In [1]:
```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense
```

Stacked LSTM

In [2]:
```python
file_path = (r'C:\Users\begba\Desktop\praktika-master\PIB.xlsx')
data = pd.read_excel(file_path)
```

In [3]:
```python
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data)
```

In [4]:
```python
train_size = int(len(scaled_data) * 0.8)
test_size = len(scaled_data) - train_size
train_data, test_data = scaled_data[0:train_size,:], scaled_data[train_size:len(scaled_d
```

In [5]:
```python
def create_dataset(dataset, time_steps=1):
    X, Y = [], []
    for i in range(len(dataset)-time_steps-1):
        a = dataset[i:(i+time_steps), 0]
        X.append(a)
        Y.append(dataset[i + time_steps, 0])
    return np.array(X), np.array(Y)
```

In [6]:
```python
time_steps = 1
X_train, Y_train = create_dataset(train_data, time_steps)
X_test, Y_test = create_dataset(test_data, time_steps)
```

In [7]:
```python
# Reshape input to be [samples, time steps, features]
X_train = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_test = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
```

In [8]:
```python
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(1, time_steps)))
model.add(LSTM(units=50))
model.add(Dense(units=1))
```

In [9]:
```python
model.compile(optimizer='adam', loss='mean_squared_error')
```

In [10]:
```python
model.fit(X_train, Y_train, epochs=100, batch_size=1, verbose=2)
```

```
Epoch 1/100
14/14 - 9s - loss: 0.1887 - 9s/epoch - 636ms/step
Epoch 2/100
14/14 - 0s - loss: 0.1467 - 88ms/epoch - 6ms/step
Epoch 3/100
14/14 - 0s - loss: 0.1053 - 81ms/epoch - 6ms/step
Epoch 4/100
14/14 - 0s - loss: 0.0662 - 77ms/epoch - 6ms/step
Epoch 5/100
14/14 - 0s - loss: 0.0463 - 86ms/epoch - 6ms/step
Epoch 6/100
14/14 - 0s - loss: 0.0355 - 135ms/epoch - 10ms/step
Epoch 7/100
14/14 - 0s - loss: 0.0331 - 71ms/epoch - 5ms/step
Epoch 8/100
14/14 - 0s - loss: 0.0309 - 71ms/epoch - 5ms/step
Epoch 9/100
```

```
14/14 - 0s - loss: 0.0276 - 85ms/epoch - 6ms/step
Epoch 10/100
14/14 - 0s - loss: 0.0253 - 79ms/epoch - 6ms/step
Epoch 11/100
14/14 - 0s - loss: 0.0227 - 67ms/epoch - 5ms/step
Epoch 12/100
14/14 - 0s - loss: 0.0202 - 83ms/epoch - 6ms/step
Epoch 13/100
14/14 - 0s - loss: 0.0184 - 73ms/epoch - 5ms/step
Epoch 14/100
14/14 - 0s - loss: 0.0168 - 80ms/epoch - 6ms/step
Epoch 15/100
14/14 - 0s - loss: 0.0153 - 82ms/epoch - 6ms/step
Epoch 16/100
14/14 - 0s - loss: 0.0146 - 76ms/epoch - 5ms/step
Epoch 17/100
14/14 - 0s - loss: 0.0136 - 83ms/epoch - 6ms/step
Epoch 18/100
14/14 - 0s - loss: 0.0131 - 70ms/epoch - 5ms/step
Epoch 19/100
14/14 - 0s - loss: 0.0122 - 83ms/epoch - 6ms/step
Epoch 20/100
14/14 - 0s - loss: 0.0123 - 71ms/epoch - 5ms/step
Epoch 21/100
14/14 - 0s - loss: 0.0118 - 87ms/epoch - 6ms/step
Epoch 22/100
14/14 - 0s - loss: 0.0117 - 76ms/epoch - 5ms/step
Epoch 23/100
14/14 - 0s - loss: 0.0118 - 77ms/epoch - 5ms/step
Epoch 24/100
14/14 - 0s - loss: 0.0118 - 74ms/epoch - 5ms/step
Epoch 25/100
14/14 - 0s - loss: 0.0114 - 86ms/epoch - 6ms/step
Epoch 26/100
14/14 - 0s - loss: 0.0115 - 65ms/epoch - 5ms/step
Epoch 27/100
14/14 - 0s - loss: 0.0115 - 78ms/epoch - 6ms/step
Epoch 28/100
14/14 - 0s - loss: 0.0116 - 79ms/epoch - 6ms/step
Epoch 29/100
14/14 - 0s - loss: 0.0119 - 73ms/epoch - 5ms/step
Epoch 30/100
14/14 - 0s - loss: 0.0118 - 81ms/epoch - 6ms/step
Epoch 31/100
14/14 - 0s - loss: 0.0116 - 79ms/epoch - 6ms/step
Epoch 32/100
14/14 - 0s - loss: 0.0121 - 78ms/epoch - 6ms/step
Epoch 33/100
14/14 - 0s - loss: 0.0116 - 81ms/epoch - 6ms/step
Epoch 34/100
14/14 - 0s - loss: 0.0118 - 72ms/epoch - 5ms/step
Epoch 35/100
14/14 - 0s - loss: 0.0115 - 80ms/epoch - 6ms/step
Epoch 36/100
14/14 - 0s - loss: 0.0115 - 71ms/epoch - 5ms/step
Epoch 37/100
14/14 - 0s - loss: 0.0113 - 73ms/epoch - 5ms/step
Epoch 38/100
14/14 - 0s - loss: 0.0119 - 75ms/epoch - 5ms/step
Epoch 39/100
14/14 - 0s - loss: 0.0112 - 71ms/epoch - 5ms/step
Epoch 40/100
14/14 - 0s - loss: 0.0117 - 85ms/epoch - 6ms/step
Epoch 41/100
14/14 - 0s - loss: 0.0113 - 78ms/epoch - 6ms/step
Epoch 42/100
```

```
14/14 - 0s - loss: 0.0115 - 77ms/epoch - 6ms/step
Epoch 43/100
14/14 - 0s - loss: 0.0115 - 87ms/epoch - 6ms/step
Epoch 44/100
14/14 - 0s - loss: 0.0113 - 65ms/epoch - 5ms/step
Epoch 45/100
14/14 - 0s - loss: 0.0118 - 87ms/epoch - 6ms/step
Epoch 46/100
14/14 - 0s - loss: 0.0112 - 69ms/epoch - 5ms/step
Epoch 47/100
14/14 - 0s - loss: 0.0111 - 88ms/epoch - 6ms/step
Epoch 48/100
14/14 - 0s - loss: 0.0113 - 80ms/epoch - 6ms/step
Epoch 49/100
14/14 - 0s - loss: 0.0111 - 68ms/epoch - 5ms/step
Epoch 50/100
14/14 - 0s - loss: 0.0111 - 83ms/epoch - 6ms/step
Epoch 51/100
14/14 - 0s - loss: 0.0112 - 69ms/epoch - 5ms/step
Epoch 52/100
14/14 - 0s - loss: 0.0113 - 79ms/epoch - 6ms/step
Epoch 53/100
14/14 - 0s - loss: 0.0113 - 80ms/epoch - 6ms/step
Epoch 54/100
14/14 - 0s - loss: 0.0113 - 87ms/epoch - 6ms/step
Epoch 55/100
14/14 - 0s - loss: 0.0113 - 87ms/epoch - 6ms/step
Epoch 56/100
14/14 - 0s - loss: 0.0116 - 119ms/epoch - 8ms/step
Epoch 57/100
14/14 - 0s - loss: 0.0111 - 87ms/epoch - 6ms/step
Epoch 58/100
14/14 - 0s - loss: 0.0113 - 77ms/epoch - 6ms/step
Epoch 59/100
14/14 - 0s - loss: 0.0110 - 69ms/epoch - 5ms/step
Epoch 60/100
14/14 - 0s - loss: 0.0111 - 79ms/epoch - 6ms/step
Epoch 61/100
14/14 - 0s - loss: 0.0111 - 79ms/epoch - 6ms/step
Epoch 62/100
14/14 - 0s - loss: 0.0115 - 74ms/epoch - 5ms/step
Epoch 63/100
14/14 - 0s - loss: 0.0112 - 69ms/epoch - 5ms/step
Epoch 64/100
14/14 - 0s - loss: 0.0111 - 81ms/epoch - 6ms/step
Epoch 65/100
14/14 - 0s - loss: 0.0114 - 76ms/epoch - 5ms/step
Epoch 66/100
14/14 - 0s - loss: 0.0110 - 73ms/epoch - 5ms/step
Epoch 67/100
14/14 - 0s - loss: 0.0114 - 75ms/epoch - 5ms/step
Epoch 68/100
14/14 - 0s - loss: 0.0112 - 69ms/epoch - 5ms/step
Epoch 69/100
14/14 - 0s - loss: 0.0111 - 87ms/epoch - 6ms/step
Epoch 70/100
14/14 - 0s - loss: 0.0113 - 80ms/epoch - 6ms/step
Epoch 71/100
14/14 - 0s - loss: 0.0110 - 78ms/epoch - 6ms/step
Epoch 72/100
14/14 - 0s - loss: 0.0123 - 80ms/epoch - 6ms/step
Epoch 73/100
14/14 - 0s - loss: 0.0121 - 74ms/epoch - 5ms/step
Epoch 74/100
14/14 - 0s - loss: 0.0123 - 74ms/epoch - 5ms/step
Epoch 75/100
```

```
14/14 - 0s - loss: 0.0116 - 77ms/epoch - 5ms/step
Epoch 76/100
14/14 - 0s - loss: 0.0115 - 72ms/epoch - 5ms/step
Epoch 77/100
14/14 - 0s - loss: 0.0110 - 83ms/epoch - 6ms/step
Epoch 78/100
14/14 - 0s - loss: 0.0112 - 66ms/epoch - 5ms/step
Epoch 79/100
14/14 - 0s - loss: 0.0113 - 80ms/epoch - 6ms/step
Epoch 80/100
14/14 - 0s - loss: 0.0111 - 77ms/epoch - 5ms/step
Epoch 81/100
14/14 - 0s - loss: 0.0115 - 67ms/epoch - 5ms/step
Epoch 82/100
14/14 - 0s - loss: 0.0114 - 72ms/epoch - 5ms/step
Epoch 83/100
14/14 - 0s - loss: 0.0107 - 75ms/epoch - 5ms/step
Epoch 84/100
14/14 - 0s - loss: 0.0109 - 73ms/epoch - 5ms/step
Epoch 85/100
14/14 - 0s - loss: 0.0117 - 68ms/epoch - 5ms/step
Epoch 86/100
14/14 - 0s - loss: 0.0115 - 80ms/epoch - 6ms/step
Epoch 87/100
14/14 - 0s - loss: 0.0109 - 129ms/epoch - 9ms/step
Epoch 88/100
14/14 - 0s - loss: 0.0112 - 71ms/epoch - 5ms/step
Epoch 89/100
14/14 - 0s - loss: 0.0119 - 75ms/epoch - 5ms/step
Epoch 90/100
14/14 - 0s - loss: 0.0119 - 82ms/epoch - 6ms/step
Epoch 91/100
14/14 - 0s - loss: 0.0112 - 80ms/epoch - 6ms/step
Epoch 92/100
14/14 - 0s - loss: 0.0116 - 77ms/epoch - 6ms/step
Epoch 93/100
14/14 - 0s - loss: 0.0109 - 73ms/epoch - 5ms/step
Epoch 94/100
14/14 - 0s - loss: 0.0110 - 76ms/epoch - 5ms/step
Epoch 95/100
14/14 - 0s - loss: 0.0108 - 80ms/epoch - 6ms/step
Epoch 96/100
14/14 - 0s - loss: 0.0115 - 75ms/epoch - 5ms/step
Epoch 97/100
14/14 - 0s - loss: 0.0111 - 74ms/epoch - 5ms/step
Epoch 98/100
14/14 - 0s - loss: 0.0108 - 72ms/epoch - 5ms/step
Epoch 99/100
14/14 - 0s - loss: 0.0114 - 79ms/epoch - 6ms/step
Epoch 100/100
14/14 - 0s - loss: 0.0109 - 80ms/epoch - 6ms/step
```

Out[10]:

```
<keras.callbacks.History at 0x20f5adeeaf0>
```

In [11]:

```python
# predictions array with shape (2,1)
predictions = np.array([[1], [2]])

expanded_predictions = np.repeat(predictions, 3, axis=1)

# Now, expanded_predictions has shape (2, 3), which matches the expected input shape for
try:
    inverse_transformed_predictions = scaler.inverse_transform(expanded_predictions)
    print(inverse_transformed_predictions)
except Exception as e:
    print(f"Error during inverse transformation: {e}")
```

```
[[2.023000e+03 4.000000e+00 4.159069e+05]
 [2.027000e+03 7.000000e+00 5.917351e+05]]
```

Vanilla LSTM

In [13]:
```
data.head()
```

Out[13]:

	Anul	Trimestrul	PIB
0	2019	1	252368.6
1	2019	2	262370.8
2	2019	3	267966.3
3	2019	4	276961.0
4	2020	1	272031.8

In [14]:
```
scaler = MinMaxScaler(feature_range=(0, 1))
data['PIB'] = scaler.fit_transform(data[['PIB']])
```

In [15]:
```
def create_sequences(data, sequence_length):
    X, y = [], []
    for i in range(len(data) - sequence_length):
        X.append(data[i:i+sequence_length])
        y.append(data[i+sequence_length])
    return np.array(X), np.array(y)

sequence_length = 3
X, y = create_sequences(data['PIB'].values, sequence_length)
```

In [16]:
```
print("Shape of X before reshaping:", X.shape)


if len(X.shape) == 2:
    X = X.reshape((X.shape[0], X.shape[1], 1))

print("Shape of X after reshaping:", X.shape)


model = Sequential()
model.add(LSTM(units=50, input_shape=(X.shape[1], X.shape[2])))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
Shape of X before reshaping: (17, 3)
Shape of X after reshaping: (17, 3, 1)
```

In [17]:
```
model = Sequential()
model.add(LSTM(units=50, input_shape=(X.shape[1], X.shape[2])))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')
```

In [18]:
```
history = model.fit(X, y, epochs=100, batch_size=1, verbose=2)
```

```
Epoch 1/100
17/17 - 4s - loss: 0.1996 - 4s/epoch - 211ms/step
Epoch 2/100
17/17 - 0s - loss: 0.0827 - 83ms/epoch - 5ms/step
Epoch 3/100
17/17 - 0s - loss: 0.0272 - 80ms/epoch - 5ms/step
Epoch 4/100
17/17 - 0s - loss: 0.0224 - 82ms/epoch - 5ms/step
```

```
Epoch 5/100
17/17 - 0s - loss: 0.0179 - 86ms/epoch - 5ms/step
Epoch 6/100
17/17 - 0s - loss: 0.0165 - 77ms/epoch - 5ms/step
Epoch 7/100
17/17 - 0s - loss: 0.0131 - 87ms/epoch - 5ms/step
Epoch 8/100
17/17 - 0s - loss: 0.0113 - 81ms/epoch - 5ms/step
Epoch 9/100
17/17 - 0s - loss: 0.0097 - 88ms/epoch - 5ms/step
Epoch 10/100
17/17 - 0s - loss: 0.0094 - 79ms/epoch - 5ms/step
Epoch 11/100
17/17 - 0s - loss: 0.0082 - 88ms/epoch - 5ms/step
Epoch 12/100
17/17 - 0s - loss: 0.0080 - 81ms/epoch - 5ms/step
Epoch 13/100
17/17 - 0s - loss: 0.0075 - 81ms/epoch - 5ms/step
Epoch 14/100
17/17 - 0s - loss: 0.0073 - 82ms/epoch - 5ms/step
Epoch 15/100
17/17 - 0s - loss: 0.0075 - 80ms/epoch - 5ms/step
Epoch 16/100
17/17 - 0s - loss: 0.0074 - 80ms/epoch - 5ms/step
Epoch 17/100
17/17 - 0s - loss: 0.0071 - 83ms/epoch - 5ms/step
Epoch 18/100
17/17 - 0s - loss: 0.0073 - 80ms/epoch - 5ms/step
Epoch 19/100
17/17 - 0s - loss: 0.0072 - 78ms/epoch - 5ms/step
Epoch 20/100
17/17 - 0s - loss: 0.0072 - 98ms/epoch - 6ms/step
Epoch 21/100
17/17 - 0s - loss: 0.0072 - 80ms/epoch - 5ms/step
Epoch 22/100
17/17 - 0s - loss: 0.0076 - 85ms/epoch - 5ms/step
Epoch 23/100
17/17 - 0s - loss: 0.0078 - 81ms/epoch - 5ms/step
Epoch 24/100
17/17 - 0s - loss: 0.0078 - 83ms/epoch - 5ms/step
Epoch 25/100
17/17 - 0s - loss: 0.0073 - 79ms/epoch - 5ms/step
Epoch 26/100
17/17 - 0s - loss: 0.0070 - 88ms/epoch - 5ms/step
Epoch 27/100
17/17 - 0s - loss: 0.0070 - 70ms/epoch - 4ms/step
Epoch 28/100
17/17 - 0s - loss: 0.0071 - 89ms/epoch - 5ms/step
Epoch 29/100
17/17 - 0s - loss: 0.0072 - 70ms/epoch - 4ms/step
Epoch 30/100
17/17 - 0s - loss: 0.0069 - 84ms/epoch - 5ms/step
Epoch 31/100
17/17 - 0s - loss: 0.0075 - 79ms/epoch - 5ms/step
Epoch 32/100
17/17 - 0s - loss: 0.0068 - 82ms/epoch - 5ms/step
Epoch 33/100
17/17 - 0s - loss: 0.0074 - 87ms/epoch - 5ms/step
Epoch 34/100
17/17 - 0s - loss: 0.0072 - 73ms/epoch - 4ms/step
Epoch 35/100
17/17 - 0s - loss: 0.0070 - 83ms/epoch - 5ms/step
Epoch 36/100
17/17 - 0s - loss: 0.0069 - 70ms/epoch - 4ms/step
Epoch 37/100
17/17 - 0s - loss: 0.0070 - 74ms/epoch - 4ms/step
```

```
Epoch 38/100
17/17 - 0s - loss: 0.0073 - 76ms/epoch - 4ms/step
Epoch 39/100
17/17 - 0s - loss: 0.0068 - 84ms/epoch - 5ms/step
Epoch 40/100
17/17 - 0s - loss: 0.0076 - 80ms/epoch - 5ms/step
Epoch 41/100
17/17 - 0s - loss: 0.0068 - 87ms/epoch - 5ms/step
Epoch 42/100
17/17 - 0s - loss: 0.0073 - 76ms/epoch - 4ms/step
Epoch 43/100
17/17 - 0s - loss: 0.0077 - 85ms/epoch - 5ms/step
Epoch 44/100
17/17 - 0s - loss: 0.0065 - 86ms/epoch - 5ms/step
Epoch 45/100
17/17 - 0s - loss: 0.0070 - 84ms/epoch - 5ms/step
Epoch 46/100
17/17 - 0s - loss: 0.0071 - 106ms/epoch - 6ms/step
Epoch 47/100
17/17 - 0s - loss: 0.0071 - 88ms/epoch - 5ms/step
Epoch 48/100
17/17 - 0s - loss: 0.0072 - 80ms/epoch - 5ms/step
Epoch 49/100
17/17 - 0s - loss: 0.0067 - 95ms/epoch - 6ms/step
Epoch 50/100
17/17 - 0s - loss: 0.0069 - 85ms/epoch - 5ms/step
Epoch 51/100
17/17 - 0s - loss: 0.0069 - 85ms/epoch - 5ms/step
Epoch 52/100
17/17 - 0s - loss: 0.0070 - 101ms/epoch - 6ms/step
Epoch 53/100
17/17 - 0s - loss: 0.0075 - 131ms/epoch - 8ms/step
Epoch 54/100
17/17 - 0s - loss: 0.0074 - 86ms/epoch - 5ms/step
Epoch 55/100
17/17 - 0s - loss: 0.0075 - 103ms/epoch - 6ms/step
Epoch 56/100
17/17 - 0s - loss: 0.0068 - 123ms/epoch - 7ms/step
Epoch 57/100
17/17 - 0s - loss: 0.0066 - 102ms/epoch - 6ms/step
Epoch 58/100
17/17 - 0s - loss: 0.0070 - 107ms/epoch - 6ms/step
Epoch 59/100
17/17 - 0s - loss: 0.0066 - 77ms/epoch - 5ms/step
Epoch 60/100
17/17 - 0s - loss: 0.0069 - 76ms/epoch - 4ms/step
Epoch 61/100
17/17 - 0s - loss: 0.0065 - 90ms/epoch - 5ms/step
Epoch 62/100
17/17 - 0s - loss: 0.0071 - 95ms/epoch - 6ms/step
Epoch 63/100
17/17 - 0s - loss: 0.0072 - 88ms/epoch - 5ms/step
Epoch 64/100
17/17 - 0s - loss: 0.0068 - 87ms/epoch - 5ms/step
Epoch 65/100
17/17 - 0s - loss: 0.0068 - 78ms/epoch - 5ms/step
Epoch 66/100
17/17 - 0s - loss: 0.0066 - 82ms/epoch - 5ms/step
Epoch 67/100
17/17 - 0s - loss: 0.0067 - 134ms/epoch - 8ms/step
Epoch 68/100
17/17 - 0s - loss: 0.0070 - 107ms/epoch - 6ms/step
Epoch 69/100
17/17 - 0s - loss: 0.0069 - 92ms/epoch - 5ms/step
Epoch 70/100
17/17 - 0s - loss: 0.0080 - 78ms/epoch - 5ms/step
```

```
Epoch 71/100
17/17 - 0s - loss: 0.0067 - 92ms/epoch - 5ms/step
Epoch 72/100
17/17 - 0s - loss: 0.0068 - 142ms/epoch - 8ms/step
Epoch 73/100
17/17 - 0s - loss: 0.0062 - 117ms/epoch - 7ms/step
Epoch 74/100
17/17 - 0s - loss: 0.0067 - 121ms/epoch - 7ms/step
Epoch 75/100
17/17 - 0s - loss: 0.0068 - 90ms/epoch - 5ms/step
Epoch 76/100
17/17 - 0s - loss: 0.0064 - 87ms/epoch - 5ms/step
Epoch 77/100
17/17 - 0s - loss: 0.0063 - 74ms/epoch - 4ms/step
Epoch 78/100
17/17 - 0s - loss: 0.0064 - 96ms/epoch - 6ms/step
Epoch 79/100
17/17 - 0s - loss: 0.0065 - 88ms/epoch - 5ms/step
Epoch 80/100
17/17 - 0s - loss: 0.0069 - 91ms/epoch - 5ms/step
Epoch 81/100
17/17 - 0s - loss: 0.0068 - 107ms/epoch - 6ms/step
Epoch 82/100
17/17 - 0s - loss: 0.0062 - 82ms/epoch - 5ms/step
Epoch 83/100
17/17 - 0s - loss: 0.0064 - 86ms/epoch - 5ms/step
Epoch 84/100
17/17 - 0s - loss: 0.0064 - 124ms/epoch - 7ms/step
Epoch 85/100
17/17 - 0s - loss: 0.0065 - 66ms/epoch - 4ms/step
Epoch 86/100
17/17 - 0s - loss: 0.0062 - 81ms/epoch - 5ms/step
Epoch 87/100
17/17 - 0s - loss: 0.0066 - 88ms/epoch - 5ms/step
Epoch 88/100
17/17 - 0s - loss: 0.0065 - 77ms/epoch - 5ms/step
Epoch 89/100
17/17 - 0s - loss: 0.0066 - 86ms/epoch - 5ms/step
Epoch 90/100
17/17 - 0s - loss: 0.0065 - 83ms/epoch - 5ms/step
Epoch 91/100
17/17 - 0s - loss: 0.0066 - 64ms/epoch - 4ms/step
Epoch 92/100
17/17 - 0s - loss: 0.0064 - 72ms/epoch - 4ms/step
Epoch 93/100
17/17 - 0s - loss: 0.0065 - 73ms/epoch - 4ms/step
Epoch 94/100
17/17 - 0s - loss: 0.0065 - 72ms/epoch - 4ms/step
Epoch 95/100
17/17 - 0s - loss: 0.0074 - 79ms/epoch - 5ms/step
Epoch 96/100
17/17 - 0s - loss: 0.0068 - 83ms/epoch - 5ms/step
Epoch 97/100
17/17 - 0s - loss: 0.0068 - 78ms/epoch - 5ms/step
Epoch 98/100
17/17 - 0s - loss: 0.0063 - 80ms/epoch - 5ms/step
Epoch 99/100
17/17 - 0s - loss: 0.0061 - 79ms/epoch - 5ms/step
Epoch 100/100
17/17 - 0s - loss: 0.0061 - 81ms/epoch - 5ms/step
```

In [19]:

```
# Make predictions
predictions = model.predict(X)
predictions_inverse = scaler.inverse_transform(predictions)
y_inverse = scaler.inverse_transform(y.reshape(-1, 1))
```

```
1/1 [==============================] - 1s 971ms/step
```

In [20]:
```
from sklearn.metrics import mean_squared_error
rmse = np.sqrt(mean_squared_error(y_inverse, predictions_inverse))
print("Root Mean Squared Error:", rmse)
```

```
Root Mean Squared Error: 13474.31446350236
```

Bidirectional LSTM

In [21]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Bidirectional
import matplotlib.pyplot as plt
from keras.models import load_model
from numpy import array
```

In [22]:
```
def split_sequence(sequence, n_steps):
    X, y = list(), list()
    for i in range(len(sequence)):
        #find the end of this pattern
        end_ix = i + n_steps
        #check if we are beyond the sequence
        if end_ix > len(sequence)-1:
            break
            #gather input and output parts of the pattern
            seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
            X.append(seq_x)
            y.append(seq_y)
            return array(X), array(y)
```

In [23]:
```
#define input sequence
raw_sequence = data['PIB'].tolist()
```

In [24]:
```
n_steps = 3
```

In [25]:
```
n_features = 1
```

In [26]:
```
# Adjust the input_shape parameter to match data's shape
model = Sequential([
    LSTM(units=50, return_sequences=True, input_shape=(3, 1)),  # Adjusted to (3, 1)
    LSTM(units=50),
    Dense(1)
])

# Compile model as before
model.compile(optimizer='adam', loss='mse')
```

In [27]:
```
model.fit(X, y, epochs=100, verbose=1)
```

```
Epoch 1/100
1/1 [==============================] - 8s 8s/step - loss: 0.3144
Epoch 2/100
1/1 [==============================] - 0s 23ms/step - loss: 0.3008
Epoch 3/100
1/1 [==============================] - 0s 25ms/step - loss: 0.2874
Epoch 4/100
1/1 [==============================] - 0s 21ms/step - loss: 0.2743
Epoch 5/100
1/1 [==============================] - 0s 22ms/step - loss: 0.2613
Epoch 6/100
1/1 [==============================] - 0s 25ms/step - loss: 0.2483
Epoch 7/100
1/1 [==============================] - 0s 24ms/step - loss: 0.2353
```

```
Epoch 8/100
1/1 [==============================] - 0s 18ms/step - loss: 0.2223
Epoch 9/100
1/1 [==============================] - 0s 23ms/step - loss: 0.2091
Epoch 10/100
1/1 [==============================] - 0s 19ms/step - loss: 0.1959
Epoch 11/100
1/1 [==============================] - 0s 21ms/step - loss: 0.1826
Epoch 12/100
1/1 [==============================] - 0s 22ms/step - loss: 0.1691
Epoch 13/100
1/1 [==============================] - 0s 21ms/step - loss: 0.1557
Epoch 14/100
1/1 [==============================] - 0s 27ms/step - loss: 0.1422
Epoch 15/100
1/1 [==============================] - 0s 16ms/step - loss: 0.1287
Epoch 16/100
1/1 [==============================] - 0s 19ms/step - loss: 0.1154
Epoch 17/100
1/1 [==============================] - 0s 21ms/step - loss: 0.1022
Epoch 18/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0894
Epoch 19/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0771
Epoch 20/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0656
Epoch 21/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0549
Epoch 22/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0455
Epoch 23/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0375
Epoch 24/100
1/1 [==============================] - 0s 13ms/step - loss: 0.0314
Epoch 25/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0272
Epoch 26/100
1/1 [==============================] - 0s 22ms/step - loss: 0.0253
Epoch 27/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0254
Epoch 28/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0274
Epoch 29/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0305
Epoch 30/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0338
Epoch 31/100
1/1 [==============================] - 0s 29ms/step - loss: 0.0363
Epoch 32/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0375
Epoch 33/100
1/1 [==============================] - 0s 15ms/step - loss: 0.0372
Epoch 34/100
1/1 [==============================] - 0s 27ms/step - loss: 0.0355
Epoch 35/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0329
Epoch 36/100
1/1 [==============================] - 0s 31ms/step - loss: 0.0299
Epoch 37/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0269
Epoch 38/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0243
Epoch 39/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0222
Epoch 40/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0208
```

```
Epoch 41/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0199
Epoch 42/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0194
Epoch 43/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0193
Epoch 44/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0194
Epoch 45/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0195
Epoch 46/100
1/1 [==============================] - 0s 62ms/step - loss: 0.0196
Epoch 47/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0196
Epoch 48/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0194
Epoch 49/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0191
Epoch 50/100
1/1 [==============================] - 0s 31ms/step - loss: 0.0185
Epoch 51/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0179
Epoch 52/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0171
Epoch 53/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0163
Epoch 54/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0155
Epoch 55/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0147
Epoch 56/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0140
Epoch 57/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0134
Epoch 58/100
1/1 [==============================] - 0s 27ms/step - loss: 0.0130
Epoch 59/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0126
Epoch 60/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0124
Epoch 61/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0122
Epoch 62/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0121
Epoch 63/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0119
Epoch 64/100
1/1 [==============================] - 0s 29ms/step - loss: 0.0117
Epoch 65/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0115
Epoch 66/100
1/1 [==============================] - 0s 32ms/step - loss: 0.0112
Epoch 67/100
1/1 [==============================] - 0s 28ms/step - loss: 0.0108
Epoch 68/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0105
Epoch 69/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0101
Epoch 70/100
1/1 [==============================] - 0s 36ms/step - loss: 0.0098
Epoch 71/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0096
Epoch 72/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0094
Epoch 73/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0092
```

```
Epoch 74/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0091
Epoch 75/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0090
Epoch 76/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0089
Epoch 77/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0088
Epoch 78/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0087
Epoch 79/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0086
Epoch 80/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0085
Epoch 81/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0084
Epoch 82/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0083
Epoch 83/100
1/1 [==============================] - 0s 22ms/step - loss: 0.0082
Epoch 84/100
1/1 [==============================] - 0s 52ms/step - loss: 0.0082
Epoch 85/100
1/1 [==============================] - 0s 29ms/step - loss: 0.0081
Epoch 86/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0081
Epoch 87/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0081
Epoch 88/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0081
Epoch 89/100
1/1 [==============================] - 0s 29ms/step - loss: 0.0080
Epoch 90/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0080
Epoch 91/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0080
Epoch 92/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0080
Epoch 93/100
1/1 [==============================] - 0s 15ms/step - loss: 0.0080
Epoch 94/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0080
Epoch 95/100
1/1 [==============================] - 0s 29ms/step - loss: 0.0079
Epoch 96/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0079
Epoch 97/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0079
Epoch 98/100
1/1 [==============================] - 0s 13ms/step - loss: 0.0079
Epoch 99/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0079
Epoch 100/100
1/1 [==============================] - 0s 28ms/step - loss: 0.0079
```

Out[27]:

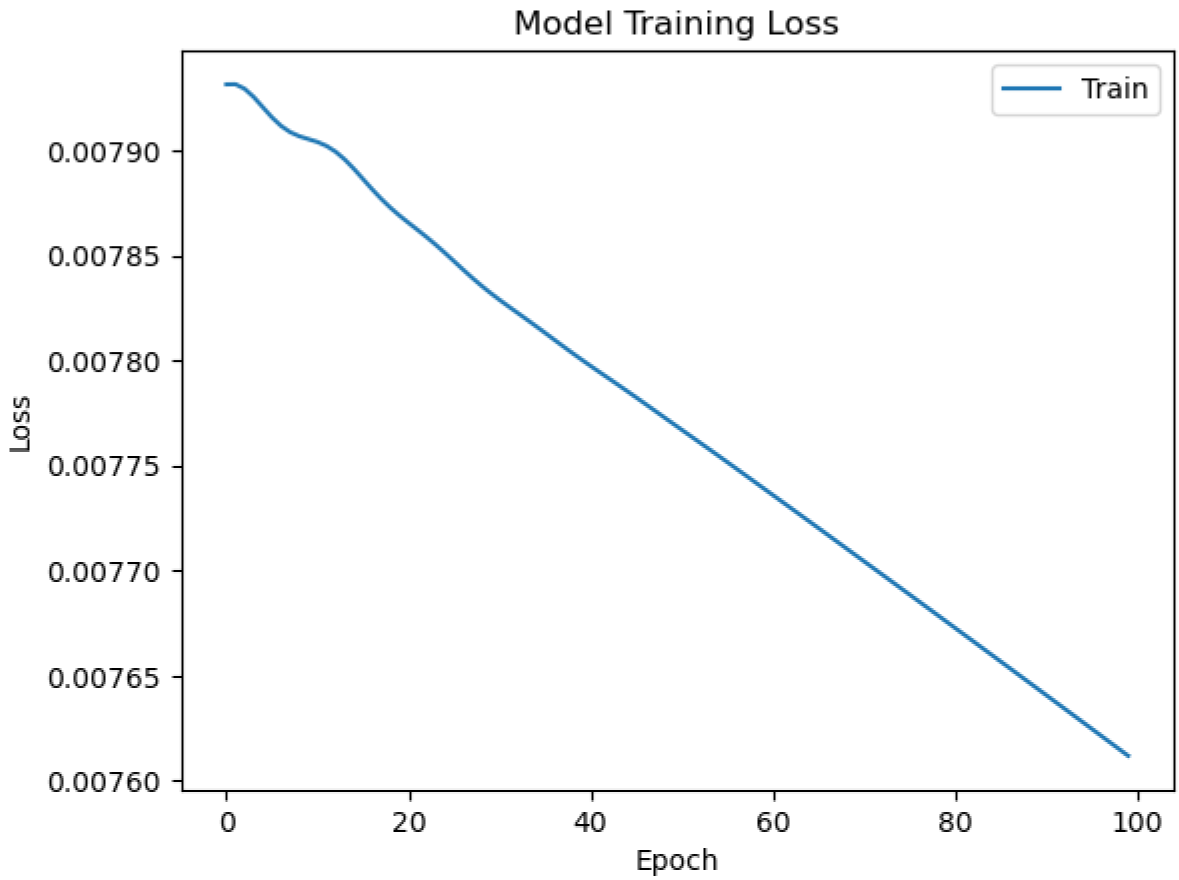
```
<keras.callbacks.History at 0x20f62eaf880>
```

In [28]:

```python
# Assuming 'history' is the History object returned by model.fit()
history = model.fit(X, y, epochs=100, verbose=0)

# Plot training loss
plt.plot(history.history['loss'])
plt.title('Model Training Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
```

```
plt.legend(['Train'], loc='upper right')
plt.show()
```



In [29]:
```
last_pib_values = array([391468.3, 402454.9, 415906.9])
```

In [30]:
```
n_steps = 3  # Number of time steps your model was trained on
n_features = 1  # Number of features per time step (univariate means 1)
x_input = last_pib_values.reshape((1, n_steps, n_features))
```

In [31]:
```
yhat = model.predict(x_input, verbose=0)
```

In [32]:
```
print(yhat)
```

```
[[4.800913]]
```

CNN LSTM

In [33]:
```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import LSTM, Dense, Flatten, TimeDistributed, Conv1D, MaxPooling1D
from sklearn.preprocessing import MinMaxScaler
from numpy import array
```

In [34]:
```
sequence = data['PIB'].values
```

In [35]:
```
scaler = MinMaxScaler(feature_range=(0, 1))
sequence = scaler.fit_transform(sequence.reshape(-1, 1)).flatten()
```

In [36]:
```
def split_sequence(sequence, n_steps):
    X, y = [], []
    for i in range(len(sequence)):
        end_ix = i + n_steps
        if end_ix > len(sequence)-1:
```

```
            break
        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)
```

In [37]:
```
n_steps = 4  # Timesteps per sample
X, y = split_sequence(sequence, n_steps)
```

In [38]:
```
n_features = 1  # Features per step (univariate)
n_seq = 2  # Number of subsequences
n_steps = 2  # Steps per subsequence
X = X.reshape((X.shape[0], n_seq, n_steps, n_features))
```

In [39]:
```
model = Sequential([
    TimeDistributed(Conv1D(filters=64, kernel_size=1, activation='relu'), input_shape=(N
    TimeDistributed(MaxPooling1D(pool_size=2)),
    TimeDistributed(Flatten()),
    LSTM(50, activation='relu'),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')
```

In [40]:
```
model.fit(X, y, epochs=100, verbose=1)
```

```
Epoch 1/100
1/1 [==============================] - 4s 4s/step - loss: 0.3254
Epoch 2/100
1/1 [==============================] - 0s 12ms/step - loss: 0.3150
Epoch 3/100
1/1 [==============================] - 0s 28ms/step - loss: 0.3053
Epoch 4/100
1/1 [==============================] - 0s 19ms/step - loss: 0.2962
Epoch 5/100
1/1 [==============================] - 0s 11ms/step - loss: 0.2875
Epoch 6/100
1/1 [==============================] - 0s 20ms/step - loss: 0.2790
Epoch 7/100
1/1 [==============================] - 0s 19ms/step - loss: 0.2708
Epoch 8/100
1/1 [==============================] - 0s 16ms/step - loss: 0.2629
Epoch 9/100
1/1 [==============================] - 0s 18ms/step - loss: 0.2554
Epoch 10/100
1/1 [==============================] - 0s 16ms/step - loss: 0.2480
Epoch 11/100
1/1 [==============================] - 0s 16ms/step - loss: 0.2407
Epoch 12/100
1/1 [==============================] - 0s 18ms/step - loss: 0.2332
Epoch 13/100
1/1 [==============================] - 0s 21ms/step - loss: 0.2257
Epoch 14/100
1/1 [==============================] - 0s 20ms/step - loss: 0.2180
Epoch 15/100
1/1 [==============================] - 0s 18ms/step - loss: 0.2103
Epoch 16/100
1/1 [==============================] - 0s 16ms/step - loss: 0.2026
Epoch 17/100
1/1 [==============================] - 0s 20ms/step - loss: 0.1948
Epoch 18/100
1/1 [==============================] - 0s 20ms/step - loss: 0.1871
Epoch 19/100
1/1 [==============================] - 0s 26ms/step - loss: 0.1793
Epoch 20/100
```

```
1/1 [==============================] - 0s 21ms/step - loss: 0.1715
Epoch 21/100
1/1 [==============================] - 0s 20ms/step - loss: 0.1637
Epoch 22/100
1/1 [==============================] - 0s 16ms/step - loss: 0.1557
Epoch 23/100
1/1 [==============================] - 0s 19ms/step - loss: 0.1476
Epoch 24/100
1/1 [==============================] - 0s 15ms/step - loss: 0.1395
Epoch 25/100
1/1 [==============================] - 0s 12ms/step - loss: 0.1314
Epoch 26/100
1/1 [==============================] - 0s 15ms/step - loss: 0.1232
Epoch 27/100
1/1 [==============================] - 0s 14ms/step - loss: 0.1150
Epoch 28/100
1/1 [==============================] - 0s 19ms/step - loss: 0.1068
Epoch 29/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0987
Epoch 30/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0907
Epoch 31/100
1/1 [==============================] - 0s 27ms/step - loss: 0.0828
Epoch 32/100
1/1 [==============================] - 0s 22ms/step - loss: 0.0751
Epoch 33/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0676
Epoch 34/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0604
Epoch 35/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0536
Epoch 36/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0472
Epoch 37/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0412
Epoch 38/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0358
Epoch 39/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0310
Epoch 40/100
1/1 [==============================] - 0s 22ms/step - loss: 0.0269
Epoch 41/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0235
Epoch 42/100
1/1 [==============================] - 0s 9ms/step - loss: 0.0208
Epoch 43/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0189
Epoch 44/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0177
Epoch 45/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0172
Epoch 46/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0173
Epoch 47/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0178
Epoch 48/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0187
Epoch 49/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0196
Epoch 50/100
1/1 [==============================] - 0s 27ms/step - loss: 0.0206
Epoch 51/100
1/1 [==============================] - 0s 40ms/step - loss: 0.0214
Epoch 52/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0219
Epoch 53/100
```

```
1/1 [==============================] - 0s 16ms/step - loss: 0.0221
Epoch 54/100
1/1 [==============================] - 0s 35ms/step - loss: 0.0219
Epoch 55/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0214
Epoch 56/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0207
Epoch 57/100
1/1 [==============================] - 0s 33ms/step - loss: 0.0198
Epoch 58/100
1/1 [==============================] - 0s 47ms/step - loss: 0.0189
Epoch 59/100
1/1 [==============================] - 0s 33ms/step - loss: 0.0180
Epoch 60/100
1/1 [==============================] - 0s 27ms/step - loss: 0.0171
Epoch 61/100
1/1 [==============================] - 0s 63ms/step - loss: 0.0164
Epoch 62/100
1/1 [==============================] - 0s 53ms/step - loss: 0.0158
Epoch 63/100
1/1 [==============================] - 0s 36ms/step - loss: 0.0153
Epoch 64/100
1/1 [==============================] - 0s 32ms/step - loss: 0.0150
Epoch 65/100
1/1 [==============================] - 0s 32ms/step - loss: 0.0148
Epoch 66/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0147
Epoch 67/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0147
Epoch 68/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0147
Epoch 69/100
1/1 [==============================] - 0s 40ms/step - loss: 0.0147
Epoch 70/100
1/1 [==============================] - 0s 39ms/step - loss: 0.0147
Epoch 71/100
1/1 [==============================] - 0s 37ms/step - loss: 0.0147
Epoch 72/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0146
Epoch 73/100
1/1 [==============================] - 0s 32ms/step - loss: 0.0145
Epoch 74/100
1/1 [==============================] - 0s 22ms/step - loss: 0.0144
Epoch 75/100
1/1 [==============================] - 0s 33ms/step - loss: 0.0142
Epoch 76/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0140
Epoch 77/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0137
Epoch 78/100
1/1 [==============================] - 0s 29ms/step - loss: 0.0135
Epoch 79/100
1/1 [==============================] - 0s 31ms/step - loss: 0.0133
Epoch 80/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0131
Epoch 81/100
1/1 [==============================] - 0s 47ms/step - loss: 0.0129
Epoch 82/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0127
Epoch 83/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0125
Epoch 84/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0124
Epoch 85/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0123
Epoch 86/100
```

```
1/1 [==============================] - 0s 27ms/step - loss: 0.0121
Epoch 87/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0120
Epoch 88/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0120
Epoch 89/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0119
Epoch 90/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0118
Epoch 91/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0117
Epoch 92/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0116
Epoch 93/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0115
Epoch 94/100
1/1 [==============================] - 0s 52ms/step - loss: 0.0113
Epoch 95/100
1/1 [==============================] - 0s 15ms/step - loss: 0.0112
Epoch 96/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0111
Epoch 97/100
1/1 [==============================] - 0s 11ms/step - loss: 0.0110
Epoch 98/100
1/1 [==============================] - 0s 36ms/step - loss: 0.0109
Epoch 99/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0108
Epoch 100/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0106
```

Out[40]:
```
<keras.callbacks.History at 0x20f698bd1c0>
```

In [41]:
```
x_input = sequence[-4:]  # Last four entries from the sequence
x_input = scaler.transform(x_input.reshape(-1, 1)).flatten()  # Normalize input
x_input = x_input.reshape((1, n_seq, n_steps, n_features))
yhat = model.predict(x_input, verbose=0)
print(yhat)
```

```
[[1.0792195]]
```

ConvLSTM

In [42]:
```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Flatten, ConvLSTM2D
from sklearn.preprocessing import MinMaxScaler
from numpy import array
```

In [43]:
```
sequence = data['PIB'].values
```

In [44]:
```
scaler = MinMaxScaler(feature_range=(0, 1))
sequence = scaler.fit_transform(sequence.reshape(-1, 1)).flatten()
```

In [45]:
```
def split_sequence(sequence, n_steps):
    X, y = [], []
    for i in range(len(sequence)):
        end_ix = i + n_steps
        if end_ix > len(sequence)-1:
            break
        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)
```

In [46]:
```
n_steps = 4  # Timesteps per sample
X, y = split_sequence(sequence, n_steps)
```

In [47]:
```
n_features = 1  # Features per step (univariate)
n_seq = 2  # Number of subsequences
n_steps = 2  # Steps per subsequence
X = X.reshape((X.shape[0], n_seq, 1, n_steps, n_features))
```

In [48]:
```
model = Sequential([
    ConvLSTM2D(filters=64, kernel_size=(1,2), activation='relu', input_shape=(n_seq, 1,
    Flatten(),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')
```

In [49]:
```
model.fit(X, y, epochs=100, verbose=1)
```

```
Epoch 1/100
1/1 [==============================] - 6s 6s/step - loss: 0.3295
Epoch 2/100
1/1 [==============================] - 0s 26ms/step - loss: 0.3236
Epoch 3/100
1/1 [==============================] - 0s 22ms/step - loss: 0.3178
Epoch 4/100
1/1 [==============================] - 0s 32ms/step - loss: 0.3122
Epoch 5/100
1/1 [==============================] - 0s 24ms/step - loss: 0.3067
Epoch 6/100
1/1 [==============================] - 0s 19ms/step - loss: 0.3013
Epoch 7/100
1/1 [==============================] - 0s 22ms/step - loss: 0.2960
Epoch 8/100
1/1 [==============================] - 0s 21ms/step - loss: 0.2910
Epoch 9/100
1/1 [==============================] - 0s 26ms/step - loss: 0.2860
Epoch 10/100
1/1 [==============================] - 0s 28ms/step - loss: 0.2812
Epoch 11/100
1/1 [==============================] - 0s 52ms/step - loss: 0.2764
Epoch 12/100
1/1 [==============================] - 0s 25ms/step - loss: 0.2718
Epoch 13/100
1/1 [==============================] - 0s 17ms/step - loss: 0.2672
Epoch 14/100
1/1 [==============================] - 0s 18ms/step - loss: 0.2627
Epoch 15/100
1/1 [==============================] - 0s 22ms/step - loss: 0.2582
Epoch 16/100
1/1 [==============================] - 0s 14ms/step - loss: 0.2536
Epoch 17/100
1/1 [==============================] - 0s 15ms/step - loss: 0.2490
Epoch 18/100
1/1 [==============================] - 0s 23ms/step - loss: 0.2444
Epoch 19/100
1/1 [==============================] - 0s 17ms/step - loss: 0.2397
Epoch 20/100
1/1 [==============================] - 0s 20ms/step - loss: 0.2350
Epoch 21/100
1/1 [==============================] - 0s 18ms/step - loss: 0.2301
Epoch 22/100
1/1 [==============================] - 0s 19ms/step - loss: 0.2251
Epoch 23/100
1/1 [==============================] - 0s 19ms/step - loss: 0.2201
Epoch 24/100
```

```
1/1 [==============================] - 0s 25ms/step - loss: 0.2150
Epoch 25/100
1/1 [==============================] - 0s 18ms/step - loss: 0.2098
Epoch 26/100
1/1 [==============================] - 0s 24ms/step - loss: 0.2046
Epoch 27/100
1/1 [==============================] - 0s 26ms/step - loss: 0.1993
Epoch 28/100
1/1 [==============================] - 0s 28ms/step - loss: 0.1939
Epoch 29/100
1/1 [==============================] - 0s 29ms/step - loss: 0.1885
Epoch 30/100
1/1 [==============================] - 0s 27ms/step - loss: 0.1830
Epoch 31/100
1/1 [==============================] - 0s 26ms/step - loss: 0.1776
Epoch 32/100
1/1 [==============================] - 0s 24ms/step - loss: 0.1721
Epoch 33/100
1/1 [==============================] - 0s 24ms/step - loss: 0.1667
Epoch 34/100
1/1 [==============================] - 0s 19ms/step - loss: 0.1612
Epoch 35/100
1/1 [==============================] - 0s 23ms/step - loss: 0.1558
Epoch 36/100
1/1 [==============================] - 0s 20ms/step - loss: 0.1504
Epoch 37/100
1/1 [==============================] - 0s 20ms/step - loss: 0.1449
Epoch 38/100
1/1 [==============================] - 0s 33ms/step - loss: 0.1395
Epoch 39/100
1/1 [==============================] - 0s 37ms/step - loss: 0.1342
Epoch 40/100
1/1 [==============================] - 0s 30ms/step - loss: 0.1288
Epoch 41/100
1/1 [==============================] - 0s 39ms/step - loss: 0.1235
Epoch 42/100
1/1 [==============================] - 0s 22ms/step - loss: 0.1183
Epoch 43/100
1/1 [==============================] - 0s 22ms/step - loss: 0.1131
Epoch 44/100
1/1 [==============================] - 0s 23ms/step - loss: 0.1080
Epoch 45/100
1/1 [==============================] - 0s 18ms/step - loss: 0.1030
Epoch 46/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0980
Epoch 47/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0931
Epoch 48/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0884
Epoch 49/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0837
Epoch 50/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0792
Epoch 51/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0747
Epoch 52/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0705
Epoch 53/100
1/1 [==============================] - 0s 26ms/step - loss: 0.0663
Epoch 54/100
1/1 [==============================] - 0s 28ms/step - loss: 0.0623
Epoch 55/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0585
Epoch 56/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0549
Epoch 57/100
```

```
1/1 [==============================] - 0s 27ms/step - loss: 0.0514
Epoch 58/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0481
Epoch 59/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0450
Epoch 60/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0421
Epoch 61/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0394
Epoch 62/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0369
Epoch 63/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0347
Epoch 64/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0326
Epoch 65/100
1/1 [==============================] - 0s 19ms/step - loss: 0.0308
Epoch 66/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0291
Epoch 67/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0277
Epoch 68/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0264
Epoch 69/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0253
Epoch 70/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0245
Epoch 71/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0237
Epoch 72/100
1/1 [==============================] - 0s 13ms/step - loss: 0.0231
Epoch 73/100
1/1 [==============================] - 0s 61ms/step - loss: 0.0227
Epoch 74/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0223
Epoch 75/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0221
Epoch 76/100
1/1 [==============================] - 0s 14ms/step - loss: 0.0219
Epoch 77/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0218
Epoch 78/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0218
Epoch 79/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0217
Epoch 80/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0217
Epoch 81/100
1/1 [==============================] - 0s 20ms/step - loss: 0.0217
Epoch 82/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0217
Epoch 83/100
1/1 [==============================] - 0s 13ms/step - loss: 0.0216
Epoch 84/100
1/1 [==============================] - 0s 17ms/step - loss: 0.0216
Epoch 85/100
1/1 [==============================] - 0s 15ms/step - loss: 0.0215
Epoch 86/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0214
Epoch 87/100
1/1 [==============================] - 0s 16ms/step - loss: 0.0213
Epoch 88/100
1/1 [==============================] - 0s 25ms/step - loss: 0.0211
Epoch 89/100
1/1 [==============================] - 0s 18ms/step - loss: 0.0209
Epoch 90/100
```

```
1/1 [==============================] - 0s 22ms/step - loss: 0.0207
Epoch 91/100
1/1 [==============================] - 0s 38ms/step - loss: 0.0205
Epoch 92/100
1/1 [==============================] - 0s 38ms/step - loss: 0.0203
Epoch 93/100
1/1 [==============================] - 0s 38ms/step - loss: 0.0200
Epoch 94/100
1/1 [==============================] - 0s 40ms/step - loss: 0.0198
Epoch 95/100
1/1 [==============================] - 0s 41ms/step - loss: 0.0195
Epoch 96/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0193
Epoch 97/100
1/1 [==============================] - 0s 31ms/step - loss: 0.0191
Epoch 98/100
1/1 [==============================] - 0s 24ms/step - loss: 0.0188
Epoch 99/100
1/1 [==============================] - 0s 23ms/step - loss: 0.0186
Epoch 100/100
1/1 [==============================] - 0s 21ms/step - loss: 0.0184
```

Out[49]:
```
<keras.callbacks.History at 0x20f5f60d370>
```

In [51]:
```
x_input = sequence[-4:]  # Last four entries from the sequence
x_input = scaler.transform(x_input.reshape(-1, 1)).flatten()  # Normalize input
x_input = x_input.reshape((1, n_seq, 1, n_steps, n_features))
yhat = model.predict(x_input, verbose=0)
print(yhat)
```

```
[[0.9882692]]
```

In [52]:
```
# Assuming 'scaler' is your MinMaxScaler instance and 'yhat' is your prediction
yhat_original = scaler.inverse_transform(yhat)
print("Predicted GDP value:", yhat_original)
```

```
Predicted GDP value: [[0.9882692]]
```

Multiple Input Series

In [58]:
```
import pandas as pd
import numpy as np
from numpy import array, hstack
```

In [59]:
```
data = pd.read_excel(r'C:\Users\begba\Desktop\praktika-master\PIB_Updated.xlsx')
```

In [60]:
```
print(data)
```

```
    Anul  Trimestrul       PIB  Rata angajare  Cheltuielile_totale
0   2019           1  252368.6        7892464              2347.00
1   2019           2  262370.8        8142162              2413.05
2   2019           3  267966.3        8154113              2570.57
3   2019           4  276961.0        8054722              2655.71
4   2020           1  272031.8        7954253              2551.13
5   2020           2  240078.7        7859102              2439.02
6   2020           3  265918.1        7978700              2680.12
7   2020           4  286616.6        7967302              2813.78
8   2021           1  283317.9        7300644              2813.30
9   2021           2  289048.2        7520145              2848.76
10  2021           3  299379.8        7510097              3045.97
11  2021           4  313454.7        7486105              3206.41
12  2022           1  331114.2        7439170              3267.71
13  2022           2  346335.5        7608656              3306.10
14  2022           3  355617.5        7552003              3562.13
15  2022           4  363828.5        7493612              3662.18
```

```
16  2023            1  384075.6         7378396                    3702.12
17  2023            2  391468.3         7417382                    3675.08
18  2023            3  402454.9         7455072                    4024.73
19  2023            4  415906.9         7466183                    4135.84
```

In [61]:
```
in_seq1 = data['Rata angajare'].values
in_seq2 = data['Cheltuielile_totale'].values
out_seq = data['PIB'].values
```

In [63]:
```
in_seq1 = in_seq1.reshape((len(in_seq1), 1))
in_seq2 = in_seq2.reshape((len(in_seq2), 1))
out_seq = out_seq.reshape((len(out_seq), 1))
```

In [64]:
```
dataset = hstack((in_seq1, in_seq2, out_seq))
```

In [65]:
```
n_steps = 3
```

In [66]:
```
def split_sequences(sequences, n_steps):
    X, y = list(), list()
    for i in range(len(sequences)):
        # Find the end of this pattern
        end_ix = i + n_steps
        # Check if we are beyond the dataset
        if end_ix > len(sequences):
            break
        # Gather input and output parts of the pattern
        seq_x, seq_y = sequences[i:end_ix, :-1], sequences[end_ix-1, -1]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)
```

In [67]:
```
X, y = split_sequences(dataset, n_steps)
print(X.shape, y.shape)
```

```
(18, 3, 2) (18,)
```

In [68]:
```
for i in range(len(X)):
    print(X[i], y[i])
```

```
[[7.892464e+06 2.347000e+03]
 [8.142162e+06 2.413050e+03]
 [8.154113e+06 2.570570e+03]] 267966.3
[[8.142162e+06 2.413050e+03]
 [8.154113e+06 2.570570e+03]
 [8.054722e+06 2.655710e+03]] 276961.0
[[8.154113e+06 2.570570e+03]
 [8.054722e+06 2.655710e+03]
 [7.954253e+06 2.551130e+03]] 272031.8
[[8.054722e+06 2.655710e+03]
 [7.954253e+06 2.551130e+03]
 [7.859102e+06 2.439020e+03]] 240078.7
[[7.954253e+06 2.551130e+03]
 [7.859102e+06 2.439020e+03]
 [7.978700e+06 2.680120e+03]] 265918.1
[[7.859102e+06 2.439020e+03]
 [7.978700e+06 2.680120e+03]
 [7.967302e+06 2.813780e+03]] 286616.6
[[7.978700e+06 2.680120e+03]
 [7.967302e+06 2.813780e+03]
 [7.300644e+06 2.813300e+03]] 283317.9
[[7.967302e+06 2.813780e+03]
 [7.300644e+06 2.813300e+03]
 [7.520145e+06 2.848760e+03]] 289048.2
[[7.300644e+06 2.813300e+03]
```

```
 [7.520145e+06 2.848760e+03]
 [7.510097e+06 3.045970e+03]] 299379.8
[[7.520145e+06 2.848760e+03]
 [7.510097e+06 3.045970e+03]
 [7.486105e+06 3.206410e+03]] 313454.7
[[7.510097e+06 3.045970e+03]
 [7.486105e+06 3.206410e+03]
 [7.439170e+06 3.267710e+03]] 331114.2
[[7.486105e+06 3.206410e+03]
 [7.439170e+06 3.267710e+03]
 [7.608656e+06 3.306100e+03]] 346335.5
[[7.439170e+06 3.267710e+03]
 [7.608656e+06 3.306100e+03]
 [7.552003e+06 3.562130e+03]] 355617.5
[[7.608656e+06 3.306100e+03]
 [7.552003e+06 3.562130e+03]
 [7.493612e+06 3.662180e+03]] 363828.5
[[7.552003e+06 3.562130e+03]
 [7.493612e+06 3.662180e+03]
 [7.378396e+06 3.702120e+03]] 384075.6
[[7.493612e+06 3.662180e+03]
 [7.378396e+06 3.702120e+03]
 [7.417382e+06 3.675080e+03]] 391468.3
[[7.378396e+06 3.702120e+03]
 [7.417382e+06 3.675080e+03]
 [7.455072e+06 4.024730e+03]] 402454.9
[[7.417382e+06 3.675080e+03]
 [7.455072e+06 4.024730e+03]
 [7.466183e+06 4.135840e+03]] 415906.9
```

In [69]:
```
print(data.head())
```

```
   Anul  Trimestrul       PIB  Rata angajare  Cheltuielile_totale
0  2019           1  252368.6        7892464              2347.00
1  2019           2  262370.8        8142162              2413.05
2  2019           3  267966.3        8154113              2570.57
3  2019           4  276961.0        8054722              2655.71
4  2020           1  272031.8        7954253              2551.13
```

In [70]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

In [71]:
```
print(data.columns)
```

```
Index(['Anul', 'Trimestrul', 'PIB', 'Rata angajare', 'Cheltuielile_totale'], dtype='obje
ct')
```

In [72]:
```
X = data[['Rata angajare', 'Cheltuielile_totale']]
y = data['PIB']
```

In [73]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
```

In [74]:
```
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)
```

Out[74]:
```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

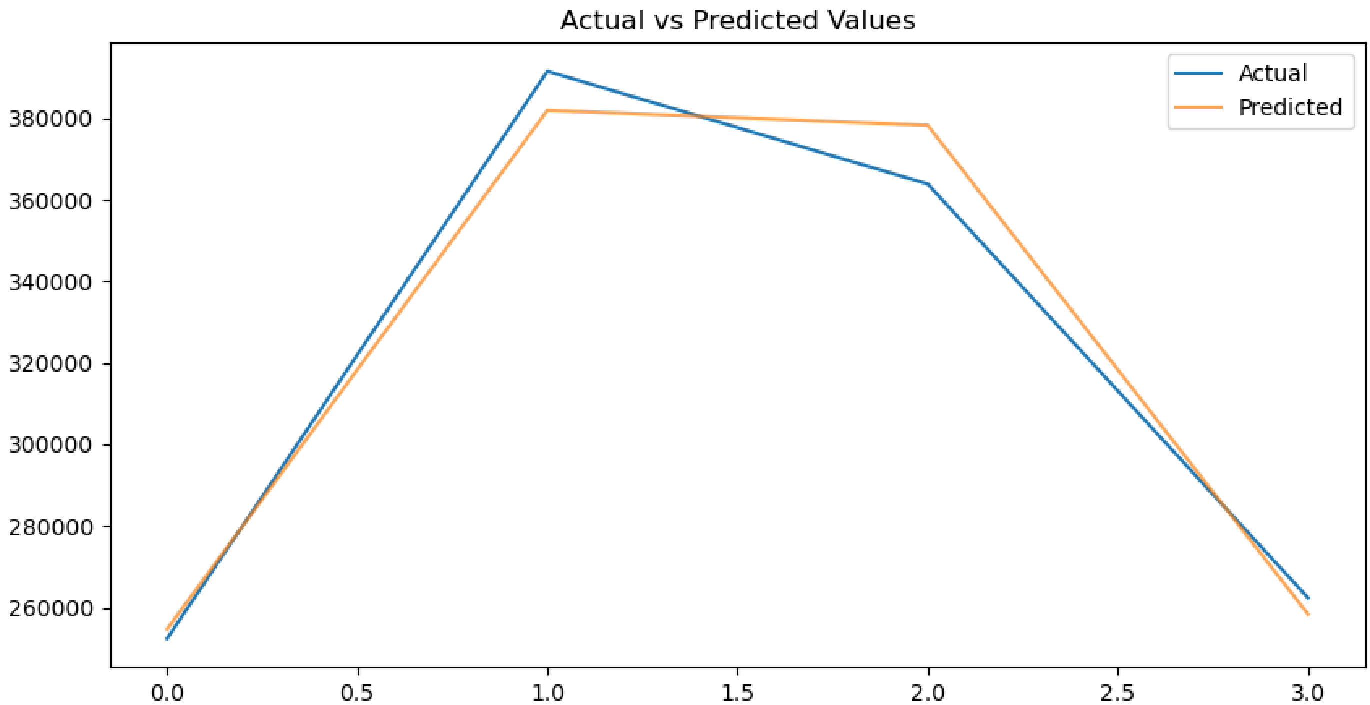
In [75]:
```
y_pred = model.predict(X_test)
```

In [76]:
```
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
```

In [77]:
```
print("Mean Absolute Error:", mae)
print("Root Mean Squared Error:", rmse)
```

```
Mean Absolute Error: 7580.220999999947
Root Mean Squared Error: 8953.360579273027
```

In [78]:
```
plt.figure(figsize=(10, 5))
plt.plot(y_test.reset_index(drop=True), label='Actual')
plt.plot(y_pred, label='Predicted', alpha=0.7)
plt.title('Actual vs Predicted Values')
plt.legend()
plt.show()
```



In [ ]: