Fish Weight Prediction

With a dataset of fish species, with some of it characteristic like it vertical, diagonal, length, height, and width. We will try to predict the weight of the fish based on their characteristic. We will use Linear Regression Method to see whether the weight of the fish related to their characteristic.

- Species: Species name of fish
- Weight: Weight of fish in gram
- Length1: Vertical length in cm
- Length2: Diagonal length in cm
- Length3: Cross length in cm
- Height: Height in cm
- Width: Diagonal width in cm

```
# Step 1 : import library
import pandas as pd
```

```
# Step 2 : import data
fish = pd.read_csv('https://github.com/ybifoundation/Dataset/raw/main/Fish.csv')
```

```
fish.head()
```

	Category	Species	Weight	Height	Width	Length1	Length2	Length3
0	1	Bream	242.0	11.5200	4.0200	23.2	25.4	30.0
1	1	Bream	290.0	12.4800	4.3056	24.0	26.3	31.2
2	1	Bream	340.0	12.3778	4.6961	23.9	26.5	31.1
3	1	Bream	363.0	12.7300	4.4555	26.3	29.0	33.5
4	1	Bream	430.0	12.4440	5.1340	26.5	29.0	34.0

Next steps: Generate code with `fish` View recommended plots

```
fish.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159 entries, 0 to 158
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Category  159 non-null    int64  
 1   Species   159 non-null    object 
 2   Weight    159 non-null    float64
 3   Height    159 non-null    float64
 4   Width     159 non-null    float64
 5   Length1   159 non-null    float64
 6   Length2   159 non-null    float64
 7   Length3   159 non-null    float64
dtypes: float64(6), int64(1), object(1)
memory usage: 10.1+ KB
```

```
fish.describe()
```

	Category	Weight	Height	Width	Length1	Length2	Lengt
count	159.000000	159.000000	159.000000	159.000000	159.000000	159.000000	159.0000
mean	3.264151	398.326415	8.970994	4.417486	26.247170	28.415723	31.2270
std	1.704249	357.978317	4.286208	1.685804	9.996441	10.716328	11.6102
min	1.000000	0.000000	1.728400	1.047600	7.500000	8.400000	8.8000
25%	2.000000	120.000000	5.944800	3.385650	19.050000	21.000000	23.1500
50%	3.000000	273.000000	7.786000	4.248500	25.200000	27.300000	29.4000
75%	4.500000	650.000000	12.365900	5.584500	32.700000	35.500000	39.6500
max	7.000000	1650.000000	18.957000	8.142000	59.000000	63.400000	68.0000

```
# Step 3 : define target (y) and features (X)
```

```
fish.columns
```

```
Index(['Category', 'Species', 'Weight', 'Height', 'Width', 'Length1',
       'Length2', 'Length3'],
      dtype='object')
```

```
y = fish['Weight']
```

```
X = fish[['Category','Height', 'Width', 'Length1',
       'Length2', 'Length3']]
```

```
# Step 4 : train test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.7, random_state=2529)
```

```
# check shape of train and test sample
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((111, 6), (48, 6), (111,), (48,))
```

```
# Step 5 : select model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```
# Step 6 : train or fit model
model.fit(X_train,y_train)
```

```
LinearRegression
LinearRegression()
```

```
model.intercept_
```

```
-684.4235918478537
```

```
model.coef_
```

```
array([ 35.19634977,  52.19372157, -37.13869125,  11.2218449 ,
        78.11233002, -59.11783139])
```

```
# Step 7 : predict model
y_pred = model.predict(X_test)
```

```
y_pred
```

```
array([ 475.93351307,  525.81910195,   77.63275849,  881.10235121,
        160.9685664 ,  255.94371856,  361.87029932,  358.87068094,
        499.83411068, -150.07834151, -115.91810869,  428.65470115,
        114.67533404,  812.51385122,  586.5071178 ,  273.38510858,
        579.63900729,  225.18126845,  639.26068037,   85.00820599,
        136.92159041,  -87.7778087 ,  629.97231046,  732.63097812,
        859.8720695 , -166.76928607,  342.04209934,  722.92198147,
        321.44827179,  787.98248357,  486.93194673,  541.89982795,
        376.74813045,  624.81211202, -170.11945033,  917.76513801,
        792.26439518,  -21.15655005,  300.24921659,  914.07325473,
        621.05636286,  934.17373986,  676.85479574,  653.92304403,
        615.51226767,  336.61090622,  505.75519147,  -33.53283763])
```

```
# Step 8 : model accuracy
from sklearn.metrics import mean_absolute_error, r2_score
```

```
mean_absolute_error(y_test,y_pred)
```

```
99.58910366731813
```

```
r2_score(y_test,y_pred)
```

```
0.83982461599445
```