In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#### impprt data
data = pd.read_excel(r"C:\Users\shubham lokare\Downloads\Data (1)\Data.xlsx")
```

In [3]:
```
data
```

Out[3]:

	Billing date	Variant	Economic Index	Industry Growth Rate (%)	Seasonality Factor
0	2022-08-18	XXX11	87.45	8.45	High
1	2022-08-19	XXX11	145.07	14.54	Medium
2	2022-08-20	XXX17	123.20	-2.96	Medium
3	2022-08-21	XXXV1	109.87	-4.83	Low
4	2022-08-22	XXX11	65.60	3.67	Low
...	...	...	...	...	...
3090	2023-03-30	XXXV5	95.03	11.05	Medium
3091	2023-07-23	XXX12	145.76	10.76	Low
3092	2023-07-25	XXX12	89.90	0.94	Low
3093	2024-04-30	XXXV2	133.98	-2.29	Low
3094	2024-04-30	XXXV2	68.85	-2.09	Medium

3095 rows × 5 columns

In [4]:
```
#### change the  billing date into date and time
### Convert 'Billing date' to datetime format
data['Billing date'] = pd.to_datetime(data['Billing date'])

# Sort the data by Billing date to ensure it's in sequence
data = data.sort_values('Billing date')
```

In [5]:
```
data
```

Out[5]:

	Billing date	Variant	Economic Index	Industry Growth Rate (%)	Seasonality Factor
0	2022-08-18	XXX11	87.45	8.45	High
1	2022-08-19	XXX11	145.07	14.54	Medium
1723	2022-08-20	XXXV3	60.10	12.76	High
1722	2022-08-20	XXXV3	80.33	11.39	Low
2	2022-08-20	XXX17	123.20	-2.96	Medium
...	...	...	...	...	...
1702	2024-04-30	XXXV2	81.18	10.66	Medium
1703	2024-04-30	XXXV2	100.61	-2.55	Medium
1704	2024-04-30	XXXV2	93.95	7.70	High
1697	2024-04-30	XXXV1	63.52	-3.80	Low
3094	2024-04-30	XXXV2	68.85	-2.09	Medium

3095 rows × 5 columns

In [8]:
```
#### top 10
data.head(10)
```

Out[8]:

	Billing date	Variant	Economic Index	Industry Growth Rate (%)	Seasonality Factor
0	2022-08-18	XXX11	87.45	8.45	High
1	2022-08-19	XXX11	145.07	14.54	Medium
1723	2022-08-20	XXXV3	60.10	12.76	High
1722	2022-08-20	XXXV3	80.33	11.39	Low
2	2022-08-20	XXX17	123.20	-2.96	Medium
3	2022-08-21	XXXV1	109.87	-4.83	Low
4	2022-08-22	XXX11	65.60	3.67	Low
1818	2022-08-22	XXX13	112.47	-1.09	Medium
1816	2022-08-22	XXX13	79.54	6.08	Low
1814	2022-08-22	XXX13	58.22	5.93	Medium

In [6]:
```
data.dtypes
```

Out[6]:
```
Billing date                datetime64[ns]
Variant                             object
Economic Index                     float64
Industry Growth Rate (%)           float64
Seasonality Factor                  object
dtype: object
```

In [7]:
```
### cponvert the categorical data into the numerical data

from sklearn.preprocessing import LabelEncoder
label = LabelEncoder()
```

In [8]:
```
data['new_Variant'] = label.fit_transform(data['Variant'])
data['new_seasonality'] = label.fit_transform(data['Seasonality Factor'])
```

In [9]:
```
data
```

Out[9]:

	Billing date	Variant	Economic Index	Industry Growth Rate (%)	Seasonality Factor	new_Variant	new_seasonality
0	2022-08-18	XXX11	87.45	8.45	High	0	0
1	2022-08-19	XXX11	145.07	14.54	Medium	0	2
1723	2022-08-20	XXXV3	60.10	12.76	High	8	0
1722	2022-08-20	XXXV3	80.33	11.39	Low	8	1
2	2022-08-20	XXX17	123.20	-2.96	Medium	4	2
...	...	...	...	...	...	...	...
1702	2024-04-30	XXXV2	81.18	10.66	Medium	7	2
1703	2024-04-30	XXXV2	100.61	-2.55	Medium	7	2
1704	2024-04-30	XXXV2	93.95	7.70	High	7	0
1697	2024-04-30	XXXV1	63.52	-3.80	Low	6	1
3094	2024-04-30	XXXV2	68.85	-2.09	Medium	7	2

3095 rows × 7 columns

In [14]:
```
#### droup unwanted columns
data.drop(['Variant' ,'Seasonality Factor'] ,axis = 1, inplace = True)
```

In [15]:
```
data
```

Out[15]:

	Billing date	Economic Index	Industry Growth Rate (%)	new_Variant	new_seasonality
0	2022-08-18	87.45	8.45	0	0
1	2022-08-19	145.07	14.54	0	2
1723	2022-08-20	60.10	12.76	8	0
1722	2022-08-20	80.33	11.39	8	1
2	2022-08-20	123.20	-2.96	4	2
...	...	...	...	...	...
1702	2024-04-30	81.18	10.66	7	2
1703	2024-04-30	100.61	-2.55	7	2
1704	2024-04-30	93.95	7.70	7	0
1697	2024-04-30	63.52	-3.80	6	1
3094	2024-04-30	68.85	-2.09	7	2

3095 rows × 5 columns

In [11]:

```
data.describe()
```

Out[11]:

	Billing date	Economic Index	Industry Growth Rate (%)	new_Variant	new_seasonality
count	3095	3095.000000	3095.00000	3095.000000	3095.000000
mean	2023-07-05 18:49:40.032310016	99.936892	4.86347	5.597092	0.978029
min	2022-08-18 00:00:00	50.000000	-5.00000	0.000000	0.000000
25%	2023-02-13 00:00:00	74.690000	-0.17000	3.000000	0.000000
50%	2023-06-27 00:00:00	100.480000	4.76000	6.000000	1.000000
75%	2023-11-30 00:00:00	125.130000	9.83000	7.000000	2.000000
max	2024-04-30 00:00:00	149.970000	14.99000	11.000000	2.000000
std	NaN	29.131357	5.75851	2.977755	0.818836

In [12]:

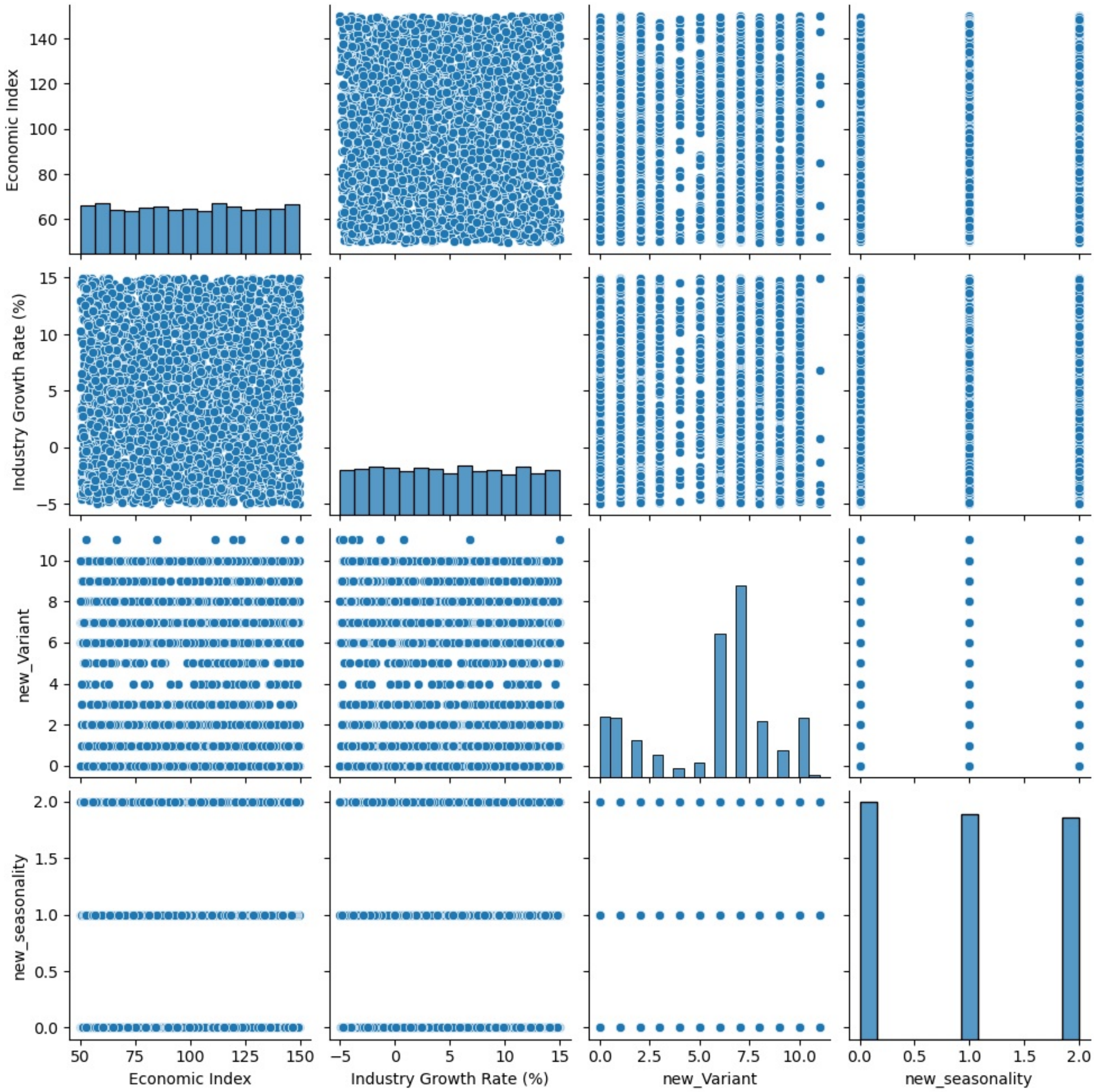
```
### check missing values
data.isna().sum()                #### there is no missing values
```

Out[12]:

```
Billing date               0
Variant                    0
Economic Index             0
Industry Growth Rate (%)   0
Seasonality Factor         0
new_Variant                0
new_seasonality            0
dtype: int64
```

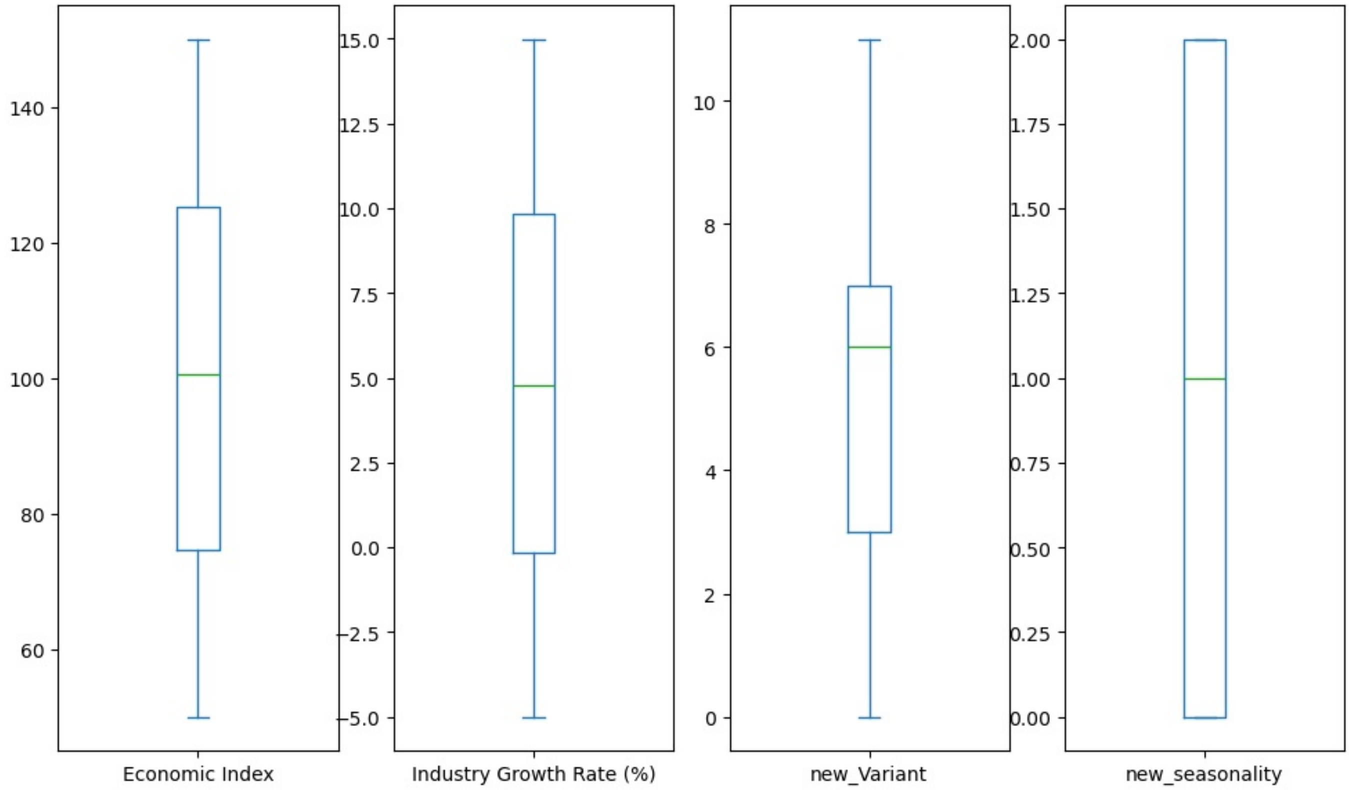
In [13]:

```
### check the shape of data and corr
sns.pairplot(data)
plt.show()
```



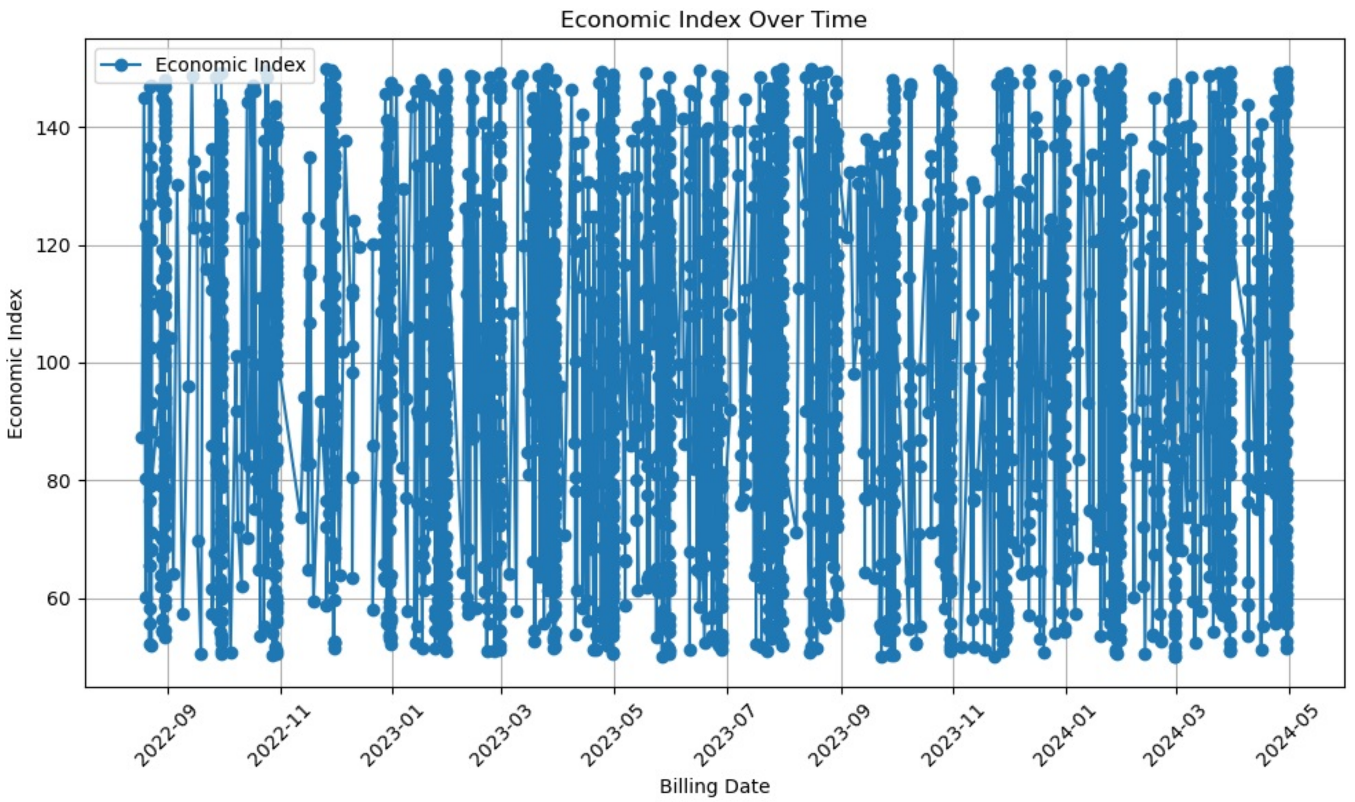
In [14]:
```
#### check outliers
data.plot(kind='box' ,subplots = True , figsize =(12,7))        ##### there is no outliers in data
```

Out[14]:
```
Economic Index                 Axes(0.125,0.11;0.168478x0.77)
Industry Growth Rate (%)    Axes(0.327174,0.11;0.168478x0.77)
new_Variant                 Axes(0.529348,0.11;0.168478x0.77)
new_seasonality             Axes(0.731522,0.11;0.168478x0.77)
dtype: object
```

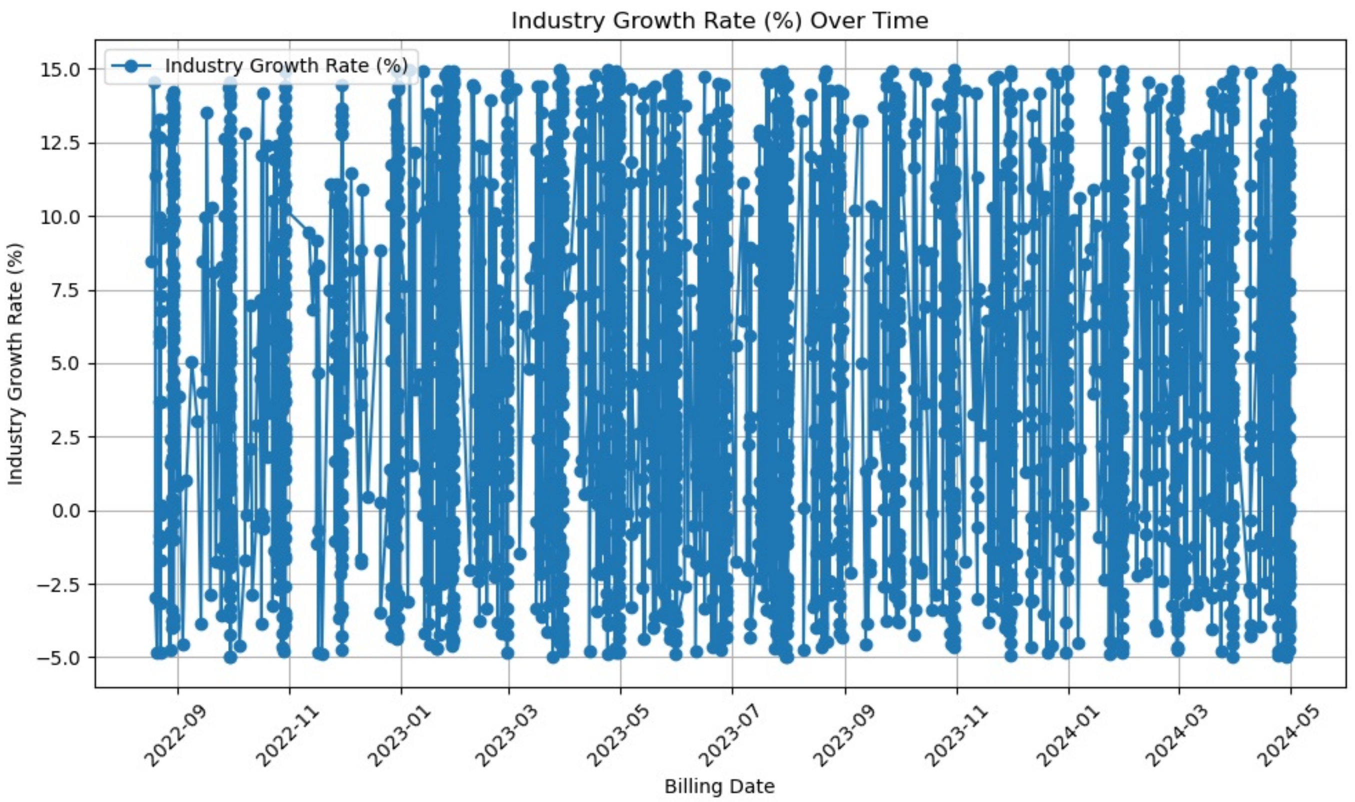


In [15]:

```
#### time series graph  for  economic index and billing date
# Plotting the time series (Economic Index over Billing Date)
plt.figure(figsize=(10, 6))
plt.plot(data['Billing date'], data['Economic Index'], marker='o', label='Economic Index')
plt.xlabel('Billing Date')
plt.ylabel('Economic Index')
plt.title('Economic Index Over Time')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.legend()
plt.show()
```



In [16]:
```
# Plotting the time series (Industry Growth Rate over Billing Date)
plt.figure(figsize=(10, 6))
plt.plot(data['Billing date'], data['Industry Growth Rate (%)'], marker='o', label='Industry Growth Rate (%)')
plt.xlabel('Billing Date')
plt.ylabel('Industry Growth Rate (%)')
plt.title('Industry Growth Rate (%) Over Time')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.legend()
plt.show()
```



In [17]:
```
#### then make data into same scale
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

In [20]:
```
# Scale the selected columns
scaled_data = scaler.fit_transform(data[['Economic Index', 'Industry Growth Rate (%)' ,'new_seasonality','new_Va
```

```
scaled_data
```

Out[20]:
```
array([[0.37461238, 0.67283642, 0.        , 0.        ],
       [0.9509853 , 0.97748874, 1.        , 0.        ],
       [0.10103031, 0.88844422, 0.        , 0.72727273],
       ...,
       [0.43963189, 0.63531766, 0.        , 0.63636364],
       [0.13524057, 0.06003002, 0.5       , 0.54545455],
       [0.18855657, 0.14557279, 1.        , 0.63636364]])
```

In [22]:
```
### split the data
y= data['new_Variant']
```

In [23]:
```
### apply the model
### we used the LSTM model
### apply the model
### here we use LSTM model for training
from keras.models import Sequential
from keras.layers import LSTM, Dense, Dropout
from sklearn.model_selection import train_test_split
```

In [24]:
```
##### using LSTM

# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, Y_test = train_test_split(scaled_data,y, test_size=0.2, random_state=42)
```

In [25]:
```
#### we need make the data into sequence
# Reshape for LSTM [samples, time steps, features]
X_train = X_train.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test = X_test.reshape((X_test.shape[0], 1, X_test.shape[1]))
```

In [26]:
```
X_train
```

Out[26]:
```
array([[[0.50605182, 0.71535768, 1.        , 0.90909091]],

       [[0.86836051, 0.08804402, 0.        , 0.90909091]],

       [[0.2849855 , 0.76488244, 1.        , 0.72727273]],

       ...,

       [[0.11493448, 0.96098049, 0.        , 0.54545455]],

       [[0.11383415, 0.24762381, 1.        , 0.        ]],

       [[0.73171952, 0.70135068, 1.        , 0.90909091]]])
```

In [27]:
```
X_test
```

Out[27]:
```
array([[[0.32159648, 0.83491746, 1.        , 0.54545455]],

       [[0.39471842, 0.73686843, 0.        , 0.54545455]],

       [[0.40952286, 0.67983992, 1.        , 0.        ]],

       ...,

       [[0.17655297, 0.75387694, 1.        , 0.63636364]],

       [[0.66519956, 0.46073037, 0.5       , 0.63636364]],

       [[0.93317995, 0.70085043, 0.        , 0.90909091]]])
```

In [28]:
```
# Build the LSTM model
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
```

```
C:\anaconda\Lib\site-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)
```

In [30]:
```
# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')
```

In [31]:
```
# Train the model
model.fit(X_train, y_train, epochs=100, batch_size=32, verbose=1)
```

```
Epoch 1/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step - loss: 38.4906
Epoch 2/100
```

```
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 27.3299
Epoch 3/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 12.7563
Epoch 4/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 7.2132
Epoch 5/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 5.2666
Epoch 6/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - loss: 3.6054
Epoch 7/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 2.2497
Epoch 8/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.4449
Epoch 9/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.0824
Epoch 10/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 1.0516
Epoch 11/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.9719
Epoch 12/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.9375
Epoch 13/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.8689
Epoch 14/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.8431
Epoch 15/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.8030
Epoch 16/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.7464
Epoch 17/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.8231
Epoch 18/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.6524
Epoch 19/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.6694
Epoch 20/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.7143
Epoch 21/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.6458
Epoch 22/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.5740
Epoch 23/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5927
Epoch 24/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.5577
Epoch 25/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.5750
Epoch 26/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.5413
Epoch 27/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.5292
Epoch 28/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.5224
Epoch 29/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.5254
Epoch 30/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.4815
Epoch 31/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.4476
Epoch 32/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.4557
Epoch 33/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.4600
Epoch 34/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - loss: 0.4410
Epoch 35/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.4577
Epoch 36/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4448
Epoch 37/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.4871
Epoch 38/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.4182
Epoch 39/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.3891
Epoch 40/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4167
Epoch 41/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3998
Epoch 42/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.3837
Epoch 43/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4174
```

```
Epoch 44/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3987
Epoch 45/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4326
Epoch 46/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.4117
Epoch 47/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 0.4205
Epoch 48/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.4137
Epoch 49/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3821
Epoch 50/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3734
Epoch 51/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3682
Epoch 52/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3571
Epoch 53/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3663
Epoch 54/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3725
Epoch 55/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3478
Epoch 56/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3590
Epoch 57/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3548
Epoch 58/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3642
Epoch 59/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3472
Epoch 60/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3366
Epoch 61/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3553
Epoch 62/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3305
Epoch 63/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3616
Epoch 64/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3366
Epoch 65/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3148
Epoch 66/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3256
Epoch 67/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3237
Epoch 68/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3679
Epoch 69/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3533
Epoch 70/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3426
Epoch 71/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3320
Epoch 72/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3239
Epoch 73/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3246
Epoch 74/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3469
Epoch 75/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.3329
Epoch 76/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3245
Epoch 77/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3486
Epoch 78/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.3324
Epoch 79/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3208
Epoch 80/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3252
Epoch 81/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.3141
Epoch 82/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.3112
Epoch 83/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2899
Epoch 84/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3239
Epoch 85/100
```

```
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3080
Epoch 86/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2976
Epoch 87/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2784
Epoch 88/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.3001
Epoch 89/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2802
Epoch 90/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2783
Epoch 91/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2902
Epoch 92/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2785
Epoch 93/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2860
Epoch 94/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2904
Epoch 95/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2834
Epoch 96/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2775
Epoch 97/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2564
Epoch 98/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2803
Epoch 99/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.2727
Epoch 100/100
78/78 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.2728
```

Out[31]:

```
<keras.src.callbacks.history.History at 0x28762f29b80>
```

In [32]:

```python
### the prediction

y_pred = model.predict(X_test)
```

```
20/20 ━━━━━━━━━━━━━━━━━━━━ 1s 30ms/step
```

In [33]:

```python
y_pred
```

Out[33]:

```
array([[ 6.021022  ],
       [ 6.08987   ],
       [ 0.15105033],
       [10.135213  ],
       [10.135684  ],
       [ 0.25777793],
       [ 0.13986647],
       [ 7.01881   ],
       [ 8.004158  ],
       [ 9.022272  ],
       [ 8.013696  ],
       [ 3.152238  ],
       [ 7.0128937 ],
       [ 6.0757046 ],
       [ 1.0897734 ],
       [ 6.0989094 ],
       [ 8.009247  ],
       [ 9.030526  ],
       [ 6.048953  ],
       [ 5.155269  ],
       [ 1.0751231 ],
       [10.115043  ],
       [ 6.095577  ],
       [ 6.0947137 ],
       [ 5.9963965 ],
       [ 8.017192  ],
       [ 7.0145755 ],
       [ 6.1047015 ],
       [10.077783  ],
       [ 0.14927137],
       [ 9.045931  ],
       [10.10702   ],
       [ 1.095721  ],
       [ 7.977317  ],
       [ 7.016039  ],
       [ 1.1436589 ],
       [ 6.103563  ],
       [ 6.9930496 ],
       [ 7.968339  ],
       [ 7.0156984 ],
       [10.123499  ],
       [ 5.124292  ],
```

```
[ 6.0984755 ],
[ 7.012118  ],
[10.11942   ],
[ 0.17086637],
[ 3.1484861 ],
[10.147792  ],
[ 7.01762   ],
[ 6.061657  ],
[ 6.0025125 ],
[ 7.0102386 ],
[ 0.15575695],
[ 1.0449927 ],
[ 5.154617  ],
[ 6.062337  ],
[10.168031  ],
[ 1.0772345 ],
[ 6.053922  ],
[ 2.0905032 ],
[ 6.0435543 ],
[ 7.9853168 ],
[ 6.0645843 ],
[ 7.0156107 ],
[ 6.0841885 ],
[ 0.10556722],
[ 8.026168  ],
[ 7.972944  ],
[ 8.0385895 ],
[ 5.1447697 ],
[ 7.0264425 ],
[ 6.051381  ],
[ 0.14795387],
[ 2.1189637 ],
[ 6.105429  ],
[ 1.1019619 ],
[ 7.035377  ],
[ 6.0715833 ],
[10.158263  ],
[ 2.1213634 ],
[ 6.0762806 ],
[ 6.0345864 ],
[ 6.0604105 ],
[10.154638  ],
[ 0.12675989],
[ 0.16446877],
[ 6.077783  ],
[ 1.1237217 ],
[ 8.038822  ],
[ 7.968453  ],
[ 6.0412817 ],
[ 6.0330086 ],
[ 6.078748  ],
[ 2.0947423 ],
[ 7.0216985 ],
[ 7.0293813 ],
[10.0673    ],
[ 6.982815  ],
[ 0.19470549],
[10.106476  ],
[ 7.025831  ],
[ 6.970239  ],
[10.10688   ],
[ 0.14580739],
[ 6.9892526 ],
[ 6.9396753 ],
[ 7.015398  ],
[ 7.026218  ],
[10.111669  ],
[11.279507  ],
[ 6.0567017 ],
[ 7.019845  ],
[ 5.1328464 ],
[ 1.067686  ],
[ 7.0002    ],
[ 7.008404  ],
[ 8.004228  ],
[ 6.9784093 ],
[ 6.9596753 ],
[ 7.0188885 ],
[ 6.9878187 ],
[ 0.14058399],
[ 7.012105  ],
[ 7.0181713 ],
[10.133175  ],
```

```
[ 7.003588  ],
[ 6.046903  ],
[ 6.10488   ],
[ 7.041297  ],
[ 8.002066  ],
[ 8.014239  ],
[ 0.15348768],
[ 6.991355  ],
[ 7.975765  ],
[ 3.1306376 ],
[ 0.12188244],
[ 7.0293627 ],
[ 6.051733  ],
[ 7.9861493 ],
[ 7.9985085 ],
[ 2.1298497 ],
[ 6.9982967 ],
[10.059594  ],
[ 0.14472318],
[ 6.0507197 ],
[ 6.0446296 ],
[ 6.988912  ],
[10.118561  ],
[ 5.10665   ],
[ 6.1032033 ],
[10.122908  ],
[ 0.152601  ],
[ 0.13889027],
[ 7.0166297 ],
[ 6.0361176 ],
[ 7.016828  ],
[ 6.0599265 ],
[ 6.981886  ],
[ 0.19035256],
[ 9.05615   ],
[ 8.014699  ],
[ 2.0886688 ],
[ 7.0092373 ],
[ 6.994029  ],
[ 6.0473185 ],
[ 6.024563  ],
[ 0.11874425],
[ 1.0685456 ],
[ 6.0345902 ],
[ 7.964611  ],
[ 6.1003065 ],
[10.078101  ],
[ 0.15223849],
[ 3.1292305 ],
[ 1.0744069 ],
[ 9.0347595 ],
[10.101339  ],
[ 6.1040154 ],
[ 0.16059685],
[ 6.9794235 ],
[ 7.018049  ],
[ 5.1122847 ],
[ 1.0698273 ],
[ 1.1121032 ],
[ 2.0967793 ],
[ 6.0708876 ],
[ 7.0381737 ],
[ 8.022687  ],
[ 1.145097  ],
[ 3.1193805 ],
[ 6.967818  ],
[ 6.9778724 ],
[ 6.0241723 ],
[ 6.9907336 ],
[ 1.0826912 ],
[ 7.021432  ],
[ 2.133772  ],
[ 7.968285  ],
[ 6.1052976 ],
[ 7.008205  ],
[ 0.15308404],
[ 0.14709365],
[ 6.9785366 ],
[ 7.995235  ],
[ 6.0667596 ],
[ 7.001771  ],
[ 6.0828    ],
[ 3.1469283 ],
```

```
[ 8.005199  ],
[10.065825  ],
[ 7.0197763 ],
[ 7.0179963 ],
[ 7.0048018 ],
[ 9.025569  ],
[10.1405735 ],
[10.145394  ],
[ 6.0376797 ],
[ 6.05797   ],
[ 8.014056  ],
[ 0.14193821],
[ 6.082362  ],
[10.158551  ],
[ 2.0942256 ],
[ 7.0293536 ],
[ 1.0599278 ],
[ 3.1147854 ],
[ 0.11240876],
[10.139358  ],
[ 6.991307  ],
[ 2.1478975 ],
[ 7.0396204 ],
[ 2.1198978 ],
[ 7.0265355 ],
[ 3.147152  ],
[ 1.112398  ],
[ 2.0870779 ],
[ 7.017908  ],
[10.125765  ],
[10.131456  ],
[ 2.1053057 ],
[ 6.0385294 ],
[ 2.072528  ],
[ 6.991527  ],
[ 2.099603  ],
[ 6.9886518 ],
[ 2.1387255 ],
[ 2.075268  ],
[ 1.0741842 ],
[ 0.17106438],
[ 6.075982  ],
[ 6.0716996 ],
[ 7.9858747 ],
[ 6.9998026 ],
[ 6.994182  ],
[ 7.9698563 ],
[ 5.1321254 ],
[ 6.9917736 ],
[ 6.101045  ],
[ 6.0859065 ],
[ 9.035705  ],
[ 6.960903  ],
[ 1.0698173 ],
[ 3.1318803 ],
[10.062216  ],
[ 6.9984813 ],
[ 1.0581108 ],
[ 0.13272929],
[ 6.9649563 ],
[ 0.1439507 ],
[ 6.051817  ],
[ 6.0856676 ],
[ 6.979359  ],
[ 0.10885882],
[ 7.956255  ],
[ 8.008529  ],
[ 5.1019006 ],
[10.099798  ],
[ 6.0855255 ],
[ 6.080196  ],
[ 7.9797306 ],
[ 7.033246  ],
[ 7.036168  ],
[ 6.036166  ],
[ 9.020245  ],
[ 6.970639  ],
[ 6.964568  ],
[ 7.0083475 ],
[ 2.0877771 ],
[10.097232  ],
[ 8.979257  ],
[ 6.989849  ],
```

```
[ 2.1698449 ],
[ 5.151864  ],
[ 1.0871863 ],
[ 6.047021  ],
[ 0.17319024],
[ 8.035996  ],
[ 6.0540147 ],
[ 0.1410693 ],
[ 1.098583  ],
[ 7.9571137 ],
[ 6.985843  ],
[ 5.1671095 ],
[ 6.0232573 ],
[ 7.0166817 ],
[ 6.0150747 ],
[10.118073  ],
[ 6.9897046 ],
[ 6.983708  ],
[ 6.9907947 ],
[ 1.0994804 ],
[ 7.01289   ],
[ 7.007201  ],
[10.084327  ],
[ 3.1507874 ],
[ 6.097691  ],
[ 7.0002675 ],
[ 7.0290146 ],
[ 0.16953826],
[ 6.9893527 ],
[ 8.021954  ],
[ 7.972704  ],
[ 2.1181712 ],
[ 1.10695   ],
[ 6.0878816 ],
[ 6.0564227 ],
[ 7.0057187 ],
[ 8.000359  ],
[ 6.985748  ],
[ 7.0381737 ],
[ 7.029575  ],
[ 7.961685  ],
[ 1.0537764 ],
[ 6.085584  ],
[ 6.0832653 ],
[10.134758  ],
[ 3.1492605 ],
[ 6.032818  ],
[ 7.0124745 ],
[ 3.138287  ],
[ 8.0160265 ],
[ 7.0349007 ],
[ 6.977312  ],
[ 6.0971055 ],
[ 7.978688  ],
[ 4.128568  ],
[ 6.0257773 ],
[ 7.013791  ],
[ 7.026543  ],
[ 6.9797983 ],
[ 0.11399448],
[ 6.052856  ],
[ 7.016554  ],
[ 6.0074987 ],
[ 6.062375  ],
[ 3.1667285 ],
[10.083578  ],
[10.132639  ],
[ 6.0319934 ],
[ 8.011808  ],
[ 7.000949  ],
[ 2.0751724 ],
[ 7.0122805 ],
[ 2.106908  ],
[ 7.01355   ],
[ 6.991316  ],
[ 6.059882  ],
[ 6.05983   ],
[ 1.0772611 ],
[ 7.0179605 ],
[ 6.0470686 ],
[ 6.1055903 ],
[ 7.0359225 ],
[ 6.0438867 ],
```

```
[ 0.14733052],
[ 7.009324  ],
[ 7.0174894 ],
[ 0.13936198],
[ 6.055589  ],
[ 6.03208   ],
[ 6.9942956 ],
[ 1.0913483 ],
[ 3.1653435 ],
[10.139615  ],
[ 7.0135345 ],
[ 5.1637034 ],
[ 6.9931765 ],
[ 6.06777   ],
[ 0.17511189],
[ 3.1624012 ],
[10.138747  ],
[ 7.019726  ],
[ 6.0465183 ],
[10.123271  ],
[ 7.040803  ],
[ 6.0519266 ],
[ 4.1397257 ],
[ 0.11990225],
[10.138334  ],
[ 8.007205  ],
[ 6.080464  ],
[ 5.0851912 ],
[ 1.0813442 ],
[10.111879  ],
[ 5.1559553 ],
[ 4.155747  ],
[ 0.23315954],
[ 1.1212332 ],
[ 6.970155  ],
[ 1.0706022 ],
[ 6.0651226 ],
[ 9.067891  ],
[ 6.0010185 ],
[ 8.031996  ],
[ 2.154224  ],
[ 6.994143  ],
[10.13823   ],
[ 0.16280746],
[ 6.0831695 ],
[ 7.005022  ],
[ 7.01261   ],
[ 0.17929244],
[ 6.096511  ],
[ 6.949476  ],
[ 6.03543   ],
[ 2.1251478 ],
[ 0.24091113],
[ 6.0815244 ],
[10.136335  ],
[ 6.947097  ],
[ 0.16627944],
[10.121002  ],
[ 1.0775511 ],
[ 7.039083  ],
[ 7.0201216 ],
[ 7.0045047 ],
[ 3.117536  ],
[ 7.003337  ],
[ 2.0896657 ],
[ 6.946771  ],
[ 6.037608  ],
[ 0.15865898],
[ 6.9552035 ],
[ 9.020374  ],
[10.13215   ],
[ 6.977142  ],
[ 0.18007064],
[ 1.0479999 ],
[ 6.9608545 ],
[10.130493  ],
[ 6.051137  ],
[ 5.1490717 ],
[10.131065  ],
[ 6.9930167 ],
[ 2.1470292 ],
[ 6.0823236 ],
[ 6.097099  ],
```

```
[ 7.0176115 ],
[ 1.1202385 ],
[ 6.9891653 ],
[ 8.9946    ],
[ 7.0180163 ],
[ 7.0316796 ],
[10.104961  ],
[ 9.03176   ],
[ 7.0069656 ],
[ 6.9777517 ],
[ 6.984913  ],
[ 2.1350722 ],
[ 0.1232475 ],
[ 5.144285  ],
[ 0.12068748],
[ 1.1247084 ],
[ 0.13513565],
[ 6.080742  ],
[ 7.039024  ],
[ 0.1454091 ],
[ 2.12255   ],
[ 2.1687946 ],
[ 8.037793  ],
[ 9.036536  ],
[ 6.983648  ],
[ 8.002326  ],
[ 7.0134487 ],
[10.128053  ],
[ 7.0189853 ],
[ 6.023443  ],
[ 0.15162289],
[ 1.0817007 ],
[ 3.136204  ],
[ 6.0645185 ],
[ 0.1564902 ],
[ 1.0626566 ],
[ 7.01287   ],
[ 4.1313553 ],
[ 4.1653404 ],
[ 7.031764  ],
[ 0.15695155],
[ 6.059165  ],
[ 8.012451  ],
[ 7.006834  ],
[ 6.9803247 ],
[ 7.0401597 ],
[ 3.145199  ],
[ 3.147925  ],
[ 1.0598049 ],
[ 6.9936905 ],
[10.097082  ],
[ 7.974835  ],
[ 0.16662729],
[ 2.1187217 ],
[ 6.051141  ],
[ 8.038301  ],
[ 7.0252776 ],
[ 6.995123  ],
[10.007093  ],
[ 5.076742  ],
[ 7.0071073 ],
[ 6.985359  ],
[ 7.0211425 ],
[ 0.1344353 ],
[ 8.012635  ],
[ 8.028751  ],
[ 6.098425  ],
[ 6.9975986 ],
[ 0.1227392 ],
[ 1.0729189 ],
[ 3.1375577 ],
[ 7.0055733 ],
[ 7.0390105 ],
[ 6.9776406 ],
[ 6.0544024 ],
[10.156045  ],
[ 7.973799  ],
[ 6.0500813 ],
[ 7.0197644 ],
[ 5.1257815 ],
[ 5.1261635 ],
[ 1.0790688 ],
[ 7.0139956 ],
```

```
       [ 6.987696  ],
       [ 0.12784445],
       [ 7.018601  ],
       [ 7.0070868 ],
       [10.110576  ],
       [ 5.102306  ],
       [ 7.998186  ],
       [ 6.064823  ],
       [ 6.0555744 ],
       [ 0.25017595],
       [ 1.0901184 ],
       [10.129139  ],
       [ 7.03703   ],
       [ 3.150619  ],
       [ 6.9768305 ],
       [ 6.988162  ],
       [ 7.0056663 ],
       [ 7.037406  ],
       [ 7.0048227 ],
       [ 7.016779  ],
       [ 6.058378  ],
       [ 1.0743792 ],
       [ 7.0407934 ],
       [ 1.0636164 ],
       [ 6.050733  ],
       [ 6.058912  ],
       [ 6.972298  ],
       [ 6.985892  ],
       [ 6.989049  ],
       [ 6.9832587 ],
       [ 8.038071  ],
       [ 7.96829   ],
       [ 5.1357903 ],
       [ 6.104649  ],
       [10.130997  ],
       [ 0.12494934],
       [ 7.032377  ],
       [ 0.19085217],
       [ 7.0060625 ],
       [ 7.0334563 ],
       [ 1.0771508 ],
       [ 6.989777  ],
       [ 7.0148945 ],
       [ 1.0899844 ],
       [ 1.091955  ],
       [ 6.991896  ],
       [ 7.9856057 ],
       [ 1.0810583 ],
       [ 6.0216227 ],
       [ 6.0793543 ],
       [10.162623  ],
       [ 0.15709305],
       [ 3.1443348 ],
       [ 7.016975  ],
       [ 7.019183  ],
       [10.10254   ],
       [ 6.992727  ],
       [ 4.1696205 ],
       [ 1.0963681 ],
       [ 3.132512  ],
       [ 9.032905  ],
       [ 7.0008364 ],
       [ 6.0770116 ],
       [ 0.19227064],
       [ 6.053252  ],
       [ 8.038356  ],
       [ 6.070347  ],
       [ 6.097933  ],
       [ 3.1503062 ],
       [ 7.0190625 ],
       [ 8.008156  ],
       [ 6.9556327 ],
       [ 6.072444  ],
       [ 9.021065  ],
       [ 3.1460419 ],
       [ 7.00875   ],
       [ 6.9744043 ],
       [ 7.019066  ],
       [10.182877  ]], dtype=float32)
```

In [35]:

```
#### then we find mean absoulate error ,r2 score , mean absoulate precentage error

from sklearn.metrics import mean_absolute_error, r2_score
```

In [37]:
```python
mean= mean_absolute_error(Y_test,y_pred)
print(mean)
```

```
0.07077863663194253
```

In [38]:
```python
### r2score
rscore = r2_score(Y_test ,y_pred)
print(rscore)
```

```
0.9990970734978895
```

In [39]:
```python
### MAPE

def mean_absolute_percentage_error(y_true, y_pred):
    # Avoid division by zero
    non_zero_idx = y_true != 0
    return np.mean(np.abs((y_true[non_zero_idx] - y_pred[non_zero_idx]) / y_true[non_zero_idx])) * 100
```

In [40]:
```python
# Assuming y_test and y_pred1 are already defined and flattened
y_test = Y_test.values.flatten()  # or Y_test.to_numpy().flatten()
y_pred = y_pred.flatten()  # y_predict should be a numpy array already
```

In [41]:
```python
mape = mean_absolute_percentage_error(y_test, y_pred)
print(f"Mean Absolute Percentage Error (MAPE): {mape}%")
```

```
Mean Absolute Percentage Error (MAPE): 1.8541176126035415%
```

In [42]:
```python
### then we need to find out the confidance intervel

### find resudiual error

error = Y_test-y_pred

print(error)
```

```
2572   -0.021022
2317   -0.089870
913    -0.151050
2863   -0.135213
2781   -0.135684
          ...
2915   -0.146042
1537   -0.008750
1095    0.025596
2189   -0.019066
2949   -0.182877
Name: new_Variant, Length: 619, dtype: float64
```

In [44]:
```python
#### the we need to find ou standered std
from scipy import stats

dev = np.std(error)
print(dev)
```

```
0.06239677121224959
```

In [45]:
```python
####  z value = 95 %
z = 1.96
# Compute the confidence interval for each prediction
ci_lower = y_pred - z * dev
ci_upper = y_pred + z * dev
```

In [47]:
```python
print(ci_lower)
```

```
[ 5.89872408e+00  5.96757221e+00  2.87526548e-02  1.00129156e+01
  1.00133867e+01  1.35480255e-01  1.75687969e-02  6.89651203e+00
  7.88186026e+00  8.89997482e+00  7.89139795e+00  3.02994013e+00
  6.89059591e+00  5.95340681e+00  9.67475772e-01  5.97661161e+00
  7.88694906e+00  8.90822887e+00  5.92665529e+00  5.03297138e+00
  9.52825427e-01  9.99274540e+00  5.97327900e+00  5.97241592e+00
  5.87409878e+00  7.89489412e+00  6.89227772e+00  5.98240376e+00
  9.95548534e+00  2.69736946e-02  8.92363358e+00  9.98472309e+00
  9.73423362e-01  7.85501909e+00  6.89374113e+00  1.02136123e+00
  5.98126507e+00  6.87075186e+00  7.84604120e+00  6.89340067e+00
  1.00012016e+01  5.00199413e+00  5.97617769e+00  6.88982010e+00
  9.99712276e+00  4.85686958e-02  3.02618837e+00  1.00254946e+01
  6.89532232e+00  5.93935919e+00  5.88021469e+00  6.88794088e+00
  3.34592760e-02  9.22695041e-01  5.03231907e+00  5.94003916e+00
  1.00457335e+01  9.54936862e-01  5.93162441e+00  1.96820557e+00
  5.92125654e+00  7.86301899e+00  5.94228649e+00  6.89331293e+00
  5.96189070e+00 -1.67304575e-02  7.90387011e+00  7.85064602e+00
  7.91629171e+00  5.02247190e+00  6.90414476e+00  5.92908335e+00
  2.56561935e-02  1.99666607e+00  5.98313141e+00  9.79664207e-01
  6.91307926e+00  5.94928551e+00  1.00359659e+01  1.99906576e+00
  5.95398283e+00  5.91228867e+00  5.93811274e+00  1.00323410e+01
```

4.46221232e-03	4.21710908e-02	5.95548534e+00	1.00142407e+00
7.91652441e+00	7.84615517e+00	5.91898394e+00	5.91071081e+00
5.95645046e+00	1.97244465e+00	6.89940071e+00	6.90708351e+00
9.94500256e+00	6.86051702e+00	7.24078119e-02	9.98417854e+00
6.90353346e+00	6.84794140e+00	9.98458290e+00	2.35097110e-02
6.86695480e+00	6.81737757e+00	6.89310026e+00	6.90392017e+00
9.98937130e+00	1.11572094e+01	5.93440390e+00	6.89754725e+00
5.01054859e+00	9.45388317e-01	6.87790203e+00	6.88610601e+00
7.88192987e+00	6.85611153e+00	6.83737755e+00	6.89659071e+00
6.86552095e+00	1.82863176e-02	6.88980722e+00	6.89587355e+00
1.00108776e+01	6.88129044e+00	5.92460537e+00	5.98258209e+00
6.91899920e+00	7.87976789e+00	7.89194155e+00	3.11900079e-02
6.86905718e+00	7.85346746e+00	3.00833988e+00	-4.15235758e-04
6.90706491e+00	5.92943525e+00	7.86385155e+00	7.87621069e+00
2.00755191e+00	6.87599897e+00	9.93729687e+00	2.24255025e-02
5.92842197e+00	5.92233181e+00	6.86661434e+00	9.99626350e+00
4.98435211e+00	5.98090553e+00	1.00006104e+01	3.03033292e-02
1.65925920e-02	6.89433193e+00	5.91381979e+00	6.89453030e+00
5.93762875e+00	6.85958815e+00	6.80548847e-02	8.93385315e+00
7.89240122e+00	1.96637118e+00	6.88693953e+00	6.87173128e+00
5.92502069e+00	5.90226507e+00	-3.55342031e-03	9.46247935e-01
5.91229248e+00	7.84231329e+00	5.97800875e+00	9.95580387e+00
2.99408138e-02	3.00693274e+00	9.52109218e-01	8.91246223e+00
9.97904205e+00	5.98171759e+00	3.82991731e-02	6.85712576e+00
6.89575100e+00	4.98998690e+00	9.47529674e-01	9.89805579e-01
1.97448170e+00	5.94858980e+00	6.91587591e+00	7.90038919e+00
1.02279937e+00	2.99708271e+00	6.84552002e+00	6.85557461e+00
5.90187454e+00	6.86843586e+00	9.60393548e-01	6.89913416e+00
2.01147413e+00	7.84598732e+00	5.98299980e+00	6.88590717e+00
3.07863653e-02	2.47959793e-02	6.85623884e+00	7.87293720e+00
5.94446182e+00	6.87947321e+00	5.96050215e+00	3.02463055e+00
7.88290167e+00	9.94352818e+00	6.89747858e+00	6.89569855e+00
6.88250399e+00	8.90327168e+00	1.00182762e+01	1.00230970e+01
5.91538191e+00	5.93567228e+00	7.89175844e+00	1.96405351e-02
5.96006441e+00	1.00362539e+01	1.97192800e+00	6.90705585e+00
9.37630177e-01	2.99248767e+00	-9.88891721e-03	1.00170603e+01
6.86900902e+00	2.02559972e+00	6.91732264e+00	1.99760020e+00
6.90423775e+00	3.02485418e+00	9.90100384e-01	1.96478021e+00
6.89561033e+00	1.00034676e+01	1.00091591e+01	1.98300803e+00
5.91623163e+00	1.95023024e+00	6.86922932e+00	1.97730529e+00
6.86635399e+00	2.01642776e+00	1.95297039e+00	9.51886535e-01
4.87667024e-02	5.95368433e+00	5.94940186e+00	7.86357689e+00
6.87750483e+00	6.87188435e+00	7.84755850e+00	5.00982761e+00
6.86947584e+00	5.97874737e+00	5.96360874e+00	8.91340733e+00
6.83860540e+00	9.47519660e-01	3.00958252e+00	9.93991852e+00
6.87618351e+00	9.35813189e-01	1.04316175e-02	6.84265852e+00
2.16530263e-02	5.92951918e+00	5.96336985e+00	6.85706139e+00
-1.34388506e-02	7.83395720e+00	7.88623095e+00	4.97960281e+00
9.97750092e+00	5.96322775e+00	5.95789814e+00	7.85743284e+00
6.91094828e+00	6.91387033e+00	5.91386843e+00	8.89794731e+00
6.84834146e+00	6.84227037e+00	6.88604975e+00	1.96547949e+00
9.97493458e+00	8.85695934e+00	6.86755133e+00	2.04754710e+00
5.02956629e+00	9.64888692e-01	5.92472315e+00	5.08925617e-02
7.91369867e+00	5.93171692e+00	1.87716186e-02	9.76285338e-01
7.83481598e+00	6.86354542e+00	5.04481173e+00	5.90095949e+00
6.89438391e+00	5.89277697e+00	9.99577618e+00	6.86740685e+00
6.86141014e+00	6.86849689e+00	9.77182746e-01	6.89059210e+00
6.88490343e+00	9.96202946e+00	3.02848959e+00	5.97539330e+00
6.87796974e+00	6.90671682e+00	4.72405851e-02	6.86705494e+00
7.89965582e+00	7.85040617e+00	1.99587357e+00	9.84652400e-01
5.96558380e+00	5.93412495e+00	6.88342094e+00	7.87806082e+00
6.86345005e+00	6.91587591e+00	6.90727711e+00	7.83938742e+00
9.31478739e-01	5.96328640e+00	5.96096754e+00	1.00124607e+01
3.02696276e+00	5.91052008e+00	6.89017677e+00	3.01598930e+00
7.89372873e+00	6.91260290e+00	6.85501432e+00	5.97480774e+00
7.85639000e+00	4.00627041e+00	5.90347958e+00	6.89149332e+00
6.90424538e+00	6.85750055e+00	-8.30319524e-03	5.93055820e+00
6.89425611e+00	5.88520098e+00	5.94007730e+00	3.04443073e+00
9.96128082e+00	1.00103416e+01	5.90969563e+00	7.88951063e+00
6.87865114e+00	1.95287478e+00	6.88998270e+00	1.98461044e+00
6.89125204e+00	6.86901808e+00	5.93758440e+00	5.93753242e+00
9.54963446e-01	6.89566278e+00	5.92477083e+00	5.98329258e+00
6.91362476e+00	5.92158890e+00	2.50328481e-02	6.88702631e+00
6.89519167e+00	1.70643032e-02	5.93329144e+00	5.90978241e+00
6.87199783e+00	9.69050646e-01	3.04304576e+00	1.00173178e+01
6.89123678e+00	5.04140568e+00	6.87087870e+00	5.94547224e+00
5.28142154e-02	3.04010344e+00	1.00164499e+01	6.89742804e+00
5.92422056e+00	1.00009737e+01	6.91850519e+00	5.92962885e+00
4.01742792e+00	-2.39542127e-03	1.00160370e+01	7.88490725e+00
5.95816612e+00	4.96289349e+00	9.59046602e-01	9.98958206e+00
5.03365755e+00	4.03344917e+00	1.10861868e-01	9.98935580e-01
6.84785700e+00	9.48304534e-01	5.94282484e+00	8.94559383e+00
5.87872076e+00	7.90969801e+00	2.03192616e+00	6.87184525e+00

```
 1.00159330e+01  4.05097902e-02  5.96087170e+00  6.88272429e+00
 6.89031219e+00  5.69947660e-02  5.97421312e+00  6.82717800e+00
 5.91313219e+00  2.00285006e+00  1.18613452e-01  5.95922661e+00
 1.00140381e+01  6.82479906e+00  4.39817607e-02  9.99870491e+00
 9.55253482e-01  6.91678524e+00  6.89782381e+00  6.88220692e+00
 2.99523830e+00  6.88103914e+00  1.96736801e+00  6.82447338e+00
 5.91531038e+00  3.63613069e-02  6.83290577e+00  8.89807701e+00
 1.00098524e+01  6.85484409e+00  5.77729642e-02  9.25702214e-01
 6.83855677e+00  1.00081959e+01  5.92883921e+00  5.02677393e+00
 1.00087681e+01  6.87071896e+00  2.02473140e+00  5.96002579e+00
 5.97480106e+00  6.89531374e+00  9.97940898e-01  6.86686754e+00
 8.87230301e+00  6.89571857e+00  6.90938187e+00  9.98266411e+00
 8.90946293e+00  6.88466787e+00  6.85545397e+00  6.86261511e+00
 2.01277447e+00  9.49829817e-04  5.02198744e+00 -1.61018968e-03
 1.00241077e+00  1.28379762e-02  5.95844412e+00  6.91672611e+00
 2.31114328e-02  2.00025225e+00  2.04649687e+00  7.91549540e+00
 8.91423893e+00  6.86135006e+00  7.88002825e+00  6.89115095e+00
 1.00057554e+01  6.89668751e+00  5.90114546e+00  2.93252170e-02
 9.59403038e-01  3.01390624e+00  5.94222069e+00  3.41925323e-02
 9.40358996e-01  6.89057207e+00  4.00905752e+00  4.04304266e+00
 6.90946627e+00  3.46538723e-02  5.93686724e+00  7.89015341e+00
 6.88453627e+00  6.85802698e+00  6.91786194e+00  3.02290130e+00
 3.02562714e+00  9.37507272e-01  6.87139273e+00  9.97478485e+00
 7.85253716e+00  4.43296134e-02  1.99642408e+00  5.92884302e+00
 7.91600370e+00  6.90297985e+00  6.87282515e+00  9.88479614e+00
 4.95444441e+00  6.88480949e+00  6.86306143e+00  6.89884472e+00
 1.21376216e-02  7.89033747e+00  7.90645361e+00  5.97612715e+00
 6.87530088e+00  4.41521406e-04  9.50621247e-01  3.01525998e+00
 6.88327551e+00  6.91671276e+00  6.85534286e+00  5.93210459e+00
 1.00337477e+01  7.85150146e+00  5.92778349e+00  6.89746666e+00
 5.00348377e+00  5.00386572e+00  9.56771135e-01  6.89169788e+00
 6.86539841e+00  5.54677844e-03  6.89630318e+00  6.88478899e+00
 9.98827839e+00  4.98000813e+00  7.87588835e+00  5.94252539e+00
 5.93327665e+00  1.27878278e-01  9.67820764e-01  1.00068417e+01
 6.91473246e+00  3.02832127e+00  6.85453272e+00  6.86586428e+00
 6.88336849e+00  6.91510820e+00  6.88252497e+00  6.89448118e+00
 5.93608046e+00  9.52081561e-01  6.91849566e+00  9.41318750e-01
 5.92843533e+00  5.93661404e+00  6.85000038e+00  6.86359406e+00
 6.86675119e+00  6.86096096e+00  7.91577291e+00  7.84599209e+00
 5.01349258e+00  5.98235130e+00  1.00086994e+01  2.65166163e-03
 6.91007900e+00  6.85544908e-02  6.88376474e+00  6.91115856e+00
 9.54853177e-01  6.86747932e+00  6.89259672e+00  9.67686772e-01
 9.69657302e-01  6.86959839e+00  7.86330795e+00  9.58760619e-01
 5.89932489e+00  5.95705652e+00  1.00403261e+01  3.47953737e-02
 3.02203703e+00  6.89467716e+00  6.89688540e+00  9.98024273e+00
 6.87042904e+00  4.04732275e+00  9.74070430e-01  3.01021433e+00
 8.91060734e+00  6.87853861e+00  5.95471382e+00  6.99729621e-02
 5.93095446e+00  7.91605806e+00  5.94804907e+00  5.97563505e+00
 3.02800846e+00  6.89676476e+00  7.88585806e+00  6.83333492e+00
 5.95014620e+00  8.89876747e+00  3.02374411e+00  6.88645220e+00
 6.85210657e+00  6.89676809e+00  1.00605793e+01]
```

In [49]:
```
print(ci_upper)
```

```
[ 6.1433196   6.2121677   0.273348   10.25751    10.257981    0.3800756
  0.26216415  7.1411076   8.126455    9.144569    8.135993    3.2745357
  7.1351914   6.1980023   1.2120711   6.221207    8.131544    9.152823
  6.171251    5.277567    1.1974207  10.23734     6.2178745   6.2170115
  6.1186943   8.139489    7.1368732   6.2269993  10.20008     0.27156904
  9.168228   10.229318    1.2180187   8.099614    7.1383367   1.2659565
  6.2258606   7.1153474   8.090636    7.137996   10.245796    5.2465897
  6.220773    7.1344156  10.241717    0.29316404  3.270784   10.270089
  7.139918    6.1839547   6.12481     7.1325364   0.27805462  1.1672903
  5.2769146   6.1846347  10.290328    1.1995322   6.17622     2.212801
  6.165852    8.1076145   6.186882    7.1379085   6.206486    0.22786489
  8.148465    8.095242    8.160887    5.2670674   7.1487403   6.173679
  0.27025154  2.2412615   6.227727    1.2242595   7.157675    6.193881
 10.2805605   2.2436612   6.1985784   6.156884    6.1827083  10.276936
  0.24905756  0.28676644  6.200081    1.2460194   8.161119    8.090751
  6.1635795   6.1553063   6.201046    2.21704     7.1439962   7.151679
 10.189597    7.1051126   0.31700316 10.228773    7.148129    7.092537
 10.229177    0.26810506  7.1115503   7.061973    7.137696    7.1485157
 10.233966   11.401804    6.1789994   7.142143    5.255144    1.1899836
  7.1224976   7.1307015   8.126525    7.100707    7.081973    7.141186
  7.1101165   0.26288167  7.1344028   7.140469   10.255472    7.125886
  6.169201    6.2271776   7.1635947   8.124363    8.136537    0.27578536
  7.1136527   8.0980625   3.2529354   0.24418011  7.1516604   6.174031
  8.108447    8.120806    2.2521474   7.1205945  10.181891    0.26702085
  6.1730175   6.1669273   7.11121    10.240858    5.2289476   6.225501
 10.245205    0.27489868  0.26118794  7.1389275   6.1584153   7.139126
  6.1822243   7.1041837   0.31265023  9.178448    8.136996    2.2109666
  7.131535    7.116327    6.169616    6.1468606   0.24104193  1.1908432
  6.156888    8.086908    6.2226043  10.200398    0.27453616  3.2515283
  1.1967045   9.157057   10.223637    6.226313    0.28289452  7.1017213
```

```
 7.1403465   5.2345824   1.192125    1.2344009   2.219077    6.1931853
 7.1604714   8.144984    1.2673947   3.2416782   7.0901155   7.10017
 6.14647     7.1130314   1.2049888   7.1437297   2.2560697   8.090583
 6.2275953   7.1305027   0.2753817   0.26939133  7.1008344   8.117533
 6.1890574   7.1240687   6.2050977   3.269226    8.127497   10.188123
 7.142074    7.140294    7.1270995   9.147866   10.262871   10.267692
 6.1599774   6.180268    8.1363535   0.26423588  6.20466    10.2808485
 2.2165234   7.1516514   1.1822255   3.2370832   0.23470643 10.261655
 7.1136045   2.2701952   7.161918    2.2421956   7.1488333   3.2694497
 1.2346957   2.2093756   7.140206   10.248062   10.253754    2.2276034
 6.160827    2.1948256   7.113825    2.2219007   7.1109495   2.2610233
 2.1975658   1.1964818   0.29336205  6.19828     6.1939974   8.108172
 7.1221004   7.11648     8.092154    5.254423    7.1140714   6.223343
 6.2082043   9.158002    7.083201    1.192115    3.254178   10.184513
 7.120779    1.1804085   0.25502697  7.087254    0.26624838  6.1741147
 6.2079654   7.101657    0.2311565   8.078552    8.130826    5.2241983
10.2220955   6.2078233   6.2024937   8.102028    7.155544    7.158466
 6.158464    9.142542    7.092937    7.086866    7.1306453   2.210075
10.219529    9.101554    7.112147    2.2921426   5.274162    1.209484
 6.1693187   0.2954879   8.158294    6.1763124   0.26336697  1.2208806
 8.0794115   7.108141    5.2894073   6.145555    7.1389794   6.1373725
10.240371    7.1120024   7.1060057   7.1130924   1.221778    7.1351876
 7.129499   10.206624    3.273085    6.219989    7.1225653   7.1513124
 0.29183593  7.1116505   8.144251    8.095001    2.240469    1.2292477
 6.2101793   6.1787205   7.1280165   8.122656    7.1080456   7.1604714
 7.1518726   8.083982    1.176074    6.207882    6.205563   10.257055
 3.2715583   6.1551156   7.1347723   3.2605848   8.138324    7.1571984
 7.09961     6.2194033   8.100986    4.250866    6.148075    7.136089
 7.148841    7.102096    0.23629215  6.1751537   7.1388516   6.1297965
 6.184673    3.2890263  10.205875   10.254936    6.154291    8.134106
 7.1232467   2.1974702   7.134578    2.2292058   7.1358476   7.1136136
 6.18218     6.182128    1.1995587   7.1402583   6.1693664   6.227888
 7.1582203   6.1661844   0.2696282   7.131622    7.139787    0.26165965
 6.177887    6.154378    7.1165934   1.2136459   3.2876413  10.261912
 7.1358323   5.286001    7.115474    6.190068    0.29740956  3.284699
10.2610445   7.1420236   6.168816   10.245568    7.1631007   6.1742244
 4.2620234   0.24219993 10.260632    8.129502    6.2027617   5.207489
 1.2036419  10.234177    5.278253    4.2780447   0.35545722  1.2435309
 7.0924525   1.1928998   6.1874204   9.190188    6.1233163   8.154293
 2.2765217   7.116441   10.260528    0.28510514  6.205467    7.12732
 7.1349077   0.3015901   6.2188087   7.0717735   6.1577277   2.2474456
 0.3632088   6.203822   10.258633    7.0693946   0.2885771  10.2432995
 1.1998488   7.161381    7.1424193   7.1268024   3.2398338   7.1256347
 2.2119634   7.069069    6.159906    0.28095666  7.0775013   9.142672
10.254447    7.0994396   0.3023683   1.1702975   7.0831523  10.25279
 6.1734347   5.2713695  10.253363    7.1153145   2.269327    6.2046213
 6.2193966   7.1399093   1.2425362   7.111463    9.116898    7.140314
 7.1539774  10.227259    9.1540575   7.1292634   7.1000495   7.1072106
 2.25737     0.24554518  5.266583    0.24298516  1.247006    0.25743333
 6.2030396   7.1613216   0.26770678  2.2448478   2.2910924   8.16009
 9.1588335   7.1059456   8.124623    7.1357465  10.25035     7.141283
 6.145741    0.27392057  1.2039983   3.2585018   6.186816    0.27878788
 1.1849543   7.1351676   4.253653    4.287638    7.154062    0.27924922
 6.181463    8.134748    7.129132    7.1026225   7.1624575   3.2674968
 3.2702227   1.1821026   7.1159883  10.219379    8.097133    0.28892496
 2.2410195   6.1734385   8.160599    7.1475754   7.1174207  10.129391
 5.19904     7.129405    7.107657    7.1434402   0.25673297  8.1349325
 8.151049    6.2207227   7.1198964   0.24503687  1.1952165   3.2598555
 7.127871    7.1613083   7.0999384   6.1767     10.278342    8.096097
 6.172379    7.142062    5.2480793   5.2484612   1.2013664   7.1362934
 7.109994    0.25014213  7.1408987   7.1293845  10.232873    5.2246037
 8.120483    6.187121    6.177872    0.37247363  1.212416   10.251436
 7.159328    3.2729168   7.0991282   7.11046     7.127964    7.1597037
 7.1271205   7.1390767   6.180676    1.1966769   7.163091    1.185914
 6.173031    6.1812096   7.094596    7.1081896   7.1113467   7.1055565
 8.160368    8.090588    5.258088    6.226947   10.253294    0.24724701
 7.1546745   0.31314984  7.1283603   7.155754    1.1994485   7.112075
 7.1371922   1.2122821   1.2142526   7.114194    8.1079035   1.2033559
 6.1439204   6.201652   10.284921    0.27939072  3.2666326   7.1392727
 7.141481   10.224837    7.1150246   4.2919183   1.2186657   3.2548099
 9.155202    7.123134    6.1993093   0.3145683   6.17555     8.160653
 6.1926446   6.2202306   3.272604    7.1413603   8.130453    7.0779305
 6.1947417   9.143362    3.2683396   7.1310477   7.096702    7.1413636
10.305174  ]
```

In [50]:
```
#### z = 90 %
z = 1.6
ci_lower = y_pred - z * dev
ci_upper = y_pred  + z * dev
```

In [52]:
```
for i in range(5):
    print(f"Prediction: {y_pred[i]}, 90% CI: [{ci_lower[i]}, {ci_upper[i]}]")
```

```
Prediction: 6.021021842956543, 90% CI: [5.921186923980713, 6.120856761932373]
Prediction: 6.089869976043701, 90% CI: [5.990035057067871, 6.189704895019531]
Prediction: 0.15105032920837402, 90% CI: [0.05121549218893051, 0.25088515877723694]
Prediction: 10.135212898254395, 90% CI: [10.035378456115723, 10.235047340393066]
Prediction: 10.1356840133667, 90% CI: [10.035849571228027, 10.235518455505371]
```

In [ ]:

Loading [MathJax]/extensions/Safe.js