```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
%matplotlib inline
```

```
columns = ['Sepal length', 'Sepal width', 'Petal length', 'Petal width', 'Class_labels']
# load the data
iris = pd.read_csv("/content/iris.data", names = columns)
iris.head(5)
```

	Sepal length	Sepal width	Petal length	Petal width	Class_labels
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
iris.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Sepal length  150 non-null    float64
 1   Sepal width   150 non-null    float64
 2   Petal length  150 non-null    float64
 3   Petal width   150 non-null    float64
 4   Class_labels  150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
iris.isnull().sum()
```

```
Sepal length    0
Sepal width     0
Petal length    0
Petal width     0
Class_labels    0
dtype: int64
```

```
iris.describe()
```

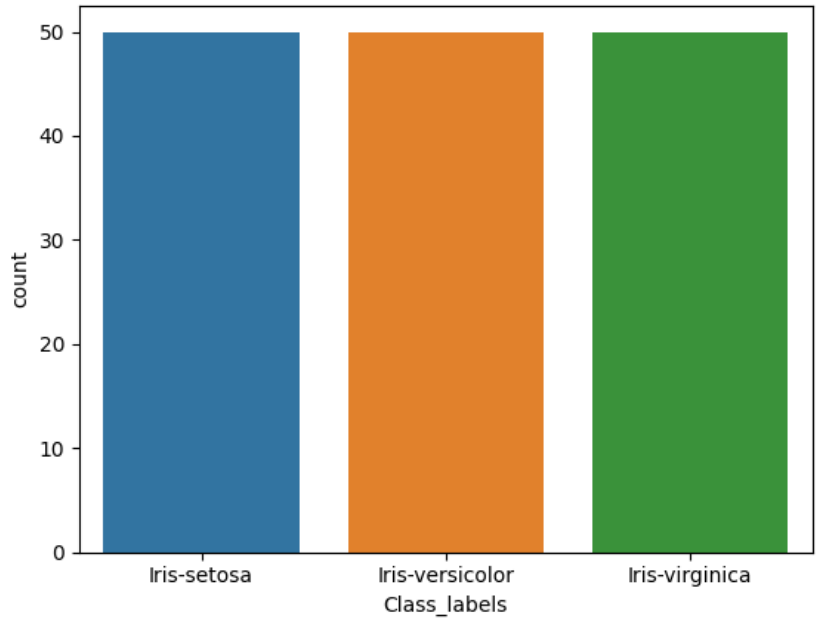
	Sepal length	Sepal width	Petal length	Petal width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000

```
iris['Class_labels'].value_counts()
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Class_labels, dtype: int64
```

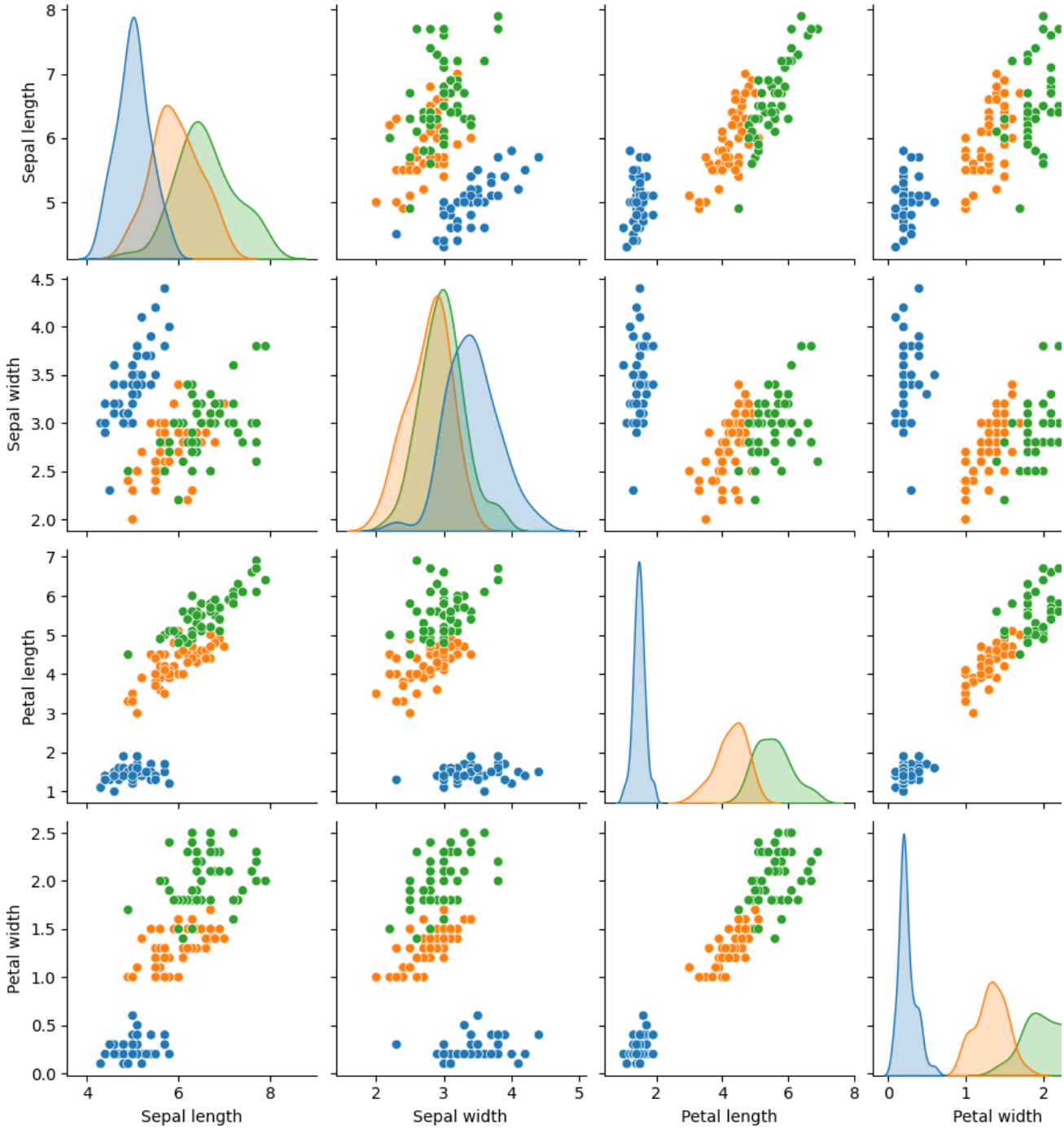
```
sns.countplot(data=iris, x='Class_labels')
```

```
<Axes: xlabel='Class_labels', ylabel='count'>
```




```
sns.pairplot(data = iris ,hue = 'Class_labels')
```

```
<seaborn.axisgrid.PairGrid at 0x7f51c73bf100>
```




```
x=iris.iloc[:,:4]
y=iris.iloc[:,4]
```

```
x
```

	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
...	...	...	...	...

```
x = iris.drop(['Sepal length', 'Class_labels'], axis=1)
y = iris['Class_labels']
```

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

```
x_train, x_test, y_train, y_test = train_test_split(x ,y, test_size = 0.2, random_state=42)
```

```
x_train.shape
```

```
(120, 3)
```

```
x_test.shape
```

```
(30, 3)
```

```
y_train.shape
```

```
(120,)
```

```
y_test.shape
```

```
(30,)
```

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

```
iris['Sepal width'] = pd.to_numeric(iris['Sepal width'], errors='coerce')
```

```
model.fit(x_train,y_train)
LogisticRegression()
```

```
▾ LogisticRegression
LogisticRegression()
```

```
y_pred = model.predict(x_test)

# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 1.0
```