)

X, y = create\_sequences(scaled\_data)

X = X.reshape(X.shape[0], X.shape[1], 1)

split = int(len(X) \* 0.8)

X\_train, X\_test = X[:split], X[split:]

y\_train, y\_test = y[:split], y[split:]

model = Sequential([

    LSTM(50, return\_sequences=False, input\_shape=(X\_train.shape[1], 1)),

    Dense(1)

])

model.compile(optimizer='adam', loss='mean\_squared\_error')

model.fit(X\_train, y\_train, epochs=10, batch\_size=64, verbose=1)

y\_pred = model.predict(X\_test)

y\_pred = scaler.inverse\_transform(y\_pred)

y\_test\_scaled = scaler.inverse\_transform(y\_test.reshape(-1, 1))

plt.figure(figsize=(12,6))

plt.plot(y\_test\_scaled, label='Actual Price', color='blue')

plt.plot(y\_pred, label='Predicted Price', color='red')

plt.title('Stock Price Prediction (AAPL)')

plt.xlabel('Time')

plt.ylabel('Stock Price (USD)')

plt.legend()

plt.grid(True)

plt.show()