

```
#steps for regression model with csv
```

```
#import library
import pandas as pd
```

```
import numpy as np
```

```
#import csv as dataframe
df=pd.read_csv('/content/New_Fish_dataset.zip')
```

```
df
```

	Species	Weight	Length1	Length2	Length3	Height	Width
0	Bream	242.0	23.2	25.4	30.0	11.5200	4.0200
1	Bream	290.0	24.0	26.3	31.2	12.4800	4.3056
2	Bream	340.0	23.9	26.5	31.1	12.3778	4.6961
3	Bream	363.0	26.3	29.0	33.5	12.7300	4.4555
4	Bream	430.0	26.5	29.0	34.0	12.4440	5.1340
...
154	Smelt	12.2	11.5	12.2	13.4	2.0904	1.3936
155	Smelt	13.4	11.7	12.4	13.5	2.4300	1.2690
156	Smelt	12.2	12.1	13.0	13.8	2.2770	1.2558
157	Smelt	19.7	13.2	14.3	15.2	2.8728	2.0672
158	Smelt	19.9	13.8	15.0	16.2	2.9322	1.8792

159 rows × 7 columns

```
#get first five rows of dataset
df.head()
```

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

```
df.head(5)
```

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



```
#get info about dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159 entries, 0 to 158
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Species     159 non-null    object
1   Weight      159 non-null    float64
2   Length1     159 non-null    float64
3   Length2     159 non-null    float64
4   Length3     159 non-null    float64
5   Height      159 non-null    float64
6   Width       159 non-null    float64
dtypes: float64(6), object(1)
memory usage: 8.8+ KB
```

```
#Get summary statistics
df.describe()
```

	Weight	Length1	Length2	Length3	Height	Width
count	159.000000	159.000000	159.000000	159.000000	159.000000	159.000000
mean	398.326415	26.247170	28.415723	31.227044	8.970994	4.417486
std	357.978317	9.996441	10.716328	11.610246	4.286208	1.685804
min	0.000000	7.500000	8.400000	8.800000	1.728400	1.047600
25%	120.000000	19.050000	21.000000	23.150000	5.944800	3.385650
50%	273.000000	25.200000	27.300000	29.400000	7.786000	4.248500
75%	650.000000	32.700000	35.500000	39.650000	12.365900	5.584500
max	1650.000000	59.000000	63.400000	68.000000	18.957000	8.142000

```
#get shape of dataset
df.shape
```

```
(159, 7)
```

```
#get coloumn names
df.columns
```

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

```
#define y(dependent or label or target variable) and X(independent or feature or attribute)
y=df['Weight']
```

```
y.shape
```

```
(159,)
```

```
y
```

```
0      242.0
1      290.0
2      340.0
3      363.0
4      430.0
...
154     12.2
155     13.4
156     12.2
157     19.7
158     19.9
```

```
Name: Weight, Length: 159, dtype: float64
```

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

```
X.shape
```

```
(159, 5)
```

```
#to Train test split
```

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

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=2529)
```

```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

```
((111, 5), (48, 5), (111,), (48,))
```

```
#Get model train
```

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

```
model=LinearRegression()
```

```
model.fit(X_train,y_train)
```

```
LinearRegression()
```

```
#Get Model Prediction
```

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

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

```
y_pred.shape
```

```
(48,)
```

```
y_pred
```

```
array([ 485.76826299,  502.24720857,   94.72381964,  876.5711712 ,
        184.0789176 ,  219.30130488,  322.32532246,  376.22325991,
        372.35730485, -182.67537078, -160.60486837,  454.33586185,
        159.59755829,  843.48525226,  587.21680573,  299.53521445,
        597.72950823,  197.14605397,  639.89046741,   91.20067876,
        150.95424753, -103.08320574,  627.19712753,  795.69176861,
        814.68732975, -204.1496511 ,  329.98746856,  715.89288013,
        359.75634357,  792.3243925 ,  532.7036706 ,  552.00832342,
        433.48472727,  687.61750267, -204.76362537,  932.53668294,
        810.74234216,  -80.06217174,  284.36287887,  907.08036021,
        642.5828335 ,  959.33848223,  675.28792291,  718.86305458,
        623.89849226,  376.48346981,  530.83828119, -86.2357066 ])
```

```
#get model evaluation
```

```
from sklearn.metrics import mean_squared_error,mean_absolute_error,mean_absolute_percentage_error
```

```
mean_squared_error(y_test,y_pred)
```

```
16397.344524411372
```

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

```
103.02952922678541
```

```
mean_absolute_percentage_error(y_test,y_pred)
```

```
2.5082853471600246
```


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

```
0.8349141424416878
```

```
#Get future prediction
```

```
df_new=df.sample(1)
```

```
df_new
```

	Species	Weight	Length1	Length2	Length3	Height	Width	
146	Smelt	7.5	10.0	10.5	11.6	1.972	1.16	

```
X_new=df_new[['Height','Width','Length1','Length2','Length3']]
```

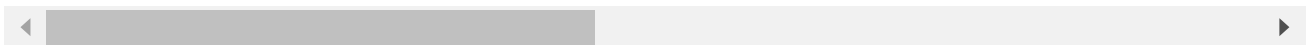
```
X_new.shape
```

```
(1, 5)
```

```
y_pred_new=model.predict(X_new)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:493: FutureWarning: The feature names must be in the same order as they were in fit.
```

```
warnings.warn(message, FutureWarning)
```



```
y_pred_new
```

```
array([-416.07874204])
```

✓ 0s completed at 11:52 AM ● ✕