In [1]:
```
#IMPORTING LIBRARIES
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
```

In [23]:
```
#LOADING DATA
```

In [5]:
```
Iris = pd.read_csv(r"C:\Users\91918\Downloads\Iris.csv")
```

In [6]:
```
Iris
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [24]:
```
# DATA DESCRIPTION
```

In [7]:
```
Iris.describe()
```

Out[7]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [26]:
```
# DATA EXTRACTION
```

In [25]:
```
Data = Iris[['Id','SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']]
```

In [9]:
```
Data
```

Out[9]:

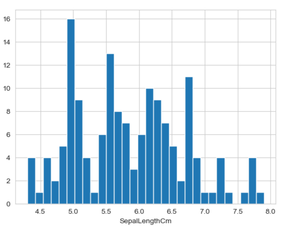
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	1	5.1	3.5	1.4	0.2
1	2	4.9	3.0	1.4	0.2
2	3	4.7	3.2	1.3	0.2
3	4	4.6	3.1	1.5	0.2
4	5	5.0	3.6	1.4	0.2
...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3
146	147	6.3	2.5	5.0	1.9
147	148	6.5	3.0	5.2	2.0
148	149	6.2	3.4	5.4	2.3
149	150	5.9	3.0	5.1	1.8

150 rows × 5 columns

In [27]:
```
#VISUALIZATION OF DATA
```

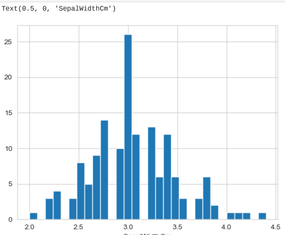
In [10]:
```
sns.set_style('whitegrid')
Iris['SepalLengthCm'].hist(bins=30)
plt.xlabel('SepalLengthCm')
```

Out[10]:
```
Text(0.5, 0, 'SepalLengthCm')
```



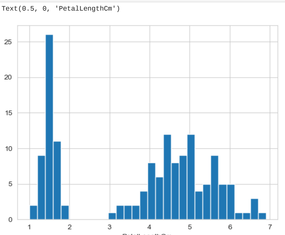
In [11]:
```
sns.set_style('whitegrid')
Iris['SepalWidthCm'].hist(bins=30)
plt.xlabel('SepalWidthCm')
```

Out[11]:
```
Text(0.5, 0, 'SepalWidthCm')
```



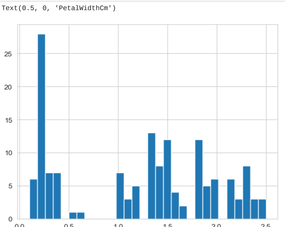
In [12]:
```
sns.set_style('whitegrid')
Iris['PetalLengthCm'].hist(bins=30)
plt.xlabel('PetalLengthCm')
```

Out[12]:
```
Text(0.5, 0, 'PetalLengthCm')
```



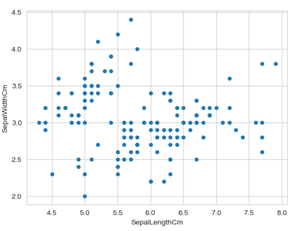
In [13]:
```
sns.set_style('whitegrid')
Iris['PetalWidthCm'].hist(bins=30)
plt.xlabel('PetalWidthCm')
```

Out[13]:
```
Text(0.5, 0, 'PetalWidthCm')
```



In [14]:
```
Iris.plot.scatter(x = "SepalLengthCm", y = "SepalWidthCm")
```

Out[14]:
```
<Axes: xlabel='SepalLengthCm', ylabel='SepalWidthCm'>
```



In [ ]:
```
#SPLITING DATA
```

In [17]:
```
x=Data
y=Iris['Species']
```

In [18]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33)
```

In [28]:
```
#MODEL BUILDING BY LOGISTIC REGRESSION ALGO
```

In [29]:
```
reg = LogisticRegression()
```

In [30]:
```
reg.fit(x_train,y_train)
```

Out[30]:
```
▾ LogisticRegression
LogisticRegression()
```

In [31]:
```
# CLASSIFICATION OF REPORT
```

In [32]:
```
prediction = reg.predict(x_test)
```

In [33]:
```
print(classification_report(y_test,prediction))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        16
Iris-versicolor       1.00      1.00      1.00        20
 Iris-virginica       1.00      1.00      1.00        14

       accuracy                           1.00        50
      macro avg       1.00      1.00      1.00        50
   weighted avg       1.00      1.00      1.00        50
```

In [ ]:
```

```