Trainig Code

The model being used is 'Linear Regression'. Basically, multivariate regression by training on multiple independent variables using linear regression. The DC_POWER was predicted using the features: AMBIENT_TEMPERATURE, MODULE_TEMPERATURE, IRRADIATION.
The data was split into one-third testing dataset and the rest training dataset. After predicting using the model, the R2 score showed up to be: 0.9921417157083059, and bias and variance as 122386.029 and 439.856 respectively.

In [83]:

```
import pandas as pd
import numpy as np
```

In [84]:

```
df=pd.read_csv('Dataset/P1.csv')
df=df.drop('Unnamed: 0',axis=1)
```

In [85]:

```
df
```

Out[85]:

	DC_POWER	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION
0	0.0	23.128673	20.464305	0.0
1	0.0	23.032562	20.341429	0.0
2	0.0	22.967493	20.269493	0.0
3	0.0	22.810594	20.198918	0.0
4	0.0	22.611436	20.085866	0.0
...	...	...	...	...
3152	0.0	23.670292	21.691071	0.0
3153	0.0	23.795434	22.067778	0.0
3154	0.0	23.727901	21.662972	0.0
3155	0.0	23.497284	21.051402	0.0
3156	0.0	23.244698	20.774560	0.0

3157 rows × 4 columns

Setting up target variable and independent variable

In [86]:

```
X=df[['AMBIENT_TEMPERATURE','MODULE_TEMPERATURE', 'IRRADIATION']]
y=df['DC_POWER']
```

In [87]:

```
from sklearn.model_selection import train_test_split
```

Splitting data into training, testing data

In [88]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3
3, random_state=42)
```

In [89]:

```
from sklearn.linear_model import LinearRegression
```

Fitting data using linear regression model

In [90]:

```
reg=LinearRegression()
reg.fit(X_train, y_train)
```

Out[90]:

```
LinearRegression()
```

Predicting using the model

In [91]:

```
y_pred=reg.predict(X_test)
y_pred
```

Out[91]:

```
array([  62.32466286,   45.22896531,   52.58443125, ...,   83.11129224,
        118.26822688, 4796.07909294])
```

In [97]:

```
from sklearn.metrics import r2_score, confusion_matrix, accuracy_score
from sklearn.metrics import mean_squared_error
from mlxtend.evaluate import bias_variance_decomp
```

Calculating mean square error, bias and variance

In [99]:

```
mse, bias, var = bias_variance_decomp(reg, X_train.values, y_train.valu
es, X_test.values, y_test.values, loss='mse', num_rounds=200, random_se
ed=1)
print('MSE: %.3f' % mse)
print('Bias: %.3f' % bias)
print('Variance: %.3f' % var)
```

```
MSE: 122825.884
Bias: 122386.029
Variance: 439.856
```

Calculating R2 score and root mean square error

In [93]:

In [93]:

```
score=r2_score(y_test,y_pred)
print('r2 socre is ',score)
print('root_mean_squared error of is==',np.sqrt(mean_squared_error(y_tes
t,y_pred)))
```

```
r2 socre is  0.9921417157083059
mean_sqrd_error is== 122378.64598680226
root_mean_squared error of is== 349.82659416745645
```

In [ ]:

```

```