Yield Prediction and Analysis

About Dataset

This dataset encompasses agricultural data for multiple crops cultivated across various states in India from the year 2011 till 2020. The dataset provides crucial features related to crop yield prediction, including crop types, crop years, cropping seasons, states, areas under cultivation, production quantities, annual rainfall, fertilizer usage, pesticide usage, and calculated yields.

Data Fields

- Crop: The name of the crop cultivated.
- Crop_Year: The year in which the crop was grown.
- Season: The specific cropping season (e.g., Kharif, Rabi, Whole Year).
- State: The Indian state where the crop was cultivated.
- Area: The total land area (in hectares) under cultivation for the specific crop.
- Production: The quantity of crop production (in metric tons).
- Annual_Rainfall: The annual rainfall received in the crop-growing region (in mm).
- Fertilizer: The total amount of fertilizer used for the crop (in kilograms).
- Pesticide: The total amount of pesticide used for the crop (in kilograms).
- Yield: The calculated crop yield (production per unit area).

Importing Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Importing Agriculture Crop Yield Dataset and Initial Observations

```
data = pd.read_csv('crop_yield.csv')
data.head()
```

```
           Crop  Crop_Year      Season         State   Area  Production  \
0      Arecanut       2011  Kharif       Puducherry    60.0          77
1         Bajra       2011  Kharif       Puducherry    11.0          29
2         Bajra       2011  Summer       Puducherry    20.0          51
3        Banana       2011  Kharif       Puducherry   266.0        3263
4  Black pepper       2011  Kharif       Puducherry    11.0           5
```

```
   Annual_Rainfall  Fertilizer  Pesticide      Yield
0        1434.5875    10051.20      19.80   1.280000
1        1434.5875     1842.72       3.63   2.640000
2        1434.5875     3350.40       6.60   2.550000
3        1434.5875    44560.32      87.78  10.903333
4        1434.5875     1842.72       3.63   0.450000
```

```
data.tail()
```

```
           Crop  Crop_Year       Season   State      Area
Production  \
9017  Sugarcane       2018  Winter       Odisha   6778.0       417672

9018       Urad       2018  Autumn       Odisha  13720.0         3583

9019       Urad       2018  Summer       Odisha   4571.0         2336

9020       Urad       2018  Winter       Odisha  39560.0        13123

9021      Wheat       2018  Summer       Odisha    147.0          268


      Annual_Rainfall  Fertilizer  Pesticide      Yield
9017           1635.9   1099391.6    2372.30  57.584545
9018           1635.9   2225384.0    4802.00   0.336667
9019           1635.9    741416.2    1599.85   0.469091
9020           1635.9   6416632.0   13846.00   0.352759
9021           1635.9     23843.4      51.45   1.825000
```

```
data.shape
```

```
(9022, 10)
```

We understand that our data has 9022 rows and 10 features to work with

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9022 entries, 0 to 9021
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Crop             9022 non-null   object
 1   Crop_Year        9022 non-null   int64
 2   Season           9022 non-null   object
 3   State            9022 non-null   object
 4   Area             9022 non-null   float64
 5   Production       9022 non-null   int64
 6   Annual_Rainfall  9022 non-null   float64
```

```
 7   Fertilizer       9022 non-null   float64
 8   Pesticide        9022 non-null   float64
 9   Yield            9022 non-null   float64
dtypes: float64(5), int64(2), object(3)
memory usage: 705.0+ KB
```

This information tells us that, out of the 10 features in our dataset 3 are object type else all are numerical data

```
data.describe(include='all').T
```

```
                  count unique          top  freq               mean  \
Crop               9022     55         Rice   527                NaN
Crop_Year        9022.0    NaN          NaN   NaN        2015.182997
Season             9022      6  Kharif       3769                NaN
State              9022     30    Karnataka   621                NaN
Area             9022.0    NaN          NaN   NaN      159062.678774
Production       9022.0    NaN          NaN   NaN    17051337.986588
Annual_Rainfall  9022.0    NaN          NaN   NaN        1435.353493
Fertilizer       9022.0    NaN          NaN   NaN    25052366.189199
Pesticide        9022.0    NaN          NaN   NaN       53458.906243
Yield            9022.0    NaN          NaN   NaN          82.457035

                               std      min          25%
50%  \
Crop                           NaN      NaN          NaN          NaN

Crop_Year                 2.582179   2011.0       2013.0       2015.0

Season                         NaN      NaN          NaN          NaN

State                          NaN      NaN          NaN          NaN

Area                 630033.776956      0.8      1036.25       7000.0

Production       270796230.070602      0.0       1052.5      11117.5

Annual_Rainfall         785.527676    301.3        956.2       1235.6

Fertilizer        99436190.084488  120.768  163052.9676  1100093.175

Pesticide            212682.696615    0.264     347.5925      2335.53

Yield                   923.014095      0.0     0.683333       1.1255


                     75%           max
Crop                 NaN           NaN
Crop_Year         2017.0        2020.0
Season               NaN           NaN
```

```
State                    NaN           NaN
Area                 56567.5    10216517.0
Production         108095.75  6200900000.0
Annual_Rainfall       1593.9        5649.1
Fertilizer        8878633.35  1754788960.0
Pesticide           18940.05    3780111.29
Yield               2.699125       21105.0
```

```
data.isnull().sum()
```

```
Crop               0
Crop_Year          0
Season             0
State              0
Area               0
Production         0
Annual_Rainfall    0
Fertilizer         0
Pesticide          0
Yield              0
dtype: int64
```

This shows that our data has no null values in it

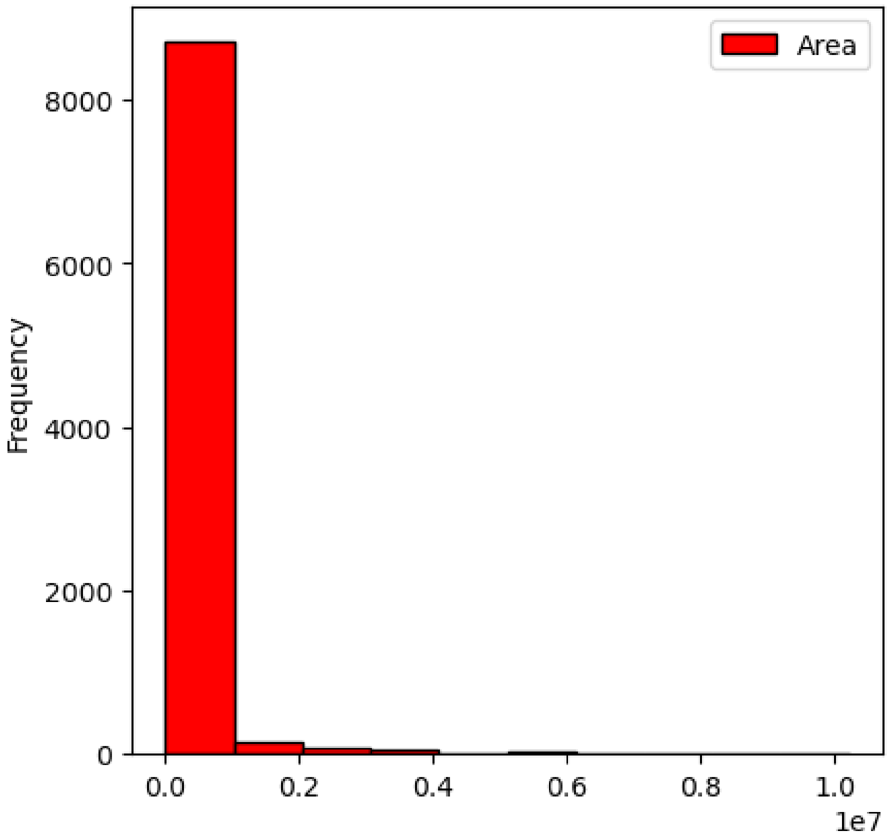
```
data.head()
```

```
           Crop  Crop_Year       Season       State   Area  Production
\
0      Arecanut       2011  Kharif       Puducherry   60.0          77

1         Bajra       2011  Kharif       Puducherry   11.0          29

2         Bajra       2011  Summer       Puducherry   20.0          51

3        Banana       2011  Kharif       Puducherry  266.0        3263

4  Black pepper       2011  Kharif       Puducherry   11.0           5


   Annual_Rainfall  Fertilizer  Pesticide      Yield
0        1434.5875    10051.20      19.80   1.280000
1        1434.5875     1842.72       3.63   2.640000
2        1434.5875     3350.40       6.60   2.550000
3        1434.5875    44560.32      87.78  10.903333
4        1434.5875     1842.72       3.63   0.450000
```

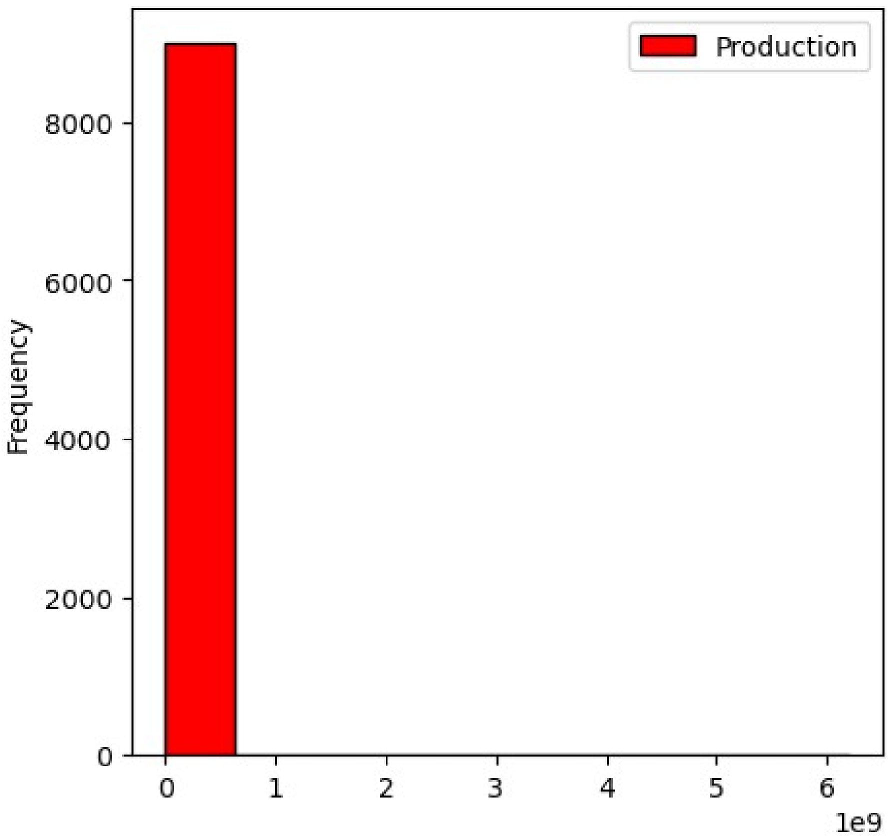
Exploratory Data Analysis

```
data['Area'].plot(kind='hist',figsize=(5,5),color='red',edgecolor='bla
ck')
```

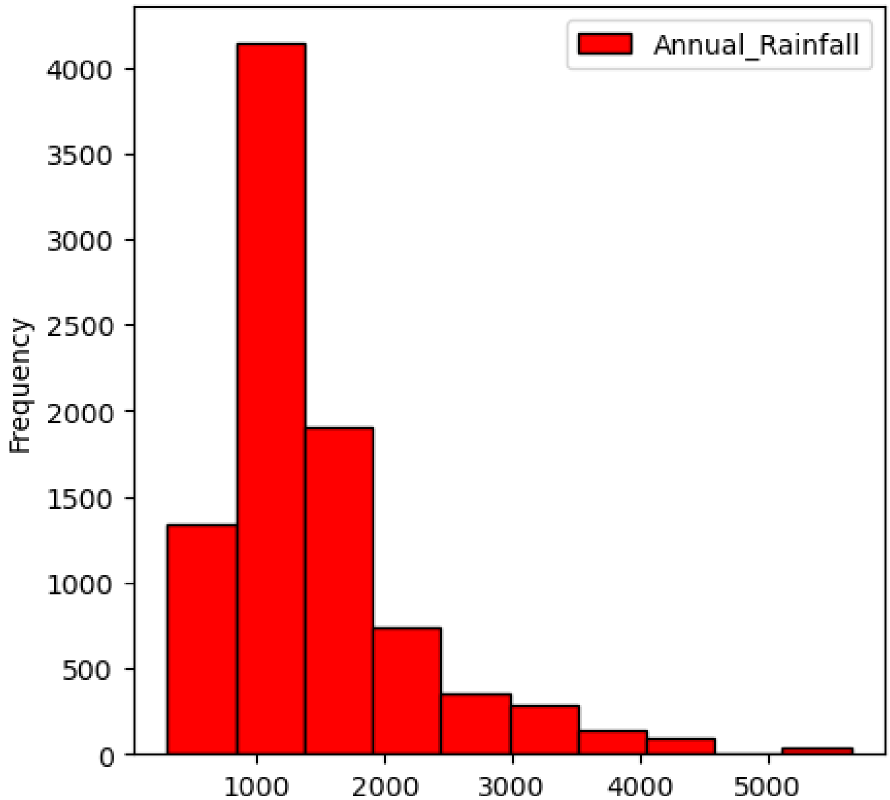
```
plt.legend()
plt.show()
```



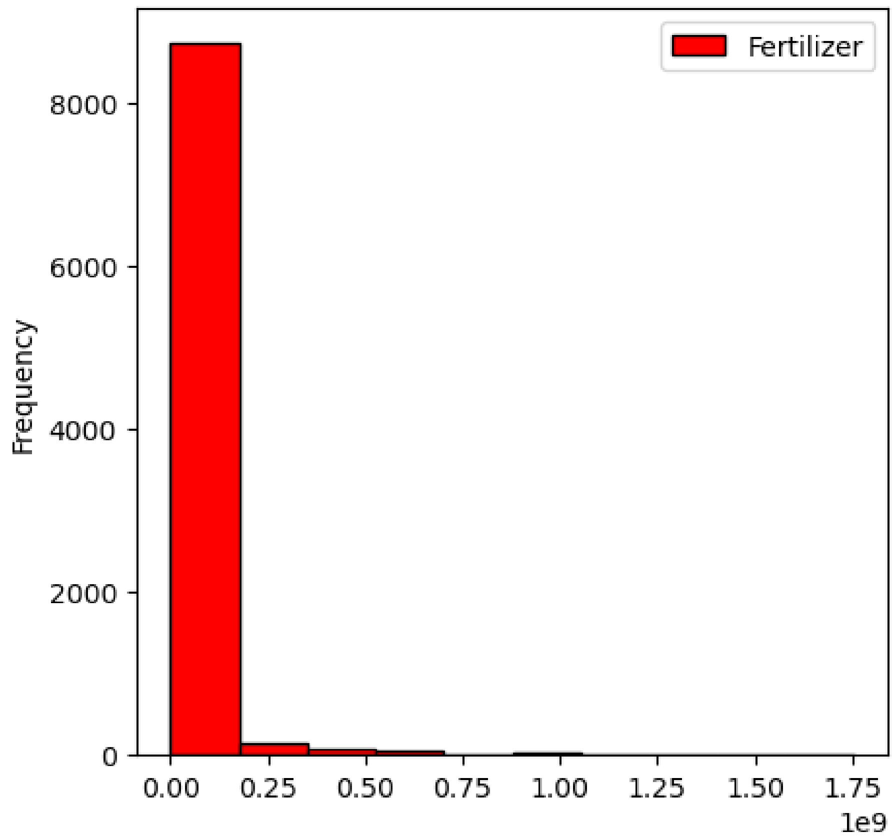

```
data['Production'].plot(kind='hist',figsize=(5,5),color='red',edgecolo
r='black')
plt.legend()
plt.show()
```



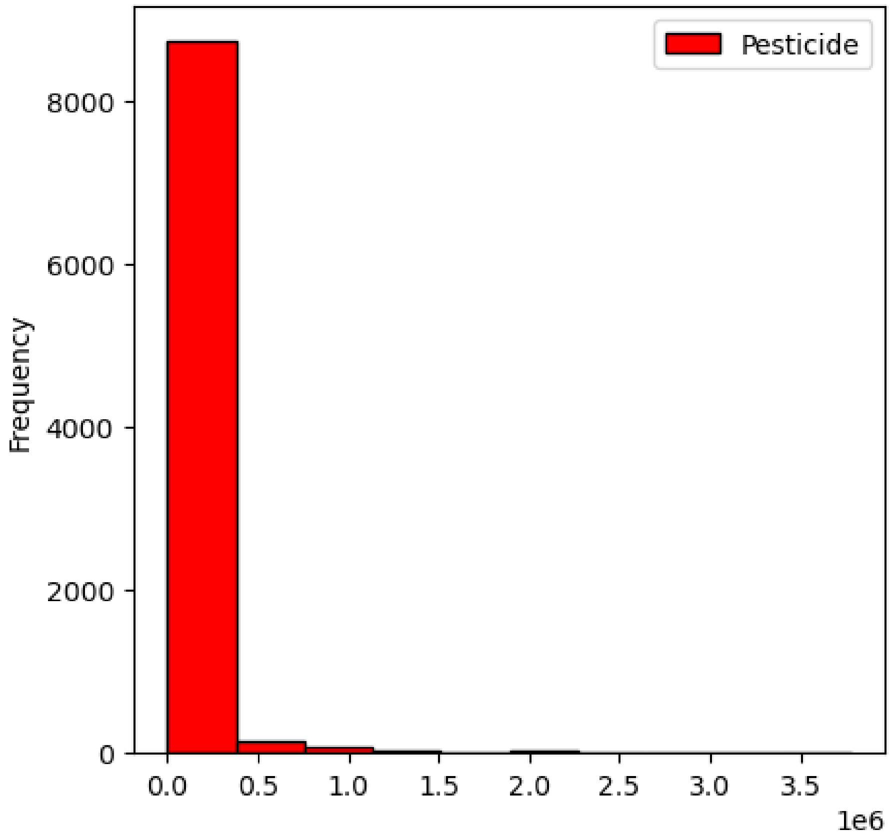

```
data['Annual_Rainfall'].plot(kind='hist',figsize=(5,5),color='red',edg
ecolor='black')
plt.legend()
plt.show()
```




```
data['Fertilizer'].plot(kind='hist',figsize=(5,5),color='red',edgecolo
r='black')
plt.legend()
plt.show()
```




```
data['Pesticide'].plot(kind='hist',figsize=(5,5),color='red',edgecolor
='black')
plt.legend()
plt.show()
```

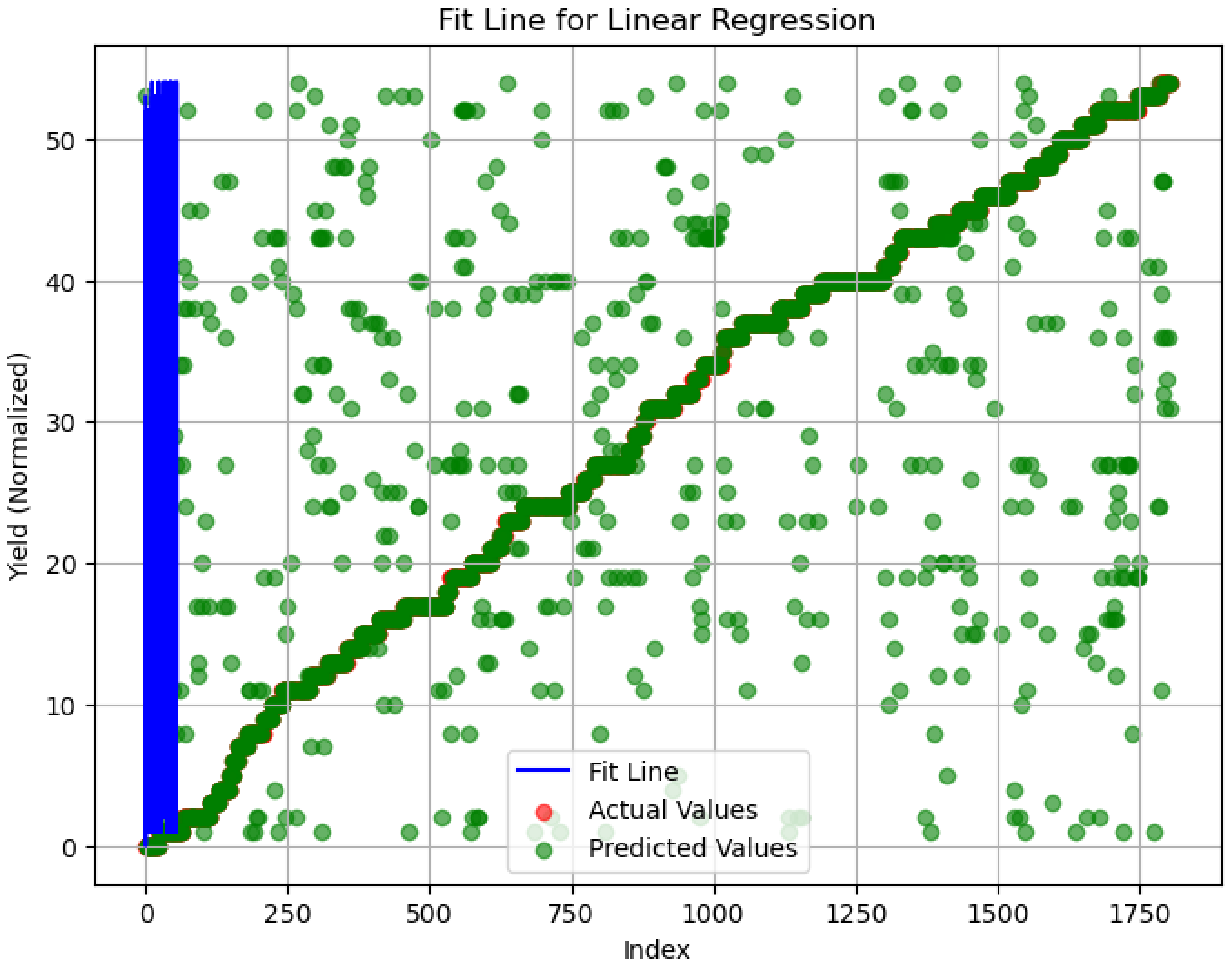



```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.metrics import mean_squared_error, r2_score

# List of categorical and numerical columns
categorical_columns = ['Crop', 'Season', 'State']
numerical_columns = ['Crop_Year', 'Area', 'Annual_Rainfall',
'Fertilizer', 'Pesticide']

# Encoding categorical variables using one-hot encoding
data_encoded = pd.get_dummies(data, columns=categorical_columns,
drop_first=True)

# Standardizing numerical features
scaler = StandardScaler()
data_encoded[numerical_columns] =
scaler.fit_transform(data_encoded[numerical_columns])

# Normalizing the target variable (Yield)
target_scaler = MinMaxScaler()
data_encoded['Yield'] =
target_scaler.fit_transform(data_encoded[['Yield']])
```

```python
# Splitting features and target variable
X = data_encoded.drop(columns=['Yield'])  # Features
y = data_encoded['Yield']  # Target

# Train-test splitting
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Trainning the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Making predictions
y_pred = model.predict(X_test)

# Model eEvaluation
r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

# Print evaluation metrics
print(f"R² Score: {r2:.4f}")
print(f"Mean Squared Error (MSE): {mse:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")
```

```
R² Score: 0.8541
Mean Squared Error (MSE): 0.0004
Root Mean Squared Error (RMSE): 0.0191
```

```python
sorted_indices = np.argsort(y_test)
y_test_sorted = y_test.iloc[sorted_indices]
y_pred_sorted = y_pred[sorted_indices]

# Plotting the Actual vs. Predicted with Fit Line
plt.figure(figsize=(8, 6))
plt.plot(y_test_sorted, y_pred_sorted, color='blue', label='Fit Line')
plt.scatter(range(len(y_test_sorted)), y_test_sorted, color='red',
label='Actual Values', alpha=0.6)
plt.scatter(range(len(y_pred_sorted)), y_pred_sorted, color='green',
label='Predicted Values', alpha=0.6)
plt.xlabel("Index")
plt.ylabel("Yield (Normalized)")
plt.title("Fit Line for Linear Regression")
plt.legend()
plt.grid(True)
plt.show()
```



Fit Line for Linear Regression
Yield (Normalized)
Index
Fit Line
Actual Values
Predicted Values
0
10
20
30
40
50
0
250
500
750
1000
1250
1500
1750