In [1]:
```
import pandas as pd
dataset=pd.read_csv("crop_yield_data.csv")
dataset
```

Out[1]:

	Rainfall_mm	Temperature_Celsius	Fertilizer_Used	Irrigation_Used	Days_to_Harvest	Yield_tons_per_hectare
0	897.077239	27.676966	False	True	122	6.555816
1	992.673282	18.026142	True	True	140	8.527341
2	147.998025	29.794042	False	False	106	1.127443
3	986.866331	16.644190	False	True	146	6.517573
4	730.379174	31.620687	True	True	110	7.248251
...	...	...	...	...	...	...
999984	302.805345	27.987428	False	False	76	1.347586
999985	932.991383	39.661039	True	False	93	7.311594
999986	867.362046	24.370042	True	False	108	5.763182
999987	492.812857	33.045505	False	False	102	2.070159
999988	180.936180	27.298847	True	False	76	2.937243

999989 rows × 6 columns

In [2]:
```
dataset.isnull().sum()
```

Out[2]:
```
Rainfall_mm               0
Temperature_Celsius       0
Fertilizer_Used           0
Irrigation_Used           0
Days_to_Harvest           0
Yield_tons_per_hectare    0
dtype: int64
```

In [3]:
```
dataset.columns
```

Out[3]:
```
Index(['Rainfall_mm', 'Temperature_Celsius', 'Fertilizer_Used',
       'Irrigation_Used', 'Days_to_Harvest', 'Yield_tons_per_hectare'],
      dtype='object')
```

In [4]:
```
independent=dataset[['Rainfall_mm', 'Temperature_Celsius', 'Fertilizer_Used','Irrigation_Used', 'Days_to_Harvest']]
independent
```

Out[4]:

	Rainfall_mm	Temperature_Celsius	Fertilizer_Used	Irrigation_Used	Days_to_Harvest
0	897.077239	27.676966	False	True	122
1	992.673282	18.026142	True	True	140
2	147.998025	29.794042	False	False	106
3	986.866331	16.644190	False	True	146
4	730.379174	31.620687	True	True	110
...	...	...	...	...	...
999984	302.805345	27.987428	False	False	76
999985	932.991383	39.661039	True	False	93
999986	867.362046	24.370042	True	False	108
999987	492.812857	33.045505	False	False	102
999988	180.936180	27.298847	True	False	76

999989 rows × 5 columns

In [5]:
```
dependent=dataset[["Yield_tons_per_hectare"]]
dependent
```

Out[5]:

	Yield_tons_per_hectare
0	6.555816
1	8.527341
2	1.127443
3	6.517573
4	7.248251
...	...
999984	1.347586
999985	7.311594
999986	5.763182
999987	2.070159
999988	2.937243

999989 rows × 1 columns

In [6]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(independent, dependent, test_size = 1/3, random_state = 0)
```

In [7]:
```
#model creation phase and linearregression library
from sklearn.linear_model import LinearRegression
#Linearregression function assign an regressor
regressor=LinearRegression()
#fit is an train dataset model
regressor.fit(X_train,y_train)
#weight linear regression
weight=regressor.coef_
#weight result
weight
```

Out[7]:
```
array([[4.99625543e-03, 1.99254708e-02, 1.49974983e+00, 1.20101593e+00,
        4.85778718e-05]])
```

In [8]:
```
#bias or initial value or minimum value
bias=regressor.intercept_
#bias or initial value result
bias
```

Out[8]:
```
array([-0.00084444])
```

In [9]:
```
y_pred=regressor.predict(X_test)
```

In [10]:
```
#R2 value or better model creation and r2 library
from sklearn.metrics import r2_score
r_score=r2_score(y_test , y_pred)
#r2 result
r_score
```

Out[10]:
```
0.9127686040639841
```

In [11]:
```
import pickle
filename="finalized_model_mul_linear.sav"
pickle.dump(regressor,open(filename,'wb'))
```

In [12]:
```
loaded_model=pickle.load(open("finalized_model_mul_linear.sav",'rb'))
result=loaded_model.predict([[897.077239, 27.676966, 0, 1, 122]])
result
```

```
C:\Anaconda3\envs\dineshML\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[12]:

```
array([[6.2396016]])
```

In [ ]:

```
```