

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [ ]: import seaborn as sns
df=sns.load_dataset("iris")
df.head()
```

```
Out[ ]:   sepal_length  sepal_width  petal_length  petal_width  species
0          5.1           3.5           1.4           0.2    setosa
1          4.9           3.0           1.4           0.2    setosa
2          4.7           3.2           1.3           0.2    setosa
3          4.6           3.1           1.5           0.2    setosa
4          5.0           3.6           1.4           0.2    setosa
```

```
In [ ]: #design x and y
X= df.iloc[:, :-1] #features
y= df.iloc[ : , -1:] #labels
```

```
In [ ]: X.head()
```

```
Out[ ]:   sepal_length  sepal_width  petal_length  petal_width
0          5.1           3.5           1.4           0.2
1          4.9           3.0           1.4           0.2
2          4.7           3.2           1.3           0.2
3          4.6           3.1           1.5           0.2
4          5.0           3.6           1.4           0.2
```

```
In [ ]: y.head()
```

```
Out[ ]:   species
0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
```

```
In [ ]: #train and fir the model
```

```
from sklearn.naive_bayes import GaussianNB
model= GaussianNB()
model.fit(X, y)
```

C:\Users\Javeria\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

Out[]: GaussianNB()

```
In [ ]: 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=0)
```

```
In [ ]: # Predicting the Test set results
y_prediction = model.predict(X_test)
y_prediction
```

Out[]: array(['virginica', 'versicolor', 'setosa', 'virginica', 'setosa',
'virginica', 'setosa', 'versicolor', 'versicolor', 'versicolor',
'virginica', 'versicolor', 'versicolor', 'versicolor',
'versicolor', 'setosa', 'versicolor', 'versicolor', 'setosa',
'setosa', 'virginica', 'versicolor', 'setosa', 'setosa',
'virginica', 'setosa', 'setosa', 'versicolor', 'versicolor',
'setosa', 'virginica', 'versicolor', 'setosa', 'virginica',
'virginica', 'versicolor', 'setosa', 'versicolor', 'versicolor',
'versicolor', 'virginica', 'setosa', 'virginica', 'setosa',
'setosa'], dtype='<U10')

```
In [ ]: from sklearn import metrics
score = metrics.accuracy_score(y_test,y_prediction)
print("Gaussian Naive Bayes model accuracy (in %):", metrics.accuracy_score(y_test,y_pr
```

Gaussian Naive Bayes model accuracy (in %): 100.0 %

```
In [ ]: from sklearn import metrics
cm= metrics.confusion_matrix(y_test, y_prediction)
cm
```

Out[]: array([[16, 0, 0],
[0, 18, 0],
[0, 0, 11]], dtype=int64)

```
In [ ]: import seaborn as sns
plt.figure(figsize= (12,12))
sns.heatmap(cm, annot = True, fmt= ".3f", linewidths=.5, square = True, cmap = 'Spectral')
plt.ylabel("Actual label")
plt.xlabel("predicted label")
sample_title= ("Gaussian Naive Bayes model accuracy in (%): {0}" .format(score*100))
plt.title(sample_title, size= 15)
```

Out[]: Text(0.5, 1.0, 'Gaussian Naive Bayes model accuracy in (%): 100.0')

