 # Step 1: Load the Data

```
import pandas as pd
data=pd.read_csv('/content/Iris.csv')
```

```
data
```

	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

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
x = data.iloc[:,:4]
x
```

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

150 rows × 4 columns

Next steps: Generate code with x | View recommended plots | New interactive sheet

```
Y = data.iloc[:,5]
Y
```

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
            ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

```
x_train = x.iloc[:120]
x_train
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm
0	1	5.1	3.5	1.4
1	2	4.9	3.0	1.4
2	3	4.7	3.2	1.3
3	4	4.6	3.1	1.5
4	5	5.0	3.6	1.4
...	...	...	...	...
115	116	6.4	3.2	5.3
116	117	6.5	3.0	5.5
117	118	7.7	3.8	6.7
118	119	7.7	2.6	6.9
119	120	6.0	2.2	5.0

120 rows × 4 columns

Next steps: Generate code with `x_train` View recommended plots New interactive sheet

```
x_test = x.iloc[120:]
x_test
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm
120	121	6.9	3.2	5.7
121	122	5.6	2.8	4.9
122	123	7.7	2.8	6.7
123	124	6.3	2.7	4.9
124	125	6.7	3.3	5.7
125	126	7.2	3.2	6.0
126	127	6.2	2.8	4.8
127	128	6.1	3.0	4.9
128	129	6.4	2.8	5.6
129	130	7.2	3.0	5.8
130	131	7.4	2.8	6.1
131	132	7.9	3.8	6.4
132	133	6.4	2.8	5.6
133	134	6.3	2.8	5.1
134	135	6.1	2.6	5.6
135	136	7.7	3.0	6.1
136	137	6.3	3.4	5.6
137	138	6.4	3.1	5.5
138	139	6.0	3.0	4.8
139	140	6.9	3.1	5.4
140	141	6.7	3.1	5.6
141	142	6.9	3.1	5.1
142	143	5.8	2.7	5.1
143	144	6.8	3.2	5.9
144	145	6.7	3.3	5.7
145	146	6.7	3.0	5.2
146	147	6.3	2.5	5.0
147	148	6.5	3.0	5.2
148	149	6.2	3.4	5.4
149	150	5.9	3.0	5.1

Next steps: Generate code with `x_test` View recommended plots New interactive sheet

```
Y_train = Y.iloc[:120]
Y_train
```

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
115    Iris-virginica
116    Iris-virginica
117    Iris-virginica
118    Iris-virginica
119    Iris-virginica
Name: Species, Length: 120, dtype: object
```

```
Y_test = Y.iloc[120:]
Y_test
```

```
120    Iris-virginica
121    Iris-virginica
122    Iris-virginica
123    Iris-virginica
124    Iris-virginica
125    Iris-virginica
126    Iris-virginica
127    Iris-virginica
128    Iris-virginica
129    Iris-virginica
130    Iris-virginica
131    Iris-virginica
132    Iris-virginica
133    Iris-virginica
134    Iris-virginica
135    Iris-virginica
136    Iris-virginica
137    Iris-virginica
138    Iris-virginica
139    Iris-virginica
140    Iris-virginica
141    Iris-virginica
142    Iris-virginica
143    Iris-virginica
144    Iris-virginica
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, dtype: object
```

 # Step 2: Split the Data

```
# Step 4: Split the Data
from sklearn.model_selection import train_test_split
x = data.drop(['Id', 'Species'], axis=1)
Y = data['Species']
x_train, x_test, Y_train, Y_test = train_test_split(x, Y, test_size=0.2, random_state=1)
```

```
x_train
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
91	6.1	3.0	4.6	1.4
135	7.7	3.0	6.1	2.3
69	5.6	2.5	3.9	1.1
128	6.4	2.8	5.6	2.1
114	5.8	2.8	5.1	2.4
...	...	...	...	...
133	6.3	2.8	5.1	1.5
137	6.4	3.1	5.5	1.8
72	6.3	2.5	4.9	1.5
140	6.7	3.1	5.6	2.4
37	4.9	3.1	1.5	0.1

120 rows × 4 columns

Next steps: Generate code with `x_train` View recommended plots New interactive sheet

```
x_test
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
14	5.8	4.0	1.2	0.2
98	5.1	2.5	3.0	1.1
75	6.6	3.0	4.4	1.4
16	5.4	3.9	1.3	0.4
131	7.9	3.8	6.4	2.0
56	6.3	3.3	4.7	1.6
141	6.9	3.1	5.1	2.3
44	5.1	3.8	1.9	0.4
29	4.7	3.2	1.6	0.2
120	6.9	3.2	5.7	2.3
94	5.6	2.7	4.2	1.3
5	5.4	3.9	1.7	0.4
102	7.1	3.0	5.9	2.1
51	6.4	3.2	4.5	1.5
78	6.0	2.9	4.5	1.5
42	4.4	3.2	1.3	0.2
92	5.8	2.6	4.0	1.2
66	5.6	3.0	4.5	1.5
31	5.4	3.4	1.5	0.4
35	5.0	3.2	1.2	0.2
90	5.5	2.6	4.4	1.2
84	5.4	3.0	4.5	1.5
77	6.7	3.0	5.0	1.7
40	5.0	3.5	1.3	0.3
125	7.2	3.2	6.0	1.8
99	5.7	2.8	4.1	1.3
33	5.5	4.2	1.4	0.2
19	5.1	3.8	1.5	0.3
73	6.1	2.8	4.7	1.2
146	6.3	2.5	5.0	1.9

Next steps: Generate code with `x_test` View recommended plots New interactive sheet

```
Y_train
```

```
91     Iris-versicolor
135     Iris-virginica
69     Iris-versicolor
128     Iris-virginica
114     Iris-virginica
            ...
133     Iris-virginica
137     Iris-virginica
72     Iris-versicolor
140     Iris-virginica
37         Iris-setosa
Name: Species, Length: 120, dtype: object
```

```
Y_test
```

```
14         Iris-setosa
98     Iris-versicolor
75     Iris-versicolor
16         Iris-setosa
131     Iris-virginica
56     Iris-versicolor
141     Iris-virginica
44         Iris-setosa
29         Iris-setosa
120     Iris-virginica
94     Iris-versicolor
5          Iris-setosa
102     Iris-virginica
51     Iris-versicolor
78     Iris-versicolor
42         Iris-setosa
92     Iris-versicolor
66     Iris-versicolor
31         Iris-setosa
35         Iris-setosa
90     Iris-versicolor
84     Iris-versicolor
77     Iris-versicolor
40         Iris-setosa
125     Iris-virginica
99     Iris-versicolor
33         Iris-setosa
19         Iris-setosa
73     Iris-versicolor
146     Iris-virginica
Name: Species, dtype: object
```

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
# Step 2: Label Encode the Target Variable
label_encoder = LabelEncoder()
data['Species'] = label_encoder.fit_transform(data['Species'])
```

 # Step 3: Initialize the Model

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
```

 # Step 4: Train the Model

```
model.fit(x_train, Y_train)
```

```
▾ GaussianNB
GaussianNB()
```

 # Step 5: Make Predictions

```
Y_pred = model.predict(x_train)
Y_pred = model.predict(x_test)
```

 # Step 6: Calculate Accuracy

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(Y_test, Y_pred)

print(f'Accuracy: {accuracy}')
```

```
Accuracy: 0.9666666666666667
```

 # Step 7: Calculate recall_score

```
from sklearn.metrics import recall_score
recall = recall_score(Y_test, Y_pred, average='macro')

print(f'Recall: {recall}')
```

```
Recall: 0.9743589743589745
```

 # Step 8: Calculate precision

```
from sklearn.metrics import precision_score
precision = precision_score(Y_test, Y_pred, average='macro')

print(f'Precision: {precision}')
```

```
Precision: 0.9523809523809524
```

step 9 = confusion_matrix

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, Y_pred)

print('Confusion Matrix:')
print(cm)
```

```
Confusion Matrix:
[[11  0  0]
 [ 0 12  1]
 [ 0  0  6]]
```

Step 10 = Calculate misclassifications

```
misclassifications = (Y_test != Y_pred).sum()
prevalence = Y_test.value_counts(normalize=True)

print(f'Misclassifications: {misclassifications}')
print(f'Prevalence:\n{prevalence}')
```

```
Misclassifications: 1
Prevalence:
Species
Iris-versicolor    0.433333
```