Module 5-2: Graphing Data

We will utilize the Iris dataset again in order to demonstrate clustering. Download the ir

Begin by importing the Iris dataset and `KMeans` from `sklearn`. Additionally, import `ma`

Python

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import pandas as pandas
from sklearn.preprocessing import LabelEncoder
```

Lets import our csv file into a dataframe name `df`. Next, insert the Sepal Length and Se named `X`.

Python

```
# Store the Iris data from Iris.csv into a python dataframe.
df = pandas.read_csv("Iris.csv")
X = df[['SepalLengthCm', 'SepalWidthCm']]
```

Insert Species into a dataframe named `y`.

Python

```
# Insert the species into a dataframe named y.
y = df['Species']
print(y)
```

Send Issue

Currently, `y` stores non numerical values. In order to use them, we need to use the labe numerical values into numerical values.

Python

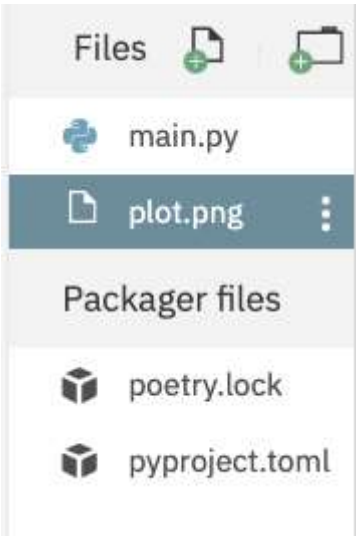
```
# Label Encoding to turn values into numerical.
le = LabelEncoder()
yEncoded = le.fit_transform(y)
print(yEncoded)
```

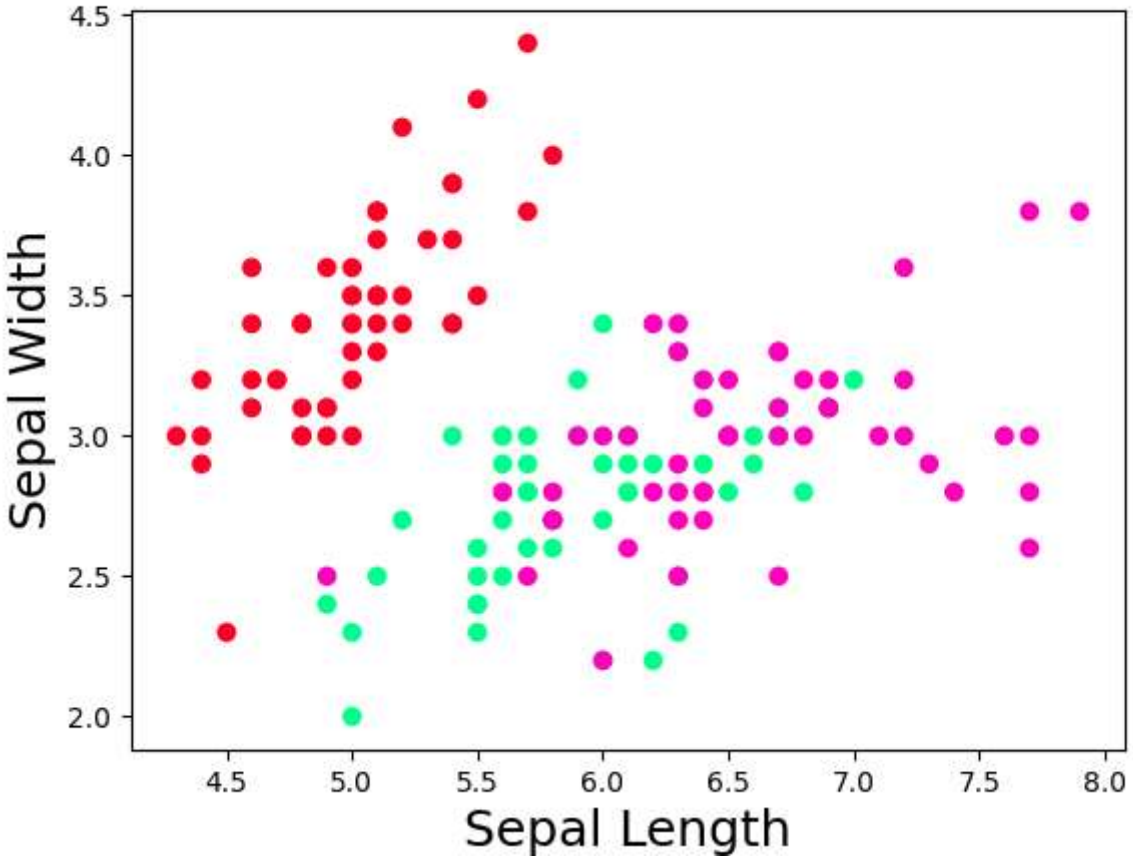
Now we can plot our data. We will use a scatterplot that is in `matplotlib` . For the x axis
For the y axis, we will call `SepalWidthCm` .

Python

```
# we will create a scatter plot.
plt.scatter(X['SepalLengthCm'],X['SepalWidthCm'], c = yEncoded, cma
plt.xlabel("Sepal Length", fontsize =18)
plt.ylabel("Sepal Width", fontsize = 18)
plt.savefig("plot.png")
```

The file `plot.png` should look something like this. The colors might be in a different or same thing.





Each data point signifies a flower, and each color shows what group that flower belongs

Define our `KMeans` model. Set the number of clusters (`n_clusters`) to 3 because we kr have only 3 classes of Iris flowers.

Python

```
km = KMeans(n_clusters = 3, random_state = 0)
```

The `fit()` function will train the machine learning model on the data.

Python

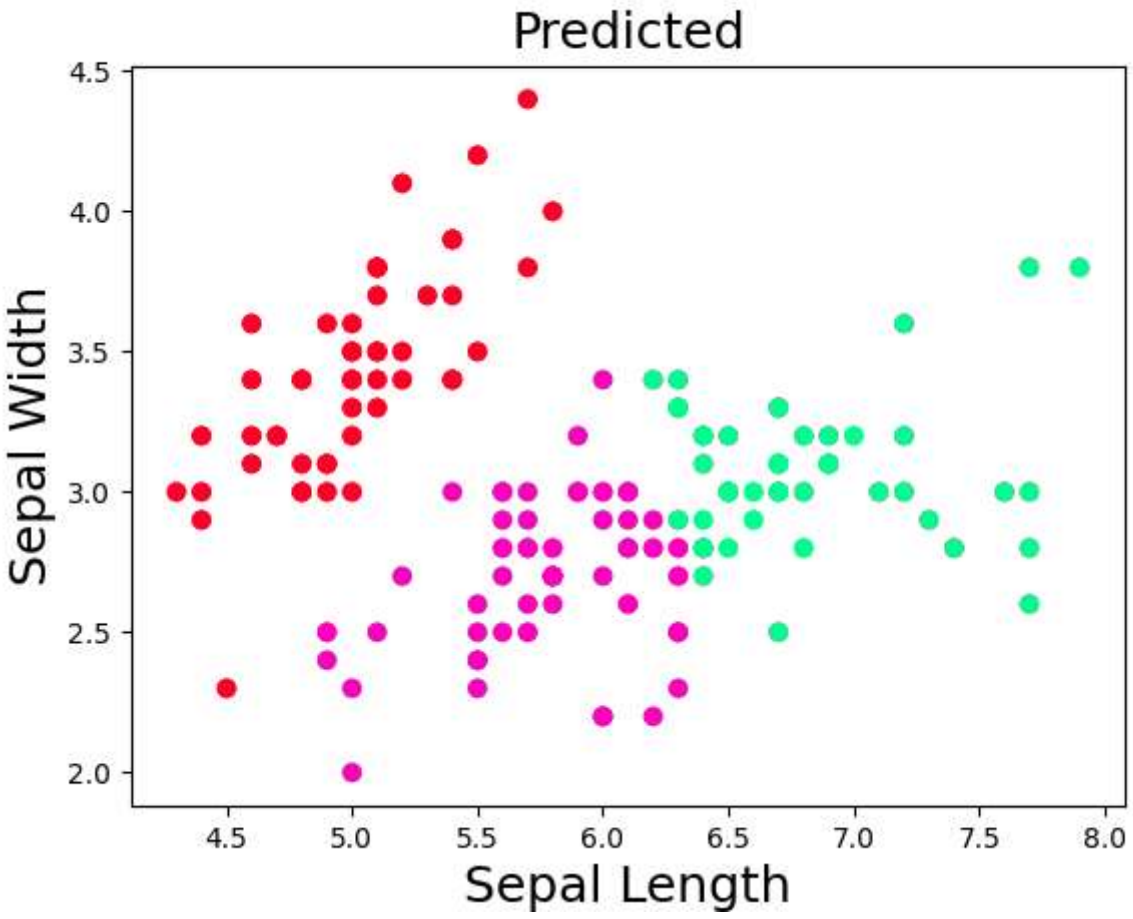
```
km.fit(X)
```

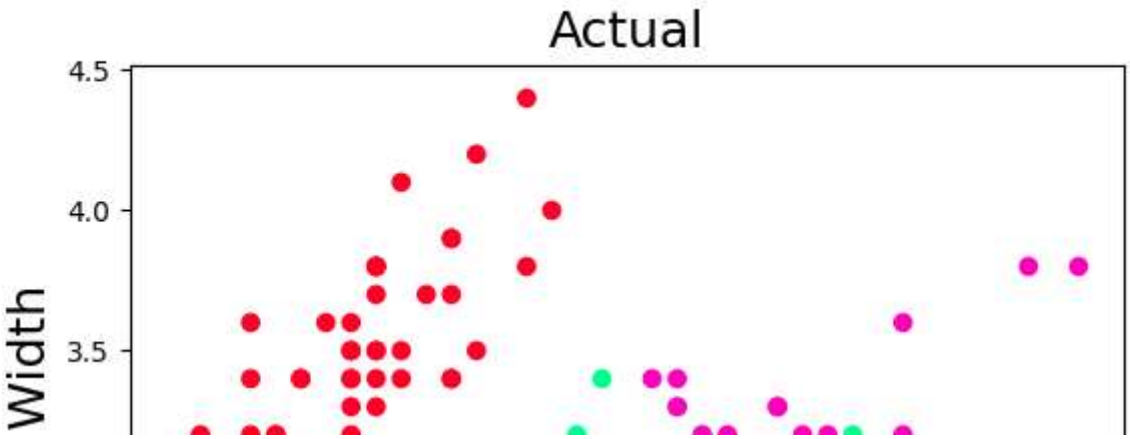
Plots the predictions of the Iris flower data set into a new chart called `predicted.png` .

Python

```
new_labels = km.labels_
plt.scatter(X['SepalLengthCm'],X['SepalWidthCm'], c = new_labels, c
plt.xlabel('Sepal Length', fontsize = 18)
plt.ylabel('Sepal Width', fontsize = 18)
plt.title("Predicted", fontsize = 18)
plt.savefig("Prediction.png")
```

Now we can take the two scatter plots and compare them. You can clearly see how KMe some of the data, but it overall did a pretty good job placing the flowers into the correc




Actual
Width
4.5
4.0
3.5