10January

January 10, 2025

[1]:

```
!pip install yfinance ccxt pandas numpy matplotlib scikit-learn tensorflow␣
  ↪transformers
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-
packages (0.2.51)
Requirement already satisfied: ccxt in /usr/local/lib/python3.10/dist-packages
(4.4.47)
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: 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: transformers in /usr/local/lib/python3.10/dist-
packages (4.47.1)
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)
Requirement already satisfied: aiohttp<=3.10.11 in
/usr/local/lib/python3.10/dist-packages (from ccxt) (3.10.11)
Requirement already satisfied: aiodns>=1.1.1 in /usr/local/lib/python3.10/dist-
packages (from ccxt) (3.2.0)
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: 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: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (24.2)
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)
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: 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: pycares>=4.0.0 in /usr/local/lib/python3.10/dist-
packages (from aiodns>=1.1.1->ccxt) (4.5.0)
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) (2024.9.0)
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)
```

[2]:

```python
# Import libraries
import yfinance as yf
import ccxt
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
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 pipeline
```

[4]:

```python
# Function to fetch stock data
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

def fetch_crypto_data(symbol='BTC/USD', 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

# Example usage
btc_data = fetch_crypto_data('BTC/USD', exchange='kraken')
eth_data = fetch_crypto_data('ETH/USD', exchange='kraken')
print(btc_data.head())


# Fetch data for Apple, Tesla, Bitcoin, and Ethereum
apple_data = fetch_stock_data('AAPL', '2020-01-01')
tesla_data = fetch_stock_data('TSLA', '2020-01-01')
btc_data = fetch_crypto_data('BTC/USDT')
eth_data = fetch_crypto_data('ETH/USDT')
```

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

```
   timestamp     open     high      low    close       volume
0 2023-01-22  22785.6  23108.5  22301.2  22717.1  3113.012916
1 2023-01-23  22717.1  23166.6  22520.1  22926.1  3015.649855
2 2023-01-24  22926.0  23158.7  22455.9  22633.8  3077.643596
3 2023-01-25  22636.0  23829.3  22320.0  23056.5  5020.204657
4 2023-01-26  23063.2  23293.3  22857.5  23010.6  3753.163921
Apple Data:
                   timestamp       open       high        low      close  \
0 2020-01-02 00:00:00-05:00  71.799881  72.856621  71.545395  72.796028
1 2020-01-03 00:00:00-05:00  72.020432  72.851761  71.862892  72.088295
2 2020-01-06 00:00:00-05:00  71.206085  72.701508  70.954017  72.662727
3 2020-01-07 00:00:00-05:00  72.672409  72.929322  72.100418  72.320976
4 2020-01-08 00:00:00-05:00  72.022858  73.787315  72.022858  73.484352

      volume
0  135480400
1  146322800
2  118387200
3  108872000
4  132079200

Bitcoin Data:
    timestamp     open     high      low    close      volume
0 2023-01-22  22762.5  23063.4  22272.9  22714.8  171.410342
1 2023-01-23  22705.0  23160.2  22495.0  22922.6  408.255323
2 2023-01-24  22921.2  23160.1  22481.9  22625.4  310.378027
3 2023-01-25  22612.8  23791.9  22343.0  23060.8  449.810482
4 2023-01-26  23077.6  23263.6  22850.0  23012.9  540.221352
```

[5]:

```python
# Load FinBERT sentiment analysis pipeline
sentiment_pipeline = pipeline("sentiment-analysis", model="ProsusAI/finbert")

# Function to add sentiment scores
def add_sentiment_to_data(data):
    # Placeholder sentiment text for demonstration
    data['sentiment_text'] = "The market sentiment is positive."  # Example text
    data['sentiment_score'] = [
        sentiment_pipeline(text)[0]['score'] for text in data['sentiment_text']
    ]
    return data

# Add sentiment scores
apple_data = add_sentiment_to_data(apple_data)
```

```python
tesla_data = add_sentiment_to_data(tesla_data)
btc_data = add_sentiment_to_data(btc_data)
eth_data = add_sentiment_to_data(eth_data)

# Show sample data with sentiment scores
print("\nApple Data with Sentiment:\n", apple_data.head())
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_auth.py:94:
UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab
(https://huggingface.co/settings/tokens), set it as secret in your Google Colab
and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
public models or datasets.
  warnings.warn(

config.json:   0%|          | 0.00/758 [00:00<?, ?B/s]

pytorch_model.bin:   0%|          | 0.00/438M [00:00<?, ?B/s]

tokenizer_config.json:   0%|          | 0.00/252 [00:00<?, ?B/s]

vocab.txt:   0%|          | 0.00/232k [00:00<?, ?B/s]

special_tokens_map.json:   0%|          | 0.00/112 [00:00<?, ?B/s]

Device set to use cuda:0
You seem to be using the pipelines sequentially on GPU. In order to maximize
efficiency please use a dataset


Apple Data with Sentiment:
                   timestamp       open       high        low      close  \
0 2020-01-02 00:00:00-05:00  71.799881  72.856621  71.545395  72.796028
1 2020-01-03 00:00:00-05:00  72.020432  72.851761  71.862892  72.088295
2 2020-01-06 00:00:00-05:00  71.206085  72.701508  70.954017  72.662727
3 2020-01-07 00:00:00-05:00  72.672409  72.929322  72.100418  72.320976
4 2020-01-08 00:00:00-05:00  72.022858  73.787315  72.022858  73.484352

      volume                     sentiment_text  sentiment_score
0  135480400  The market sentiment is positive.         0.807533
1  146322800  The market sentiment is positive.         0.807533
2  118387200  The market sentiment is positive.         0.807533
3  108872000  The market sentiment is positive.         0.807533
4  132079200  The market sentiment is positive.         0.807533
```

[7]:

```python
from sklearn.preprocessing import MinMaxScaler
import numpy as np
```

```
# Function to preprocess data
def preprocess_data(data, feature_cols=['close', 'sentiment_score'],␣
 ↪seq_length=60):
    """
    Scales the selected features and creates sequences of `seq_length`.
    Returns the preprocessed data as X (input), y (output), and the scaler␣
 ↪object.
    """
    scaler = MinMaxScaler()
    scaled_data = scaler.fit_transform(data[feature_cols])  # Scale selected␣
 ↪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

# Sequence length
seq_length = 60

# Preprocess Apple, Tesla, Bitcoin, and Ethereum datasets
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)

# Print dataset shapes for verification
print(f"Apple Data: X shape {X_apple.shape}, y shape {y_apple.shape}")
print(f"Tesla Data: X shape {X_tesla.shape}, y shape {y_tesla.shape}")
print(f"Bitcoin Data: X shape {X_btc.shape}, y shape {y_btc.shape}")
print(f"Ethereum Data: X shape {X_eth.shape}, y shape {y_eth.shape}")
```

```
Apple Data: X shape (1204, 60, 2), y shape (1204,)
Tesla Data: X shape (1204, 60, 2), y shape (1204,)
Bitcoin Data: X shape (660, 60, 2), y shape (660,)
Ethereum Data: X shape (660, 60, 2), y shape (660,)
```

[8]:

```
# Function to split data into training and testing sets
def train_test_split(X, y, train_ratio=0.8):
    split_idx = int(len(X) * train_ratio)
    return X[:split_idx], X[split_idx:], y[:split_idx], y[split_idx:]

# Split data
X_train_apple, X_test_apple, y_train_apple, y_test_apple =␣
 ↪train_test_split(X_apple, y_apple)
```

```
X_train_tesla, X_test_tesla, y_train_tesla, y_test_tesla =␣
 ↪train_test_split(X_tesla, y_tesla)
X_train_btc, X_test_btc, y_train_btc, y_test_btc = train_test_split(X_btc,␣
 ↪y_btc)
X_train_eth, X_test_eth, y_train_eth, y_test_eth = train_test_split(X_eth,␣
 ↪y_eth)
```

[9]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

# Function to build and train LSTM model
def build_and_train_lstm(X_train, y_train, X_test, y_test, epochs=20,␣
 ↪batch_size=32):
    model = Sequential([
        LSTM(50, return_sequences=True, input_shape=(X_train.shape[1], X_train.
 ↪shape[2])),
        Dropout(0.2),
        LSTM(50, return_sequences=False),
        Dropout(0.2),
        Dense(25),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mean_squared_error')
    history = model.fit(X_train, y_train, validation_data=(X_test, y_test),␣
 ↪epochs=epochs, batch_size=batch_size)
    return model, history

# Train models for all datasets
apple_model, _ = build_and_train_lstm(X_train_apple, y_train_apple,␣
 ↪X_test_apple, y_test_apple)
tesla_model, _ = build_and_train_lstm(X_train_tesla, y_train_tesla,␣
 ↪X_test_tesla, y_test_tesla)
btc_model, _ = build_and_train_lstm(X_train_btc, y_train_btc, X_test_btc,␣
 ↪y_test_btc)
eth_model, _ = build_and_train_lstm(X_train_eth, y_train_eth, X_test_eth,␣
 ↪y_test_eth)
```

```
/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 1/20
31/31                 9s 39ms/step -
loss: 0.0692 - val_loss: 0.0065
Epoch 2/20
```

```
31/31             0s 9ms/step - loss:
0.0056 - val_loss: 0.0099
Epoch 3/20
31/31             0s 9ms/step - loss:
0.0038 - val_loss: 0.0045
Epoch 4/20
31/31             0s 9ms/step - loss:
0.0028 - val_loss: 0.0023
Epoch 5/20
31/31             0s 10ms/step -
loss: 0.0023 - val_loss: 0.0018
Epoch 6/20
31/31             0s 9ms/step - loss:
0.0028 - val_loss: 0.0012
Epoch 7/20
31/31             0s 10ms/step -
loss: 0.0021 - val_loss: 0.0021
Epoch 8/20
31/31             1s 9ms/step - loss:
0.0021 - val_loss: 0.0036
Epoch 9/20
31/31             0s 9ms/step - loss:
0.0021 - val_loss: 0.0010
Epoch 10/20
31/31             0s 9ms/step - loss:
0.0019 - val_loss: 0.0016
Epoch 11/20
31/31             0s 9ms/step - loss:
0.0018 - val_loss: 0.0028
Epoch 12/20
31/31             0s 9ms/step - loss:
0.0016 - val_loss: 0.0013
Epoch 13/20
31/31             0s 9ms/step - loss:
0.0019 - val_loss: 0.0042
Epoch 14/20
31/31             0s 11ms/step -
loss: 0.0016 - val_loss: 0.0012
Epoch 15/20
31/31             1s 9ms/step - loss:
0.0015 - val_loss: 0.0010
Epoch 16/20
31/31             0s 9ms/step - loss:
0.0015 - val_loss: 0.0018
Epoch 17/20
31/31             0s 9ms/step - loss:
0.0013 - val_loss: 8.1364e-04
Epoch 18/20
```

```
31/31            0s 9ms/step - loss:
0.0013 - val_loss: 0.0011
Epoch 19/20
31/31            1s 9ms/step - loss:
0.0012 - val_loss: 8.5347e-04
Epoch 20/20
31/31            0s 9ms/step - loss:
0.0012 - val_loss: 0.0015
Epoch 1/20
31/31            2s 26ms/step -
loss: 0.0699 - val_loss: 0.0062
Epoch 2/20
31/31            1s 24ms/step -
loss: 0.0061 - val_loss: 0.0039
Epoch 3/20
31/31            1s 31ms/step -
loss: 0.0040 - val_loss: 0.0036
Epoch 4/20
31/31            1s 25ms/step -
loss: 0.0034 - val_loss: 0.0048
Epoch 5/20
31/31            1s 15ms/step -
loss: 0.0041 - val_loss: 0.0036
Epoch 6/20
31/31            0s 10ms/step -
loss: 0.0034 - val_loss: 0.0031
Epoch 7/20
31/31            0s 10ms/step -
loss: 0.0035 - val_loss: 0.0032
Epoch 8/20
31/31            1s 9ms/step - loss:
0.0033 - val_loss: 0.0024
Epoch 9/20
31/31            0s 10ms/step -
loss: 0.0029 - val_loss: 0.0029
Epoch 10/20
31/31            0s 12ms/step -
loss: 0.0025 - val_loss: 0.0027
Epoch 11/20
31/31            1s 24ms/step -
loss: 0.0032 - val_loss: 0.0023
Epoch 12/20
31/31            1s 19ms/step -
loss: 0.0024 - val_loss: 0.0021
Epoch 13/20
31/31            1s 24ms/step -
loss: 0.0023 - val_loss: 0.0024
Epoch 14/20
```

```
31/31             1s 16ms/step -
loss: 0.0021 - val_loss: 0.0020
Epoch 15/20
31/31             1s 14ms/step -
loss: 0.0019 - val_loss: 0.0019
Epoch 16/20
31/31             1s 16ms/step -
loss: 0.0018 - val_loss: 0.0021
Epoch 17/20
31/31             0s 14ms/step -
loss: 0.0023 - val_loss: 0.0018
Epoch 18/20
31/31             1s 14ms/step -
loss: 0.0019 - val_loss: 0.0018
Epoch 19/20
31/31             1s 19ms/step -
loss: 0.0018 - val_loss: 0.0018
Epoch 20/20
31/31             1s 19ms/step -
loss: 0.0017 - val_loss: 0.0016
Epoch 1/20
17/17             4s 27ms/step -
loss: 0.0473 - val_loss: 0.0689
Epoch 2/20
17/17             0s 15ms/step -
loss: 0.0082 - val_loss: 0.0059
Epoch 3/20
17/17             0s 12ms/step -
loss: 0.0027 - val_loss: 0.0040
Epoch 4/20
17/17             0s 10ms/step -
loss: 0.0026 - val_loss: 0.0042
Epoch 5/20
17/17             0s 10ms/step -
loss: 0.0023 - val_loss: 0.0043
Epoch 6/20
17/17             0s 12ms/step -
loss: 0.0020 - val_loss: 0.0064
Epoch 7/20
17/17             0s 10ms/step -
loss: 0.0017 - val_loss: 0.0031
Epoch 8/20
17/17             0s 11ms/step -
loss: 0.0014 - val_loss: 0.0027
Epoch 9/20
17/17             0s 10ms/step -
loss: 0.0019 - val_loss: 0.0027
Epoch 10/20
```

```
17/17            0s 10ms/step -
loss: 0.0017 - val_loss: 0.0036
Epoch 11/20
17/17            0s 11ms/step -
loss: 0.0019 - val_loss: 0.0048
Epoch 12/20
17/17            0s 10ms/step -
loss: 0.0014 - val_loss: 0.0069
Epoch 13/20
17/17            0s 11ms/step -
loss: 0.0019 - val_loss: 0.0063
Epoch 14/20
17/17            0s 10ms/step -
loss: 0.0016 - val_loss: 0.0095
Epoch 15/20
17/17            0s 10ms/step -
loss: 0.0016 - val_loss: 0.0042
Epoch 16/20
17/17            0s 10ms/step -
loss: 0.0015 - val_loss: 0.0033
Epoch 17/20
17/17            0s 11ms/step -
loss: 0.0016 - val_loss: 0.0044
Epoch 18/20
17/17            0s 10ms/step -
loss: 0.0013 - val_loss: 0.0041
Epoch 19/20
17/17            0s 11ms/step -
loss: 0.0012 - val_loss: 0.0057
Epoch 20/20
17/17            0s 10ms/step -
loss: 0.0014 - val_loss: 0.0037
Epoch 1/20
17/17            2s 29ms/step -
loss: 0.1087 - val_loss: 0.0304
Epoch 2/20
17/17            0s 15ms/step -
loss: 0.0147 - val_loss: 0.0073
Epoch 3/20
17/17            0s 17ms/step -
loss: 0.0065 - val_loss: 0.0067
Epoch 4/20
17/17            1s 16ms/step -
loss: 0.0063 - val_loss: 0.0072
Epoch 5/20
17/17            0s 18ms/step -
loss: 0.0062 - val_loss: 0.0066
Epoch 6/20
```

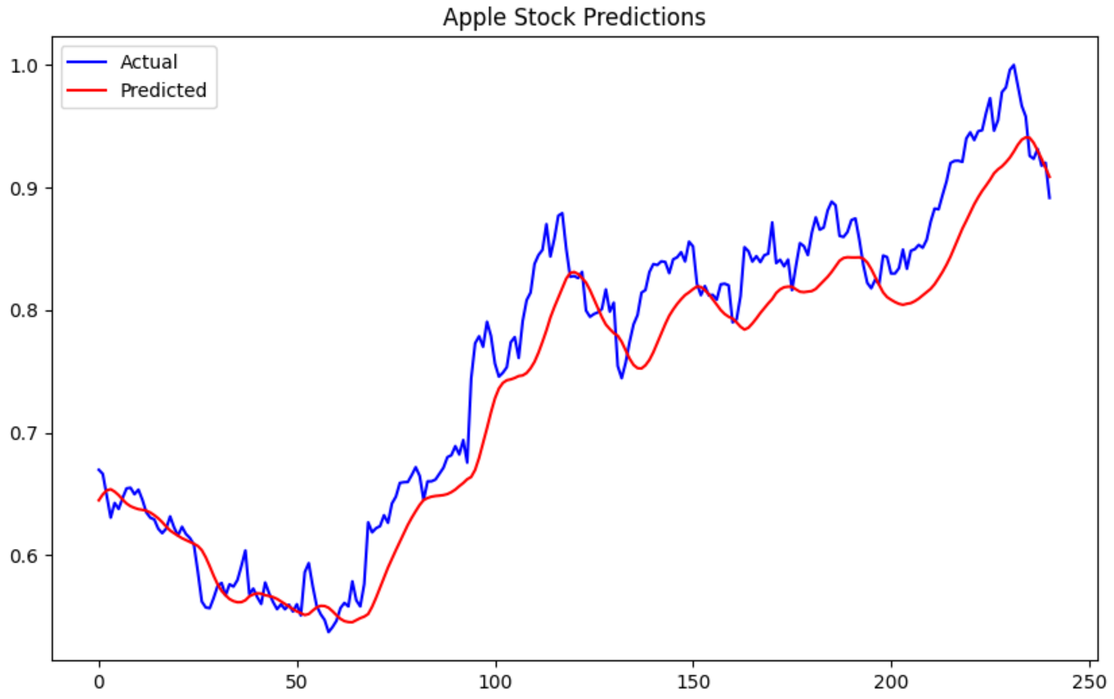
```
17/17                 1s 16ms/step -
loss: 0.0054 - val_loss: 0.0064
Epoch 7/20
17/17                 0s 13ms/step -
loss: 0.0055 - val_loss: 0.0061
Epoch 8/20
17/17                 0s 11ms/step -
loss: 0.0039 - val_loss: 0.0056
Epoch 9/20
17/17                 0s 10ms/step -
loss: 0.0049 - val_loss: 0.0051
Epoch 10/20
17/17                 0s 10ms/step -
loss: 0.0051 - val_loss: 0.0052
Epoch 11/20
17/17                 0s 10ms/step -
loss: 0.0047 - val_loss: 0.0056
Epoch 12/20
17/17                 0s 10ms/step -
loss: 0.0045 - val_loss: 0.0069
Epoch 13/20
17/17                 0s 10ms/step -
loss: 0.0055 - val_loss: 0.0046
Epoch 14/20
17/17                 0s 10ms/step -
loss: 0.0044 - val_loss: 0.0047
Epoch 15/20
17/17                 0s 10ms/step -
loss: 0.0051 - val_loss: 0.0044
Epoch 16/20
17/17                 0s 10ms/step -
loss: 0.0036 - val_loss: 0.0044
Epoch 17/20
17/17                 0s 10ms/step -
loss: 0.0040 - val_loss: 0.0054
Epoch 18/20
17/17                 0s 10ms/step -
loss: 0.0040 - val_loss: 0.0043
Epoch 19/20
17/17                 0s 11ms/step -
loss: 0.0045 - val_loss: 0.0047
Epoch 20/20
17/17                 0s 10ms/step -
loss: 0.0035 - val_loss: 0.0044
```

[10]:

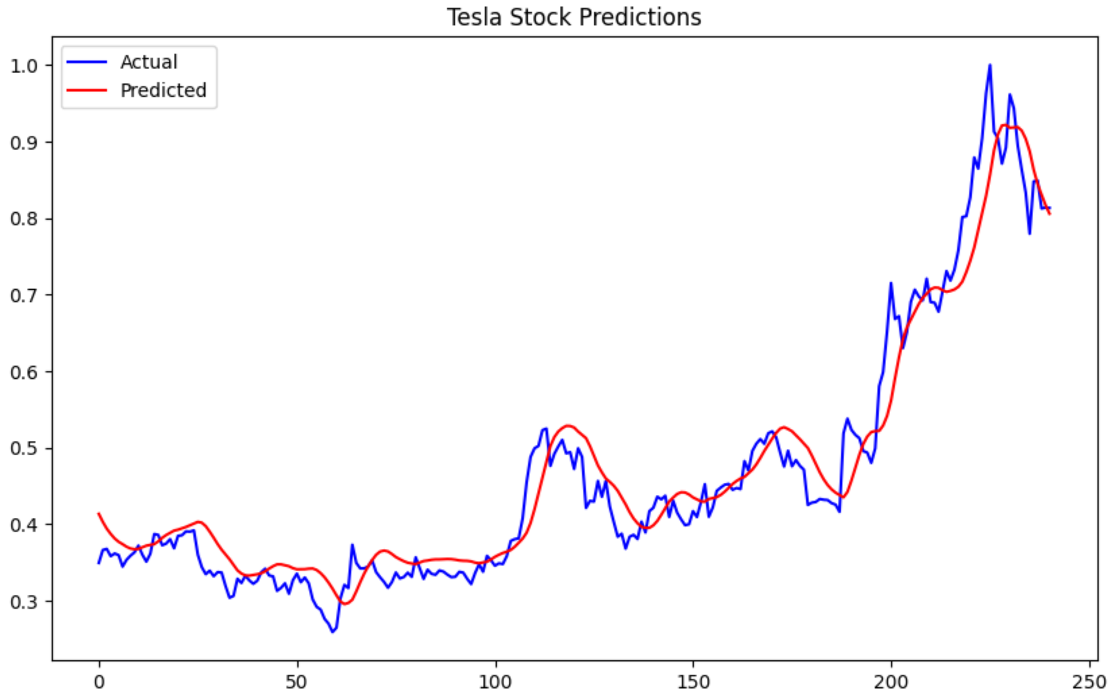
```python
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
```

```
# Function to evaluate and visualize predictions
def evaluate_model(model, X_test, y_test, title):
    predictions = model.predict(X_test)
    mse = mean_squared_error(y_test, predictions)
    rmse = np.sqrt(mse)
    print(f"{title} - MSE: {mse}, RMSE: {rmse}")

    plt.figure(figsize=(10, 6))
    plt.plot(y_test, label='Actual', color='blue')
    plt.plot(predictions, label='Predicted', color='red')
    plt.title(title)
    plt.legend()
    plt.show()

# Evaluate Models
evaluate_model(apple_model, X_test_apple, y_test_apple, "Apple Stock␣
 ↪Predictions")
evaluate_model(tesla_model, X_test_tesla, y_test_tesla, "Tesla Stock␣
 ↪Predictions")
evaluate_model(btc_model, X_test_btc, y_test_btc, "Bitcoin Predictions")
evaluate_model(eth_model, X_test_eth, y_test_eth, "Ethereum Predictions")
```

```
8/8              1s 68ms/step
Apple Stock Predictions - MSE: 0.0014648364270170866, RMSE: 0.03827318156381942
```

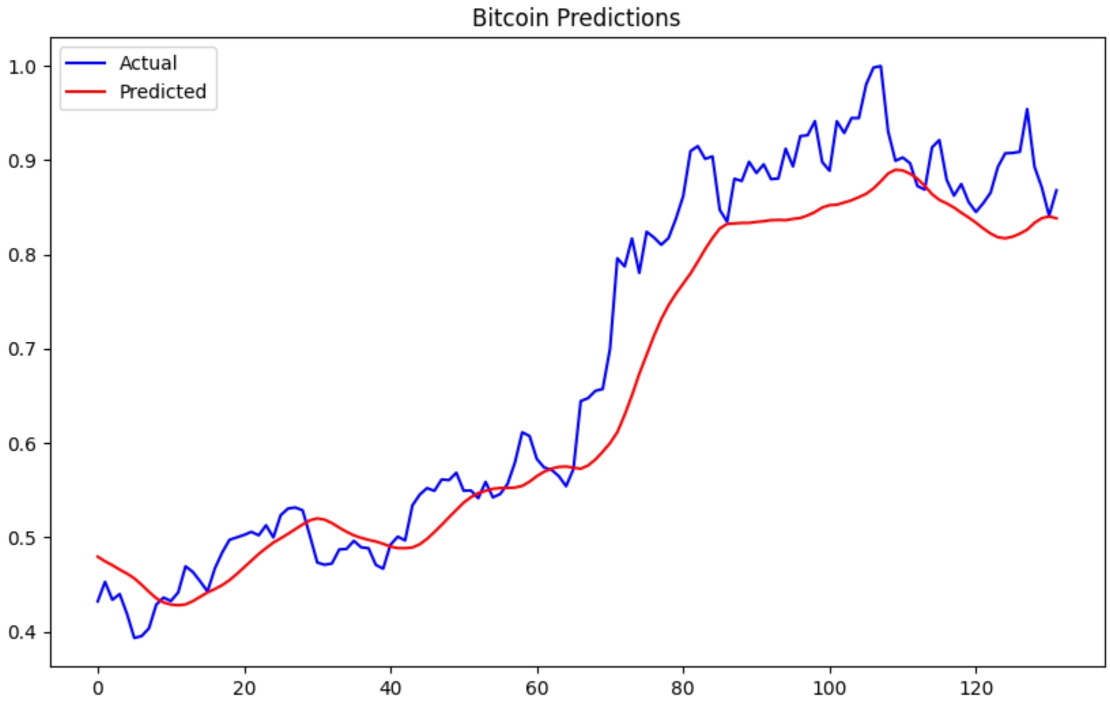


```
8/8            1s 76ms/step
Tesla Stock Predictions - MSE: 0.0016133169430655016, RMSE: 0.04016611685320728
```



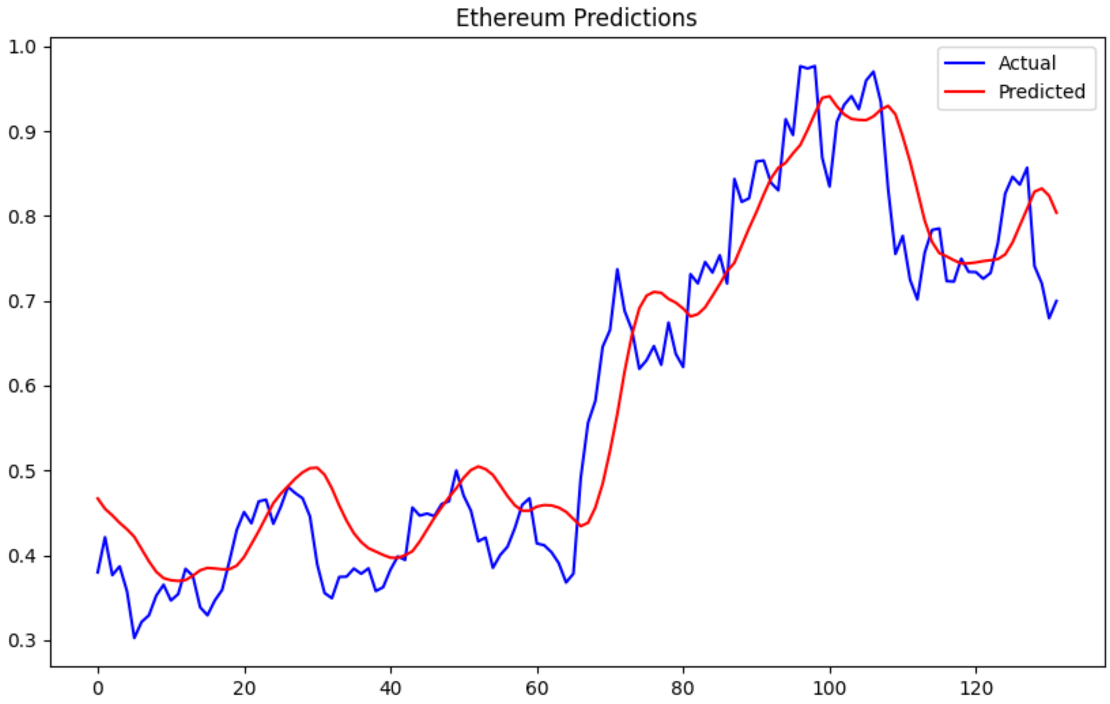
```
WARNING:tensorflow:5 out of the last 17 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x7a633d49edd0> triggered tf.function retracing. Tracing is expensive and the
excessive number of tracings could be due to (1) creating @tf.function
repeatedly 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 63ms/step
Bitcoin Predictions - MSE: 0.0036622995586673437, RMSE: 0.06051693613086624
```



```
WARNING:tensorflow:5 out of the last 14 calls to <function
TensorFlowTrainer.make_predict_function.<locals>.one_step_on_data_distributed at
0x7a633d3adab0> 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              1s 68ms/step
Ethereum Predictions - MSE: 0.0043858796111687385, RMSE: 0.06622597384084841
```


Ethereum Predictions
Actual
Predicted
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100
120

[ ]: