In [5]:

```
import pandas as pd
fish=pd.read_csv("C:/Users/ADMIN/Downloads/archive/fish.csv")
fish.head()
```

Out[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

In [6]:

```
fish['Species']. unique()
```

Out[6]:

```
array(['Bream', 'Roach', 'Whitefish', 'Parkki', 'Perch', 'Pike', 'Smelt'],
      dtype=object)
```

In [12]:

```
fish.isnull().sum()
```

Out[12]:

```
Species    0
Weight     0
Length1    0
Length2    0
Length3    0
Height     0
Width      0
dtype: int64
```

In [17]:

```
X=fish.iloc[:,1:]
y=fish.loc[:,'Species']
```

In [18]:

```
X
```

Out[18]:

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

159 rows × 6 columns

In [19]:

```
y
```

Out[19]:

```
0      Bream
1      Bream
2      Bream
3      Bream
4      Bream
       ...
154    Smelt
155    Smelt
156    Smelt
157    Smelt
158    Smelt
Name: Species, Length: 159, dtype: object
```

In [25]:

```
from sklearn.preprocessing import MinMaxScaler
Scaler=MinMaxScaler()
Scaler.fit (X)
X_scaled=Scaler.transform (X)
```

Label Encoding t the target variable using LabelEncoder

In [20]:

```
from sklearn.preprocessing import LabelEncoder
label_encoder=LabelEncoder()
y=label_encoder. fit_transform(y)
y
```

Out[20]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
       5, 5, 5, 5, 5])
```

Splitting into train and test datasets using train_test_split

In [26]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split (X_scaled, y, test_size=0.2, random_state=42)
```

Model Building and training

In [27]:

```
from sklearn.linear_model import LogisticRegression
logReg=LogisticRegression()
logReg.fit(X_train,y_train)
```

Out[27]:

```
LogisticRegression()
```

Predicting the output

In [31]:

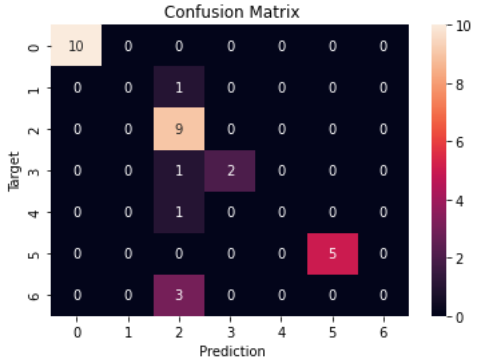
```
y_pred=logReg.predict(X_test)
```

In [33]:

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
cf= confusion_matrix(y_test, y_pred)
sns.heatmap (cf, annot=True)
plt.xlabel(" Prediction")
plt.ylabel("Target")
plt.title(" Confusion Matrix")
```

Out[33]:

```
Text(0.5, 1.0, ' Confusion Matrix')
```



In [ ]: