

```

1 import numpy as np
2 import pandas as pd
3 from pandas import read_csv, DataFrame
4 from sklearn.preprocessing import MinMaxScaler
5 from sklearn.model_selection import train_test_split
6 from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
7 from sklearn.linear_model import LinearRegression
8 from tensorflow.keras import Sequential
9 from tensorflow.keras.layers import Dense, LSTM, Dropout
10 from tensorflow.keras.callbacks import History
11 from tensorflow.keras.optimizers import Adam, SGD
12 from matplotlib import pyplot
13 from numpy import array
14 from google.colab.drive import mount
15 from numpy import concatenate

```

```
1 mount('/content/drive', force_remount=True)
```

```
Mounted at /content/drive
```

```
1 df = read_csv('/content/drive/MyDrive/CPI.csv', index_col=0)
2 df.head()
```

	Đơn vị tính	Tháng 1/2009	Tháng 2/2009	Tháng 3/2009	Tháng 4/2009	Tháng 5/2009	Tháng 6/2009	Tháng 7/2009	Tháng 8/2009	Tháng 9/2009	...
Chỉ tiêu											
Chỉ số giá tiêu dùng	%	0.32	1.17	-0.17	0.35	0.44	0.55	0.52	0.24	0.62	..
Hàng ăn và											

```

1 selected_columns = df.columns[df.columns.get_loc('Tháng 1/2009'): df.columns.get_loc('Tháng 10/2023') + 1]
2
3 # Lấy giá trị của dòng đầu tiên trong mỗi cột
4 date_values = df[selected_columns].iloc[0].values
5
6 # Tạo DataFrame mới với 2 cột Date và CPI
7 df_new = pd.DataFrame({'Date': selected_columns, 'CPI': date_values})
8
9 # Kiểm tra và chuyển đổi cột "CPI" sang kiểu số
10 df_new['CPI'] = pd.to_numeric(df_new['CPI'], errors='coerce')
11
12 # Nhân giá trị của cột "CPI" với 100
13 df_new['CPI'] *= 100
14
15 # Hiển thị DataFrame mới
16 print(df_new)

```

	Date	CPI
0	Tháng 1/2009	32.0
1	Tháng 2/2009	117.0
2	Tháng 3/2009	-17.0
3	Tháng 4/2009	35.0
4	Tháng 5/2009	44.0
..	...	...
173	Tháng 6/2023	27.0
174	Tháng 7/2023	45.0
175	Tháng 8/2023	88.0
176	Tháng 9/2023	108.0
177	Tháng 10/2023	8.0

```
[178 rows x 2 columns]
```

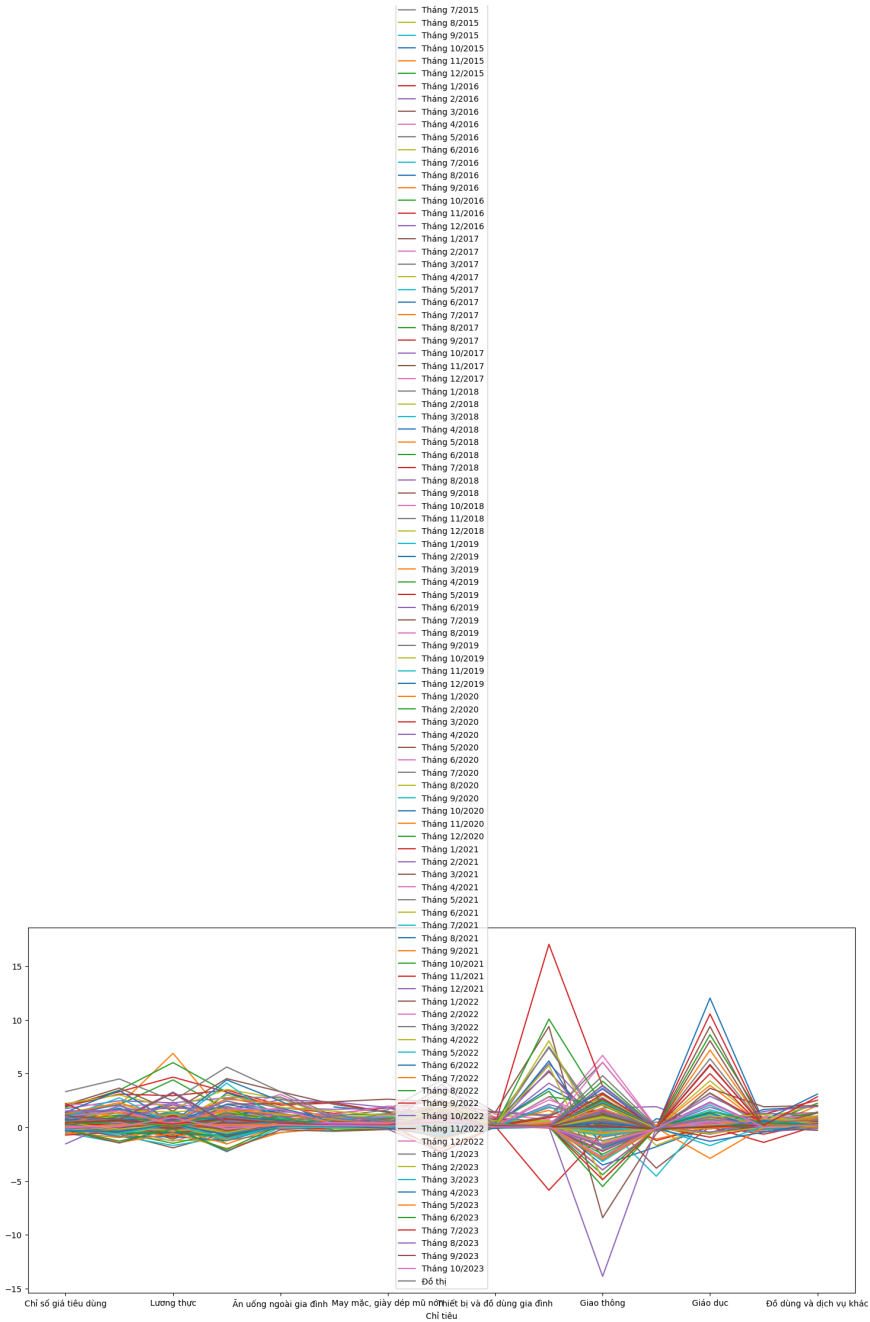
```

1 min_val = df_new['CPI'].min()
2 max_val = df_new['CPI'].max()
3 df_new['CPI'] = (df_new['CPI'] - min_val) / (max_val - min_val)

```

```
1 df.plot(figsize = (18, 8))
```

```
2  pyplot.show()
```





```

1 def split_sequence(arr, n_steps):
2     X = []
3     for i in range(len(arr) - n_steps):
4         X.append(arr[i:i + n_steps])
5     return array(X), array(arr[n_steps:])

1 n_steps = 6
2 X, y = split_sequence(df_new['CPI'].values, n_steps)
3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, shuffle=False)

1 X_train = X_train.reshape(X_train.shape[0], 1, X_train.shape[1])
2 X_test = X_test.reshape(X_test.shape[0], 1, X_test.shape[1])
3

1 model = Sequential()
2 model.add(LSTM(32, input_shape=(X_train.shape[1], X_train.shape[2]), return_sequences=True))
3 model.add(Dense(1))
4 model.compile(optimizer = Adam(learning_rate = 0.001), loss = 'mse', metrics=['mse'])
5 #model.compile(optimizer = 'adam', loss = 'mse', metrics=['mse'])
6 model.summary()

```

Model: "sequential"

Layer (type)	Output Shape	Param #
--------------	--------------	---------