gpt2_with_lstm_

January 11, 2025

[1]:
```
# Install required libraries
!pip install yfinance ccxt pandas numpy transformers matplotlib scikit-learn␣
 ↪tensorflow

# Import libraries
import yfinance as yf
import ccxt
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from transformers import AutoTokenizer, AutoModelForCausalLM
import matplotlib.pyplot as plt
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-
packages (0.2.51)
Collecting ccxt
  Downloading ccxt-4.4.47-py2.py3-none-any.whl.metadata (117 kB)
                           117.7/117.7
kB 3.9 MB/s eta 0:00:00
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-
packages (2.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(1.26.4)
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-
packages (4.47.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (3.10.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
packages (1.6.0)
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-
packages (2.17.1)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in
```

```
/usr/local/lib/python3.10/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (5.3.0)
Requirement already satisfied: platformdirs>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from yfinance) (4.3.6)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (2024.2)
Requirement already satisfied: frozendict>=2.3.4 in
/usr/local/lib/python3.10/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (3.17.8)
Requirement already satisfied: beautifulsoup4>=4.11.1 in
/usr/local/lib/python3.10/dist-packages (from yfinance) (4.12.3)
Requirement already satisfied: html5lib>=1.1 in /usr/local/lib/python3.10/dist-
packages (from yfinance) (1.1)
Requirement already satisfied: setuptools>=60.9.0 in
/usr/local/lib/python3.10/dist-packages (from ccxt) (75.1.0)
Requirement already satisfied: certifi>=2018.1.18 in
/usr/local/lib/python3.10/dist-packages (from ccxt) (2024.12.14)
Requirement already satisfied: cryptography>=2.6.1 in
/usr/local/lib/python3.10/dist-packages (from ccxt) (43.0.3)
Requirement already satisfied: typing-extensions>=4.4.0 in
/usr/local/lib/python3.10/dist-packages (from ccxt) (4.12.2)
Collecting aiohttp<=3.10.11 (from ccxt)
  Downloading aiohttp-3.10.11-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (7.7 kB)
Collecting aiodns>=1.1.1 (from ccxt)
  Downloading aiodns-3.2.0-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: yarl>=1.7.2 in /usr/local/lib/python3.10/dist-
packages (from ccxt) (1.18.3)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas) (2024.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from transformers) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.24.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.27.1)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.10/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: tokenizers<0.22,>=0.21 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.21.0)
Requirement already satisfied: safetensors>=0.4.1 in
/usr/local/lib/python3.10/dist-packages (from transformers) (0.5.1)
```

```
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-
packages (from transformers) (4.67.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.55.3)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.10/dist-
packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.2.1)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (24.12.23)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=3.10.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.12.1)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.3.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied:
protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3
in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.25.5)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.5.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (1.17.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
```

```
/usr/local/lib/python3.10/dist-packages (from tensorflow) (1.69.0)
Requirement already satisfied: tensorboard<2.18,>=2.17 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (2.17.1)
Requirement already satisfied: keras>=3.2.0 in /usr/local/lib/python3.10/dist-
packages (from tensorflow) (3.5.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.10/dist-packages (from tensorflow) (0.37.1)
Collecting pycares>=4.0.0 (from aiodns>=1.1.1->ccxt)
  Downloading pycares-4.5.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.1 kB)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<=3.10.11->ccxt) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<=3.10.11->ccxt) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp<=3.10.11->ccxt) (24.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<=3.10.11->ccxt) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<=3.10.11->ccxt) (6.1.0)
Requirement already satisfied: async-timeout<6.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp<=3.10.11->ccxt) (4.0.3)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow)
(0.45.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-
packages (from beautifulsoup4>=4.11.1->yfinance) (2.6)
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.10/dist-
packages (from cryptography>=2.6.1->ccxt) (1.17.1)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-
packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/usr/local/lib/python3.10/dist-packages (from huggingface-
hub<1.0,>=0.24.0->transformers) (2024.10.0)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
(from keras>=3.2.0->tensorflow) (0.13.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2.3.0)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.10/dist-packages (from
```

```
tensorboard<2.18,>=2.17->tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from
tensorboard<2.18,>=2.17->tensorflow) (3.1.3)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.10/dist-packages (from yarl>=1.7.2->ccxt) (0.2.1)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-
packages (from cffi>=1.12->cryptography>=2.6.1->ccxt) (2.22)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.10/dist-packages (from
werkzeug>=1.0.1->tensorboard<2.18,>=2.17->tensorflow) (3.0.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow)
(3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow)
(2.18.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->tensorflow) (0.1.2)
Downloading ccxt-4.4.47-py2.py3-none-any.whl (5.6 MB)
                         5.6/5.6 MB
23.6 MB/s eta 0:00:00
Downloading aiodns-3.2.0-py3-none-any.whl (5.7 kB)
Downloading
aiohttp-3.10.11-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2
MB)
                         1.2/1.2 MB
37.8 MB/s eta 0:00:00
Downloading
pycares-4.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (288
kB)
                         288.6/288.6 kB
17.7 MB/s eta 0:00:00
Installing collected packages: pycares, aiohttp, aiodns, ccxt
  Attempting uninstall: aiohttp
    Found existing installation: aiohttp 3.11.11
    Uninstalling aiohttp-3.11.11:
      Successfully uninstalled aiohttp-3.11.11
Successfully installed aiodns-3.2.0 aiohttp-3.10.11 ccxt-4.4.47 pycares-4.5.0
```

[2]:
```python
# Fetch stock data using Yahoo Finance
def fetch_stock_data(ticker, start_date='2020-01-01'):
    stock = yf.Ticker(ticker)
    df = stock.history(start=start_date)
```

```
    df.reset_index(inplace=True)
    df = df[['Date', 'Open', 'High', 'Low', 'Close', 'Volume']]
    df.columns = ['timestamp', 'open', 'high', 'low', 'close', 'volume']
    return df

# Fetch cryptocurrency data using CCXT
def fetch_crypto_data(symbol='BTC/USDT', exchange='kraken'):
    exchange_class = getattr(ccxt, exchange)
    exchange_instance = exchange_class()
    ohlcv = exchange_instance.fetch_ohlcv(symbol, timeframe='1d', limit=1000)
    df = pd.DataFrame(ohlcv, columns=['timestamp', 'open', 'high', 'low',␣
 ↪'close', 'volume'])
    df['timestamp'] = pd.to_datetime(df['timestamp'], unit='ms')
    return df

# Fetch data for all assets
apple_data = fetch_stock_data('AAPL', '2020-01-01')
tesla_data = fetch_stock_data('TSLA', '2020-01-01')
btc_data = fetch_crypto_data('BTC/USD')
eth_data = fetch_crypto_data('ETH/USD')

# Display data
print("Apple Data:\n", apple_data.head())
print("\nBitcoin Data:\n", btc_data.head())
```

```
Apple Data:
                  timestamp       open       high        low      close  \
0 2020-01-02 00:00:00-05:00  71.799866  72.856606  71.545380  72.796013
1 2020-01-03 00:00:00-05:00  72.020439  72.851769  71.862900  72.088303
2 2020-01-06 00:00:00-05:00  71.206077  72.701500  70.954010  72.662720
3 2020-01-07 00:00:00-05:00  72.672402  72.929314  72.100410  72.320969
4 2020-01-08 00:00:00-05:00  72.022843  73.787300  72.022843  73.484337

      volume
0  135480400
1  146322800
2  118387200
3  108872000
4  132079200

Bitcoin Data:
    timestamp     open     high      low    close       volume
0 2023-01-23  22717.1  23166.6  22520.1  22926.1  3015.649855
1 2023-01-24  22926.0  23158.7  22455.9  22633.8  3077.643596
2 2023-01-25  22636.0  23829.3  22320.0  23056.5  5020.204657
3 2023-01-26  23063.2  23293.3  22857.5  23010.6  3753.163921
4 2023-01-27  23010.6  23500.0  22492.8  23077.5  3420.533974
```

[3]:
```
# Add moving averages and volatility as features
def add_technical_indicators(data):
    data['ma_10'] = data['close'].rolling(window=10).mean()
    data['ma_20'] = data['close'].rolling(window=20).mean()
    data['volatility'] = data['close'].rolling(window=10).std()
    data = data.dropna()  # Drop rows with NaN values from rolling calculations
    return data

# Add indicators for all datasets
apple_data = add_technical_indicators(apple_data)
tesla_data = add_technical_indicators(tesla_data)
btc_data = add_technical_indicators(btc_data)
eth_data = add_technical_indicators(eth_data)
```

[4]:
```
# Preprocess data for LSTM
def preprocess_data(data, feature_cols=['close', 'ma_10', 'ma_20',␣
 ↪'volatility'], seq_length=60):
    scaler = MinMaxScaler()
    scaled_data = scaler.fit_transform(data[feature_cols])  # Scale features

    X, y = [], []
    for i in range(seq_length, len(scaled_data)):
        X.append(scaled_data[i-seq_length:i])  # Sequence input
        y.append(scaled_data[i, 0])  # Predict close price

    return np.array(X), np.array(y), scaler

# Preprocess all datasets
seq_length = 60
X_apple, y_apple, apple_scaler = preprocess_data(apple_data)
X_tesla, y_tesla, tesla_scaler = preprocess_data(tesla_data)
X_btc, y_btc, btc_scaler = preprocess_data(btc_data)
X_eth, y_eth, eth_scaler = preprocess_data(eth_data)
```

[5]:
```
# Build LSTM model
def build_lstm(input_shape):
    model = Sequential([
        LSTM(50, return_sequences=True, input_shape=input_shape),
        Dropout(0.2),
        LSTM(50, return_sequences=False),
        Dropout(0.2),
        Dense(25),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model
```

```python
# Train LSTM model
def train_lstm(model, X_train, y_train, X_test, y_test, epochs=10,␣
 ↪batch_size=32):
    history = model.fit(X_train, y_train, validation_data=(X_test, y_test),␣
 ↪epochs=epochs, batch_size=batch_size)
    return model, history

# Train LSTM models for all assets
def train_all_lstm_models():
    assets = {'Apple': (X_apple, y_apple), 'Tesla': (X_tesla, y_tesla),
              'Bitcoin': (X_btc, y_btc), 'Ethereum': (X_eth, y_eth)}
    models = {}

    for asset, (X, y) in assets.items():
        split_idx = int(len(X) * 0.8)
        X_train, X_test = X[:split_idx], X[split_idx:]
        y_train, y_test = y[:split_idx], y[split_idx:]

        print(f"\nTraining LSTM for {asset}...")
        model = build_lstm((seq_length, X.shape[2]))
        model, _ = train_lstm(model, X_train, y_train, X_test, y_test)
        models[asset] = (model, X_test, y_test)

    return models

lstm_models = train_all_lstm_models()
```

```
Training LSTM for Apple…

/usr/local/lib/python3.10/dist-packages/keras/src/layers/rnn/rnn.py:204:
UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When
using Sequential models, prefer using an `Input(shape)` object as the first
layer in the model instead.
  super().__init__(**kwargs)

Epoch 1/10
30/30             7s 57ms/step -
loss: 0.0331 - val_loss: 0.0154
Epoch 2/10
30/30             1s 13ms/step -
loss: 0.0044 - val_loss: 0.0064
Epoch 3/10
30/30             1s 9ms/step - loss:
0.0031 - val_loss: 0.0101
Epoch 4/10
30/30             0s 10ms/step -
loss: 0.0031 - val_loss: 0.0048
```

```
Epoch 5/10
30/30             0s 9ms/step - loss:
0.0028 - val_loss: 0.0048
Epoch 6/10
30/30             0s 11ms/step -
loss: 0.0022 - val_loss: 0.0036
Epoch 7/10
30/30             0s 10ms/step -
loss: 0.0021 - val_loss: 0.0054
Epoch 8/10
30/30             0s 9ms/step - loss:
0.0023 - val_loss: 0.0045
Epoch 9/10
30/30             1s 10ms/step -
loss: 0.0021 - val_loss: 0.0016
Epoch 10/10
30/30             1s 10ms/step -
loss: 0.0021 - val_loss: 0.0014

Training LSTM for Tesla…
Epoch 1/10
30/30             2s 19ms/step -
loss: 0.0676 - val_loss: 0.0074
Epoch 2/10
30/30             0s 13ms/step -
loss: 0.0076 - val_loss: 0.0068
Epoch 3/10
30/30             1s 13ms/step -
loss: 0.0051 - val_loss: 0.0043
Epoch 4/10
30/30             1s 14ms/step -
loss: 0.0045 - val_loss: 0.0043
Epoch 5/10
30/30             0s 15ms/step -
loss: 0.0043 - val_loss: 0.0030
Epoch 6/10
30/30             1s 15ms/step -
loss: 0.0038 - val_loss: 0.0039
Epoch 7/10
30/30             0s 13ms/step -
loss: 0.0038 - val_loss: 0.0028
Epoch 8/10
30/30             1s 9ms/step - loss:
0.0031 - val_loss: 0.0026
Epoch 9/10
30/30             0s 9ms/step - loss:
0.0032 - val_loss: 0.0025
Epoch 10/10
```

```
30/30             0s 9ms/step - loss:
0.0028 - val_loss: 0.0026

Training LSTM for Bitcoin…
Epoch 1/10
16/16             2s 30ms/step -
loss: 0.0392 - val_loss: 0.0623
Epoch 2/10
16/16             0s 12ms/step -
loss: 0.0070 - val_loss: 0.0211
Epoch 3/10
16/16             0s 12ms/step -
loss: 0.0045 - val_loss: 0.0072
Epoch 4/10
16/16             0s 11ms/step -
loss: 0.0031 - val_loss: 0.0076
Epoch 5/10
16/16             0s 10ms/step -
loss: 0.0028 - val_loss: 0.0065
Epoch 6/10
16/16             0s 10ms/step -
loss: 0.0023 - val_loss: 0.0079
Epoch 7/10
16/16             0s 10ms/step -
loss: 0.0022 - val_loss: 0.0074
Epoch 8/10
16/16             0s 12ms/step -
loss: 0.0021 - val_loss: 0.0113
Epoch 9/10
16/16             0s 10ms/step -
loss: 0.0025 - val_loss: 0.0146
Epoch 10/10
16/16             0s 10ms/step -
loss: 0.0022 - val_loss: 0.0098

Training LSTM for Ethereum…
Epoch 1/10
16/16             3s 42ms/step -
loss: 0.0780 - val_loss: 0.0262
Epoch 2/10
16/16             1s 21ms/step -
loss: 0.0121 - val_loss: 0.0114
Epoch 3/10
16/16             1s 21ms/step -
loss: 0.0102 - val_loss: 0.0088
Epoch 4/10
16/16             1s 19ms/step -
loss: 0.0090 - val_loss: 0.0079
```

```
Epoch 5/10
16/16             0s 21ms/step -
loss: 0.0079 - val_loss: 0.0075
Epoch 6/10
16/16             1s 23ms/step -
loss: 0.0071 - val_loss: 0.0075
Epoch 7/10
16/16             0s 26ms/step -
loss: 0.0078 - val_loss: 0.0071
Epoch 8/10
16/16             0s 25ms/step -
loss: 0.0080 - val_loss: 0.0075
Epoch 9/10
16/16             1s 19ms/step -
loss: 0.0056 - val_loss: 0.0065
Epoch 10/10
16/16             0s 16ms/step -
loss: 0.0069 - val_loss: 0.0099
```

[8]:
```python
# Load GPT-2 model and tokenizer
model_name = "gpt2"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model_gpt2 = AutoModelForCausalLM.from_pretrained(model_name)

# Set pad token to eos_token to handle padding properly
tokenizer.pad_token = tokenizer.eos_token

# Prepare sequences for GPT-2 with controlled input length
def prepare_gpt2_input(X, max_features=20):
    """
    Convert sequences to text format for GPT-2. Limit input length by␣
 ↪restricting the number of features.
    """
    return [" ".join(map(str, seq.flatten()[:max_features])) for seq in X]

# Prepare GPT-2 inputs for all assets
assets_gpt2_inputs = {
    "Apple": prepare_gpt2_input(X_apple[:10]),
    "Tesla": prepare_gpt2_input(X_tesla[:10]),
    "Bitcoin": prepare_gpt2_input(X_btc[:10]),
    "Ethereum": prepare_gpt2_input(X_eth[:10])
}

# GPT-2 Prediction function
def gpt2_predict(sequence, model, tokenizer, max_new_tokens=10,␣
 ↪max_input_length=1024):
    """
```

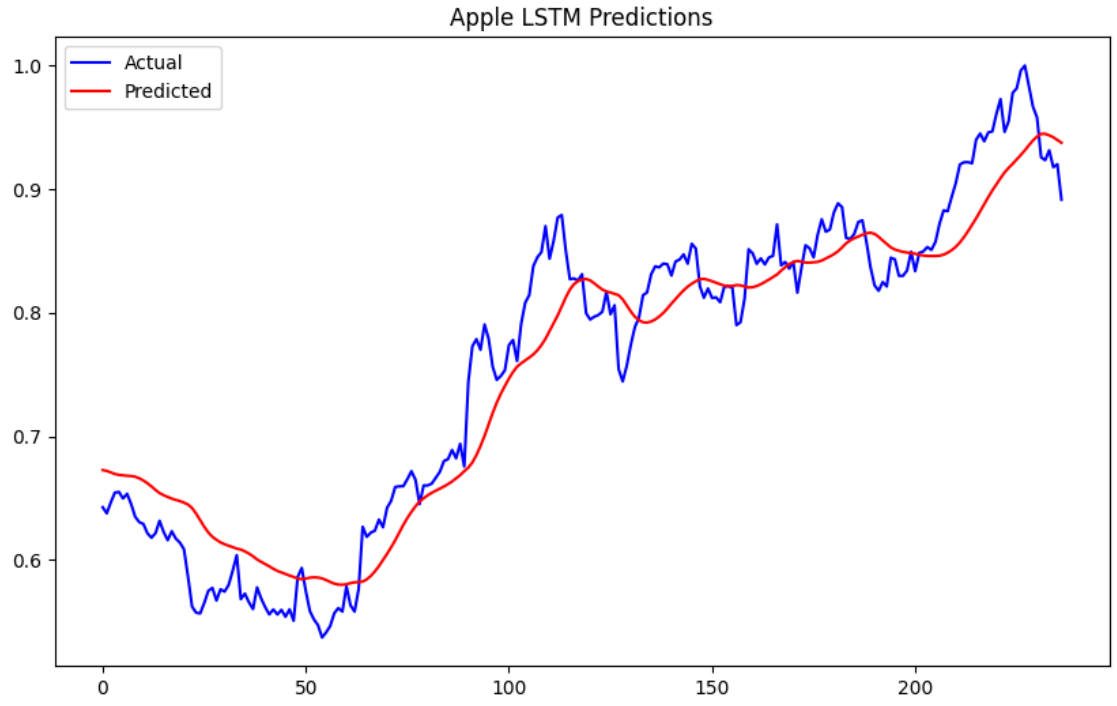
```
    Generate predictions using GPT-2 for a given input sequence.
    """
    # Tokenize the input sequence with truncation and padding
    inputs = tokenizer(sequence, return_tensors="pt", truncation=True,␣
 ↪max_length=max_input_length, padding=True)

    # Generate predictions
    outputs = model.generate(
        inputs["input_ids"],
        attention_mask=inputs["attention_mask"],
        max_new_tokens=max_new_tokens,
        pad_token_id=tokenizer.pad_token_id  # Ensure proper handling of padding
    )

    # Decode the output and extract the predicted value
    prediction = tokenizer.decode(outputs[0], skip_special_tokens=True)
    try:
        return float(prediction.split()[-1])  # Extract the last numeric value
    except ValueError:
        return None  # Return None if the prediction is not numeric

# Predict for all assets
gpt2_predictions = {asset: [gpt2_predict(seq, model_gpt2, tokenizer) for seq in␣
 ↪inputs]
                    for asset, inputs in assets_gpt2_inputs.items()}

# Display predictions
for asset, predictions in gpt2_predictions.items():
    print(f"{asset} GPT-2 Predictions:", predictions)
```

```
Apple GPT-2 Predictions: [0.098477514757514, 0.098477575757575,
0.098477514757575, 0.098477514757575, 0.098477514757575, 0.1034751401494909,
0.098477514757514, 0.088477514751475, 0.08908847751475, 0.08908847751475]
Tesla GPT-2 Predictions: [0.03901098477514, 0.03909847751455, 0.03901098477575,
0.03901098477514, 0.03901098477514, 0.03909847751401, 0.03909847757575,
0.064751475757575, 0.06814010984775, 0.03909847757514]
Bitcoin GPT-2 Predictions: [0.03909847757569, 0.010984775757575,
0.010984775140155, 0.03909847751401, 0.038160175140149, 0.010109847751401,
0.010109847751401, 0.04901098477514, 0.007514010984775, 0.01098477575897]
Ethereum GPT-2 Predictions: [0.03909088477514, 0.03909847751401,
0.03909847751401, 0.099571584152028, 0.099571584152028, 0.03909847995555,
0.03909847995555, 0.0390101010101, 0.03909847995514, 0.0674954179615905]
```

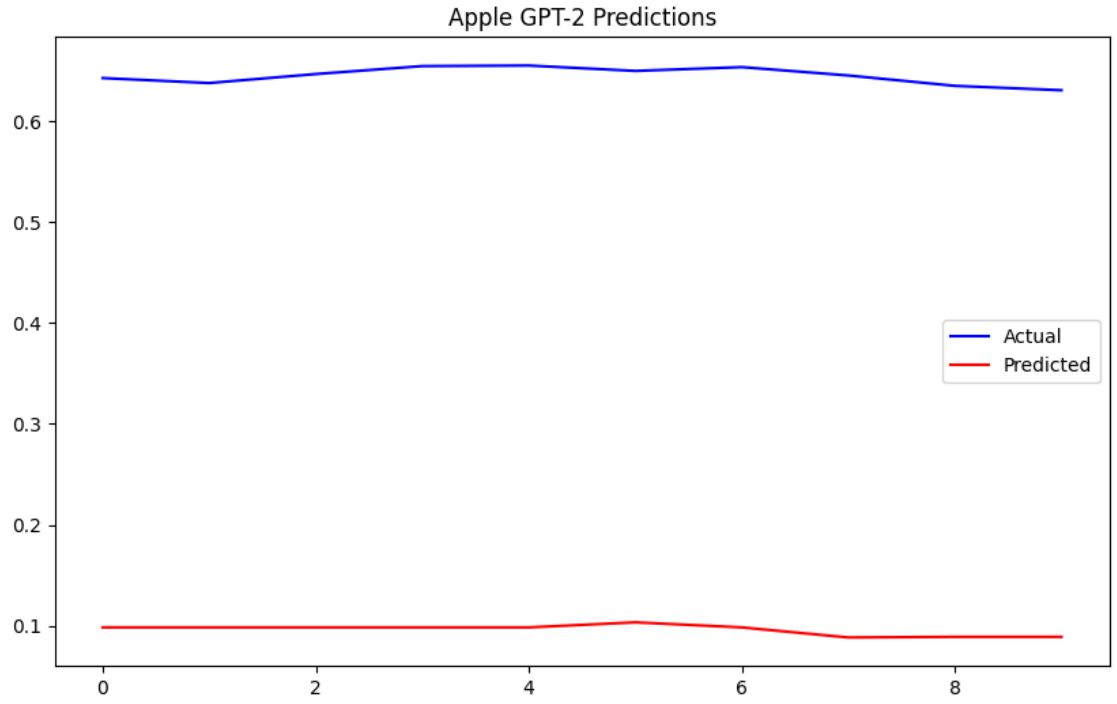
[9]:
```
# Evaluate predictions
def evaluate_predictions(actual, predictions, title="Evaluation"):
    mse = mean_squared_error(actual[:len(predictions)], predictions)
    rmse = np.sqrt(mse)
```

```
    print(f"{title} - MSE: {mse}, RMSE: {rmse}")
    plt.figure(figsize=(10, 6))
    plt.plot(actual[:len(predictions)], label="Actual", color="blue")
    plt.plot(predictions, label="Predicted", color="red")
    plt.title(title)
    plt.legend()
    plt.show()

# Evaluate LSTM and GPT-2 for all assets
for asset, (model, X_test, y_test) in lstm_models.items():
    print(f"\nEvaluating LSTM for {asset}...")
    lstm_predictions = model.predict(X_test)
    evaluate_predictions(y_test, lstm_predictions, title=f"{asset} LSTM␣
 ↪Predictions")

    print(f"\nEvaluating GPT-2 for {asset}...")
    evaluate_predictions(y_test[:len(gpt2_predictions[asset])],␣
 ↪gpt2_predictions[asset], title=f"{asset} GPT-2 Predictions")
```

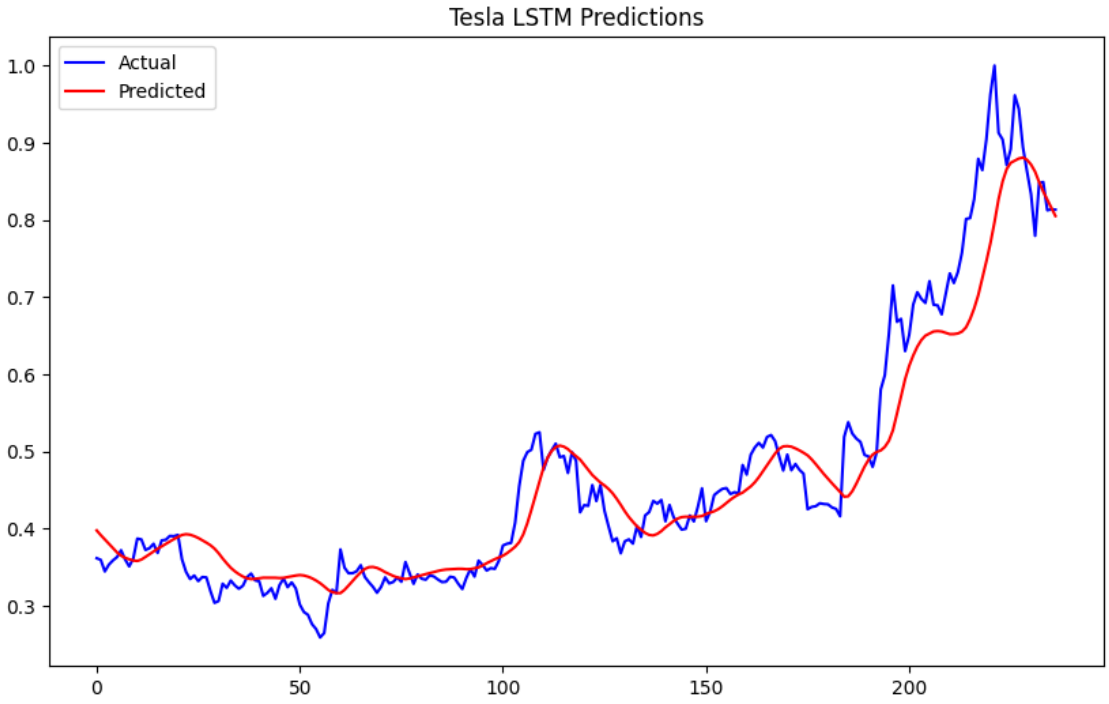
```
Evaluating LSTM for Apple…
8/8             1s 41ms/step
Apple LSTM Predictions - MSE: 0.0013595209814867123, RMSE: 0.0368716826506021
```



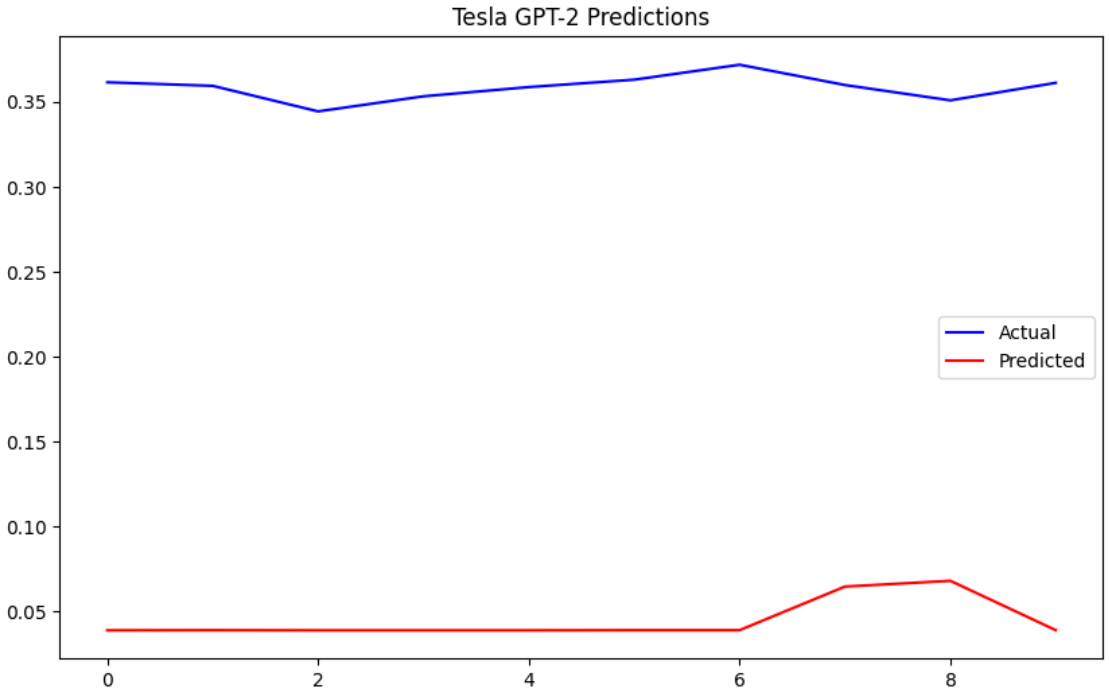
```
Evaluating GPT-2 for Apple…
Apple GPT-2 Predictions - MSE: 0.30145934998841606, RMSE: 0.5490531394941806
```



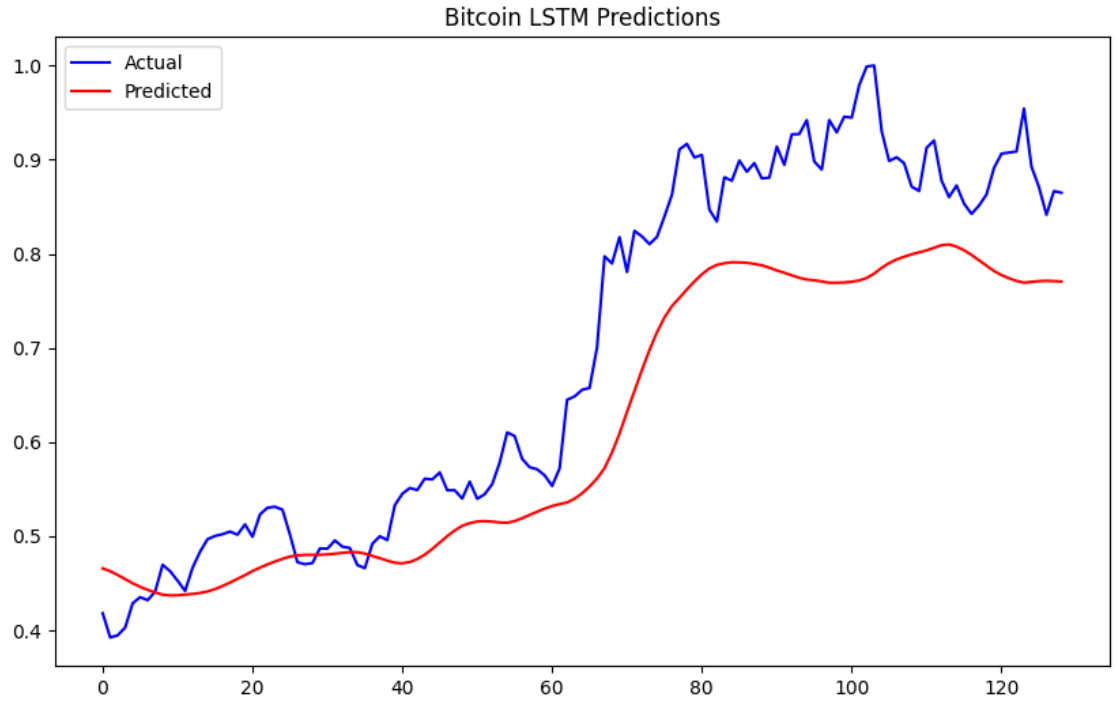

```
Evaluating LSTM for Tesla…
8/8             0s 38ms/step
Tesla LSTM Predictions - MSE: 0.0025606140840830622, RMSE: 0.050602510649997025
```



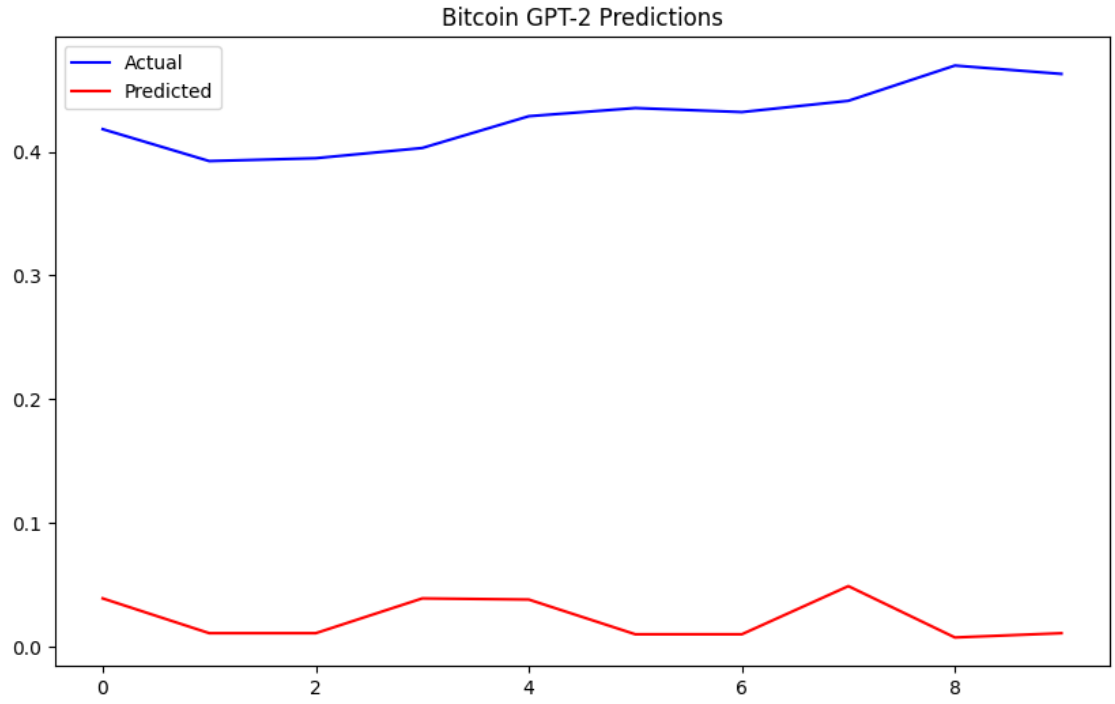

```
Evaluating GPT-2 for Tesla…
Tesla GPT-2 Predictions - MSE: 0.09874337581403328, RMSE: 0.3142345872338583
```



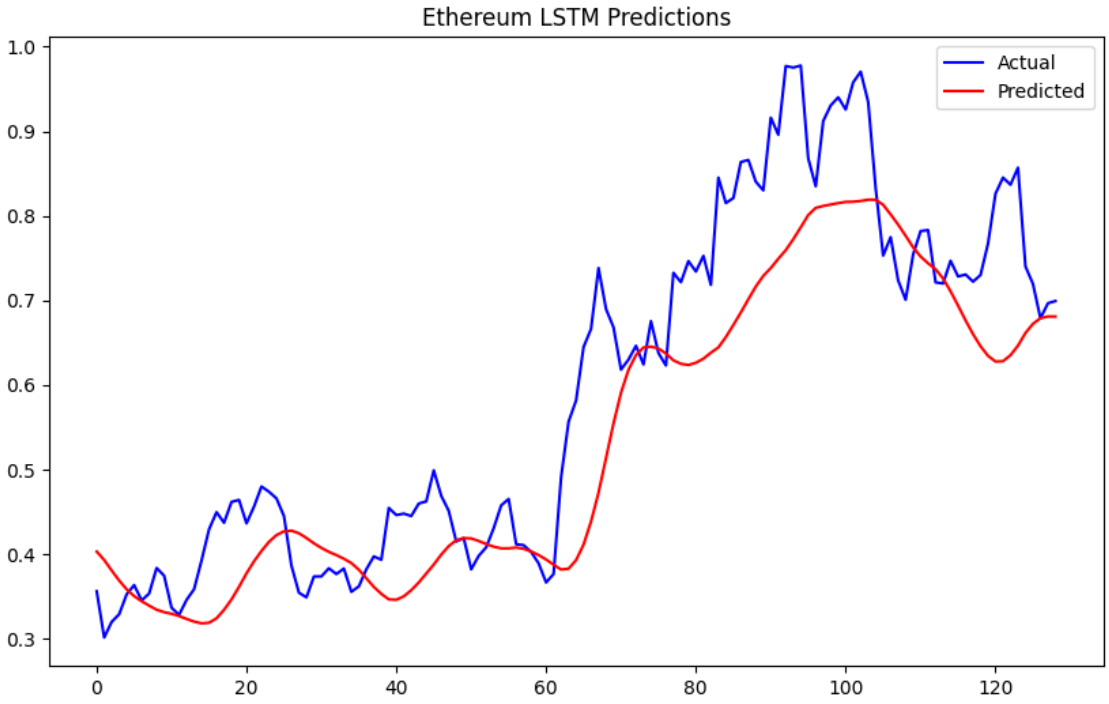
```
Evaluating LSTM for Bitcoin…

WARNING:tensorflow:5 out of the last 17 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x7e3ef0349b40> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly in a loop, (2) passing tensors with different shapes, (3) passing
Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that
can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

5/5             0s 62ms/step
Bitcoin LSTM Predictions - MSE: 0.00980266044385667, RMSE: 0.09900838572493074
```



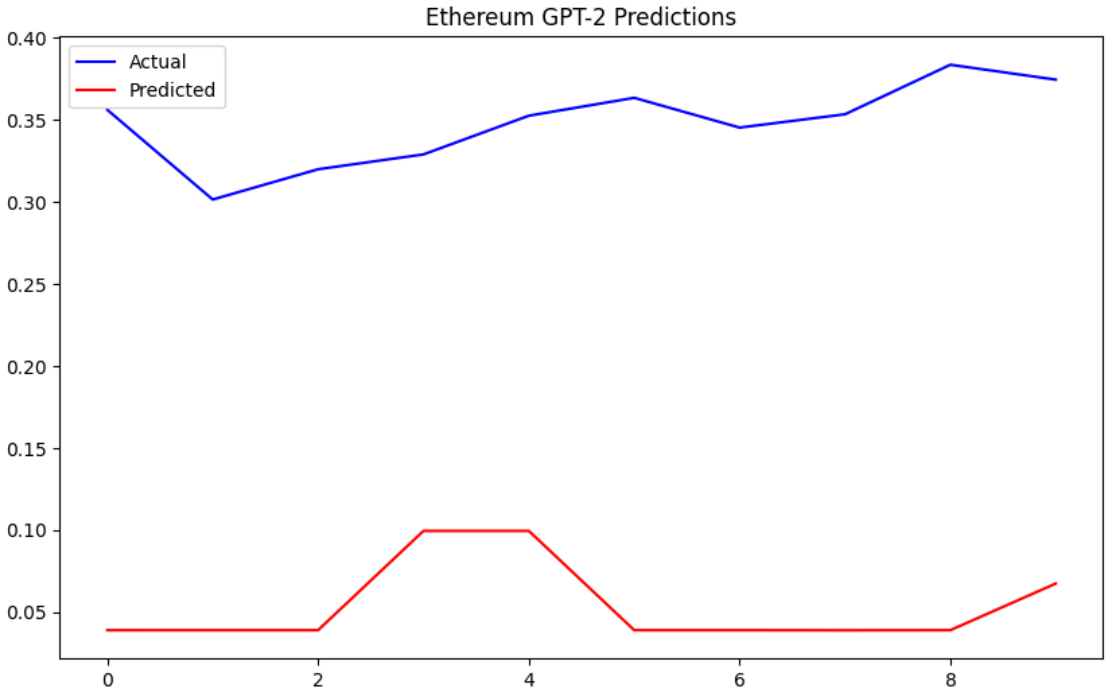

```
Evaluating GPT-2 for Bitcoin…
Bitcoin GPT-2 Predictions - MSE: 0.16519932523797076, RMSE: 0.40644719858546297
```




```
Evaluating LSTM for Ethereum…

WARNING:tensorflow:5 out of the last 14 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x7e3ef0243f40> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly in a loop, (2) passing tensors with different shapes, (3) passing
Python objects instead of tensors. For (1), please define your @tf.function
outside of the loop. For (2), @tf.function has reduce_retracing=True option that
can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

5/5             0s 66ms/step
Ethereum LSTM Predictions - MSE: 0.009905561957084474, RMSE: 0.09952668967208984
```




```
Evaluating GPT-2 for Ethereum…
Ethereum GPT-2 Predictions - MSE: 0.0875588981307763, RMSE: 0.2959035284189364
```



[ ]: