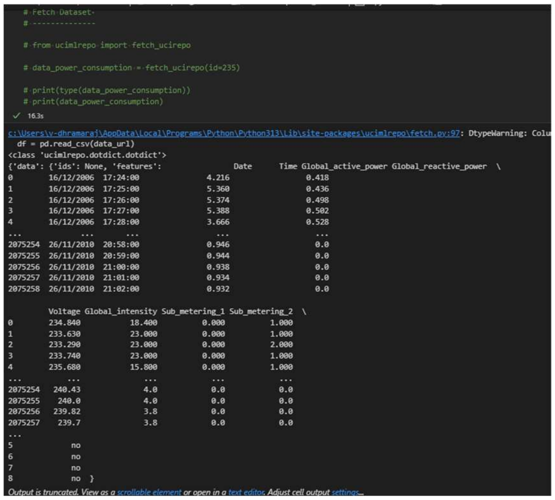
Fetch Dataset & Load Features



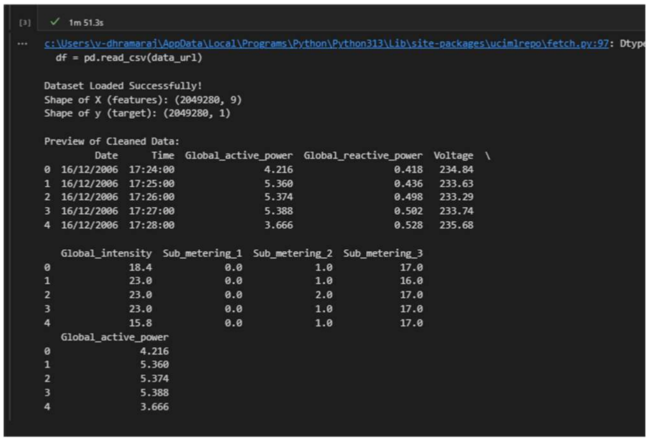
```
# Fetch Dataset
# -------------

# from ucimlrepo import fetch_ucirepo

# data_power_consumption = fetch_ucirepo(id=235)

# print(type(data_power_consumption))
# print(data_power_consumption)
```

✓ 16.3s

```
c:\Users\v-dhramaraj\AppData\Local\Programs\Python\Python313\Lib\site-packages\ucimlrepo\fetch.py:97: DtypeWarning: Colu
  df = pd.read_csv(data_url)
<class 'ucimlrepo.dotdict.dotdict'>
{'data': {'ids': None, 'features':               Date      Time Global_active_power Global_reactive_power  \
0        16/12/2006  17:24:00               4.216                 0.418
1        16/12/2006  17:25:00               5.360                 0.436
2        16/12/2006  17:26:00               5.374                 0.498
3        16/12/2006  17:27:00               5.388                 0.502
4        16/12/2006  17:28:00               3.666                 0.528
...             ...       ...                 ...                   ...
2075254  26/11/2010  20:58:00               0.946                   0.0
2075255  26/11/2010  20:59:00               0.944                   0.0
2075256  26/11/2010  21:00:00               0.938                   0.0
2075257  26/11/2010  21:01:00               0.934                   0.0
2075258  26/11/2010  21:02:00               0.932                   0.0

        Voltage Global_intensity Sub_metering_1 Sub_metering_2  \
0       234.840           18.400          0.000          1.000
1       233.630           23.000          0.000          1.000
2       233.290           23.000          0.000          2.000
3       233.740           23.000          0.000          1.000
4       235.680           15.800          0.000          1.000
...         ...              ...            ...            ...
2075254  240.43              4.0            0.0            0.0
2075255   240.0              4.0            0.0            0.0
2075256  239.82              3.8            0.0            0.0
2075257   239.7              3.8            0.0            0.0
...
5          no
6          no
7          no
8          no  }
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

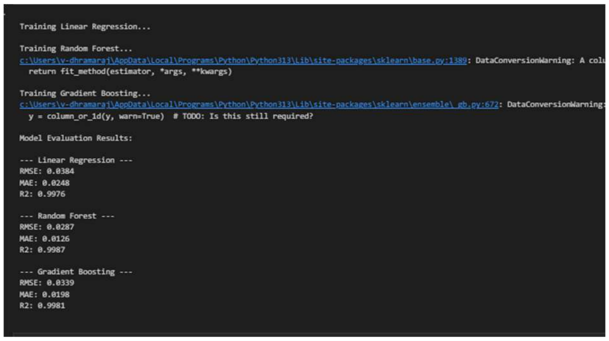
Data after EDA



[3] ✓ 1m 51.3s

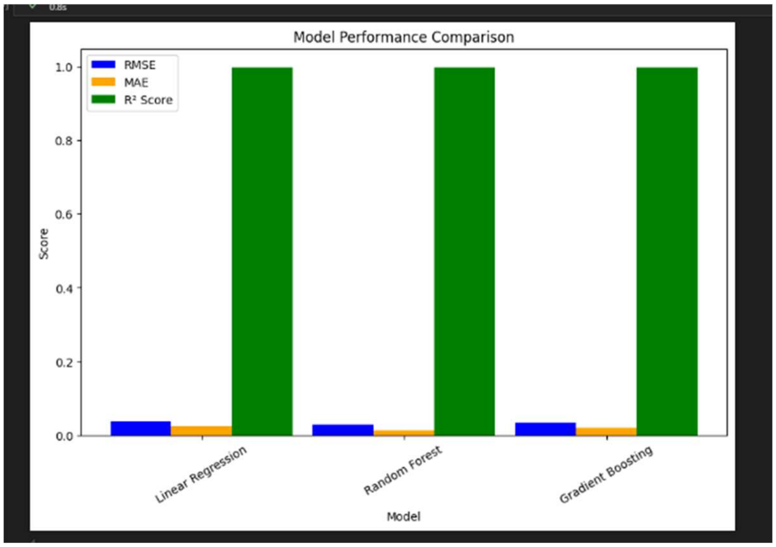
```
... c:\Users\v-dhramaraj\AppData\Local\Programs\Python\Python313\Lib\site-packages\ucimlrepo\fetch.py:97: Dtype
      df = pd.read_csv(data_url)

    Dataset Loaded Successfully!
    Shape of X (features): (2049280, 9)
    Shape of y (target): (2049280, 1)

    Preview of Cleaned Data:
             Date      Time  Global_active_power  Global_reactive_power  Voltage  \
    0  16/12/2006  17:24:00                4.216                  0.418   234.84
    1  16/12/2006  17:25:00                5.360                  0.436   233.63
    2  16/12/2006  17:26:00                5.374                  0.498   233.29
    3  16/12/2006  17:27:00                5.388                  0.502   233.74
    4  16/12/2006  17:28:00                3.666                  0.528   235.68

       Global_intensity  Sub_metering_1  Sub_metering_2  Sub_metering_3
    0              18.4             0.0             1.0            17.0
    1              23.0             0.0             1.0            16.0
    2              23.0             0.0             2.0            17.0
    3              23.0             0.0             1.0            17.0
    4              15.8             0.0             1.0            17.0
       Global_active_power
    0                4.216
    1                5.360
    2                5.374
    3                5.388
    4                3.666
```

Model Evaluation Results:

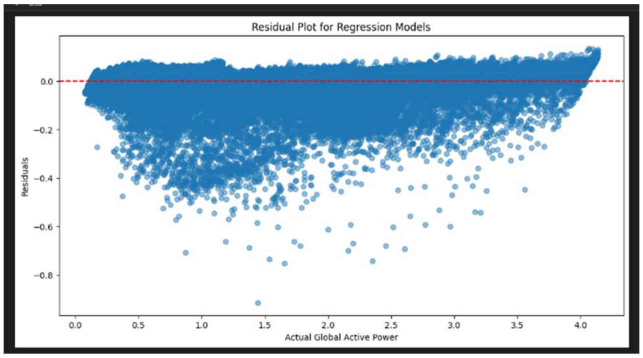


```
Training Linear Regression...

Training Random Forest...
c:\Users\v-dhramaraj\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\base.py:1389: DataConversionWarning: A colu
  return fit_method(estimator, *args, **kwargs)

Training Gradient Boosting...
c:\Users\v-dhramaraj\AppData\Local\Programs\Python\Python313\Lib\site-packages\sklearn\ensemble\_gb.py:672: DataConversionWarning:
  y = column_or_1d(y, warn=True)  # TODO: Is this still required?

Model Evaluation Results:

--- Linear Regression ---
RMSE: 0.0384
MAE: 0.0248
R2: 0.9976

--- Random Forest ---
RMSE: 0.0287
MAE: 0.0126
R2: 0.9987

--- Gradient Boosting ---
RMSE: 0.0339
MAE: 0.0198
R2: 0.9981
```

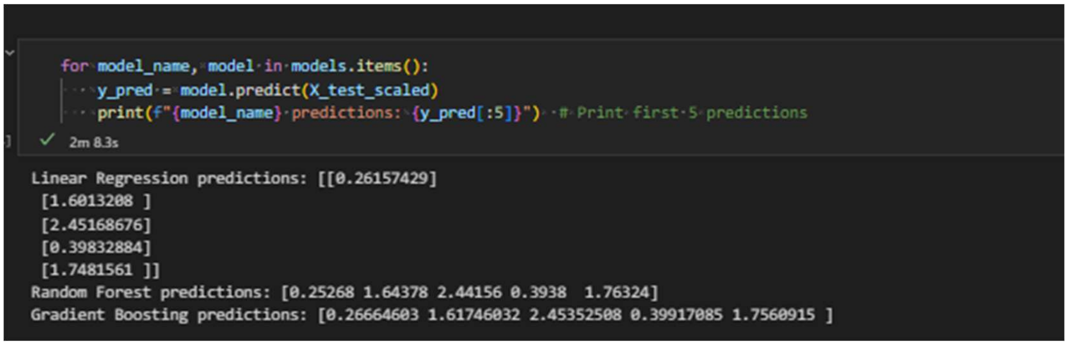
Compare model performance and select the best-performing model.



Residual Plot



Model predictions

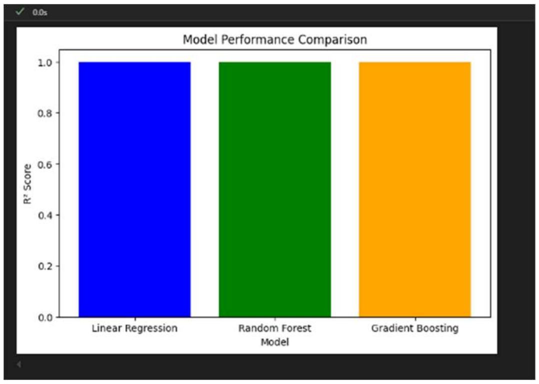


```
for model_name, model in models.items():
    y_pred = model.predict(X_test_scaled)
    print(f"{model_name} predictions: {y_pred[:5]}")  # Print first 5 predictions
```

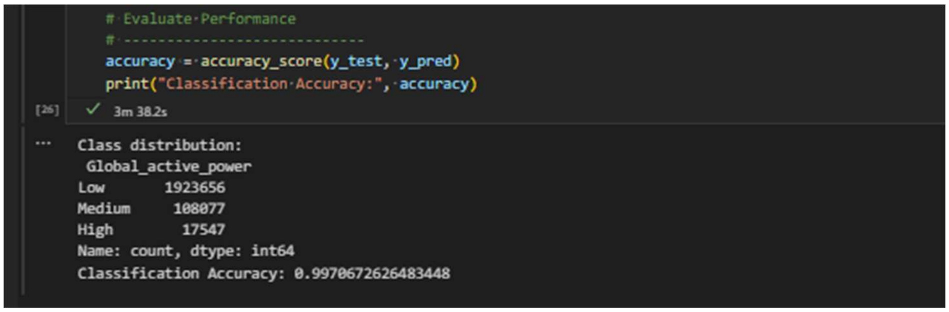
✓ 2m 8.3s

```
Linear Regression predictions: [[0.26157429]
 [1.6013208 ]
 [2.45168676]
 [0.39832884]
 [1.7481561 ]]
Random Forest predictions: [0.25268 1.64378 2.44156 0.3938  1.76324]
Gradient Boosting predictions: [0.26664603 1.61746032 2.45352508 0.39917085 1.7560915 ]
```

R^2 Score Bar Plot



Classification Model



```
# Evaluate Performance
# ----------------------------
accuracy = accuracy_score(y_test, y_pred)
print("Classification Accuracy:", accuracy)
```

```
Class distribution:
 Global_active_power
Low       1923656
Medium     108077
High        17547
Name: count, dtype: int64
Classification Accuracy: 0.9970672626483448
```