

Machine Learning

Simple Linear Regression

1. Supervised Learning

- Work under supervision
- teacher
- Prediction
- Outcome

```
In [ ]: # To install
        # pip install scikit-learn
```

How scikit-learn works?

Data > raw data and final product is divided into x & y Respectively >
Model > 1. Create model , 2. Model Learns > Prediction

Simple Linear Regression

1. Relationship between two variables
2. how much relationship exist and predict on the basis of this relationship
3. Equation $y = a + bx$

a = constant/ intercept b = Function/ slop of x

Linear = straight line in bidimensional plot

X = Independent Variable Y = Dependent Variable

Step1- Import library and Data set

About Daraset: Dataset is downloaded form FOU website, and it ranges from 1961 to 2021. Dataset based on furtilizers used to agriculture use in Pakistan. It tells us in which year how much we have used specific type of furtilizer. Now, we are using this dataset to predict its use in future so that we could make policy regarding their use for furtilizers and may be use this prediction model to meet our future needs to overcome its shortage. Also they have impact on our environment so we could also use this data to make strong policies to keep our environment clean. Here we are using Simple Linear regression machine learning model for prediction.

```
In [ ]: import pandas as pd
        import matplotlib.pyplot as plt
```

```
df = pd.read_csv('fertilizer_used.csv')
df.head()
```

```
Out[ ]:
```

	DomainCode	Domain	AreaCode	Area	ElementCode	Element	ItemCode	Item	Year
0	RFN	Fertilizers by Nutrient	165	Pakistan	5157	Agricultural Use	3102	Nutrient nitrogen N (total)	
1	RFN	Fertilizers by Nutrient	165	Pakistan	5157	Agricultural Use	3103	Nutrient phosphate P2O5 (total)	
2	RFN	Fertilizers by Nutrient	165	Pakistan	5157	Agricultural Use	3103	Nutrient phosphate P2O5 (total)	
3	RFN	Fertilizers by Nutrient	165	Pakistan	5157	Agricultural Use	3102	Nutrient nitrogen N (total)	
4	RFN	Fertilizers by Nutrient	165	Pakistan	5157	Agricultural Use	3102	Nutrient nitrogen N (total)	

```
In [ ]: # Check Null values
df.isnull().sum()
```

```
Out[ ]:
```

DomainCode	0
Domain	0
AreaCode	0
Area	0
ElementCode	0
Element	0
ItemCode	0
Item	0
YearCode	0
Year	0
Unit	0
Value	0
Flag	0
FlagDescription	0

dtype: int64

So we don't have any null value so now we don't need to perform data wrangling.

Step-2 Splitting Dataset into Training data and testing data

```
In [ ]: X = df[['Year']]
y = df['Value']
```

```
In [ ]: X.head()
```

```
Out[ ]: 0    1961
        1    1961
        2    1962
        3    1962
        4    1963
        Name: Year, dtype: int64
```

```
In [ ]: y.head()
```

```
Out[ ]: 0    41659
        1     500
        2     210
        3    41160
        4    67620
        Name: Value, dtype: int64
```

```
In [ ]: # import library and split data
        from sklearn.model_selection import train_test_split
        X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=1/5,random_state=0)
```

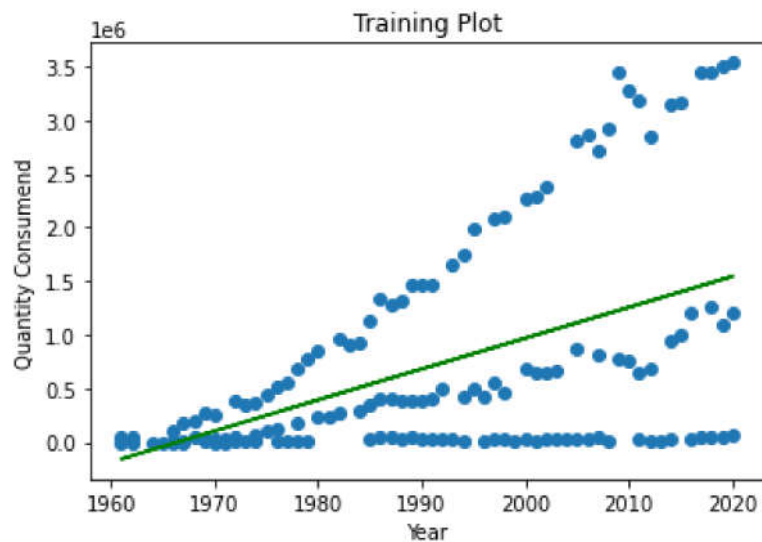
Step-3 Fit linear Regression Model

```
In [ ]: from sklearn.linear_model import LinearRegression
        model = LinearRegression()
        model=model.fit(X_train,y_train)
        model
```

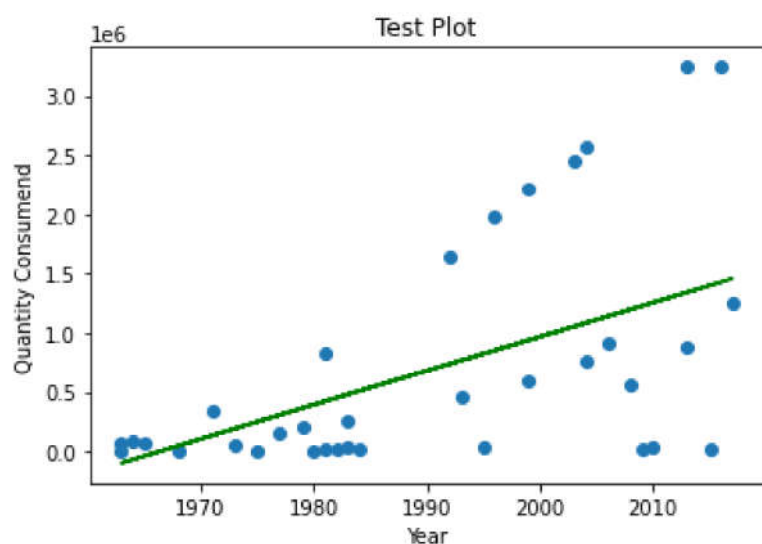
```
Out[ ]: ▾ LinearRegression
        LinearRegression()
```

Step-4 Plotting

```
In [ ]: # Training Plot
        plt.scatter(X_train,y_train)
        plt.plot(X_train,model.predict(X_train),color= 'green')
        plt.xlabel('Year')
        plt.ylabel('Quantity Consumend')
        plt.title('Training Plot')
        plt.show()
```



```
In [ ]: # Test Plot
plt.scatter(X_test,y_test)
plt.plot(X_test,model.predict(X_test),color= 'green')
plt.xlabel('Year')
plt.ylabel('Quantity Consumend')
plt.title('Test Plot')
plt.show()
```



Step-5 Testing or Evaluating your model

```
In [ ]: # Model Fitness
print('Score for Training data ',model.score(X_train,y_train))
print('Score for Testing data ',model.score(X_test,y_test))
```

Score for Training data 0.24210566159460145
Score for Testing data 0.30951958946214964

Step-6 Prediction of unknown Values

```
In [ ]: model.predict([[2021]])
```

```
c:\Users\Administrator\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

```
Out[ ]: array([1572772.73099684])
```

```
In [ ]: model.predict([[2021],[2022]])
```

```
c:\Users\Administrator\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

```
Out[ ]: array([1572772.73099684, 1601582.31015392])
```

```
In [ ]: x = ([2021],[2022],[2025])
        model.predict(x)
```

```
c:\Users\Administrator\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

```
Out[ ]: array([1572772.73099684, 1601582.31015392, 1688011.04762513])
```