```
import pandas as pd
import seaborn as sns
```

```
dt = pd.read_csv('/content/iris.csv')
```

```
dt
```

	sepal.length	sepal.width	petal.length	petal.width	variety
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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

```
dt.describe()
```

	sepal.length	sepal.width	petal.length	petal.width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
dt.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   variety       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
dt['variety'].unique()
```

```
array(['Setosa', 'Versicolor', 'Virginica'], dtype=object)
```

```
from sklearn.preprocessing import LabelEncoder
```

```
l = LabelEncoder()
```

```
dt['variety'] = l.fit_transform(dt['variety'])
```

```
dt['variety'].unique()
```

```
array([0, 1, 2])
```

```
dt
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

```
from sklearn.model_selection import train_test_split
```

```
x = dt.drop(['variety'],axis=1)
y = dt['variety']
```

```
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
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

```
dt.corr()
```

	sepal.length	sepal.width	petal.length	petal.width	variety
sepal.length	1.000000	-0.117570	0.871754	0.817941	0.782561
sepal.width	-0.117570	1.000000	-0.428440	-0.366126	-0.426658
petal.length	0.871754	-0.428440	1.000000	0.962865	0.949035
petal.width	0.817941	-0.366126	0.962865	1.000000	0.956547
variety	0.782561	-0.426658	0.949035	0.956547	1.000000

```
xtrain, xtest, ytrain, ytest = train_test_split(x,y, test_size=0.25)
```

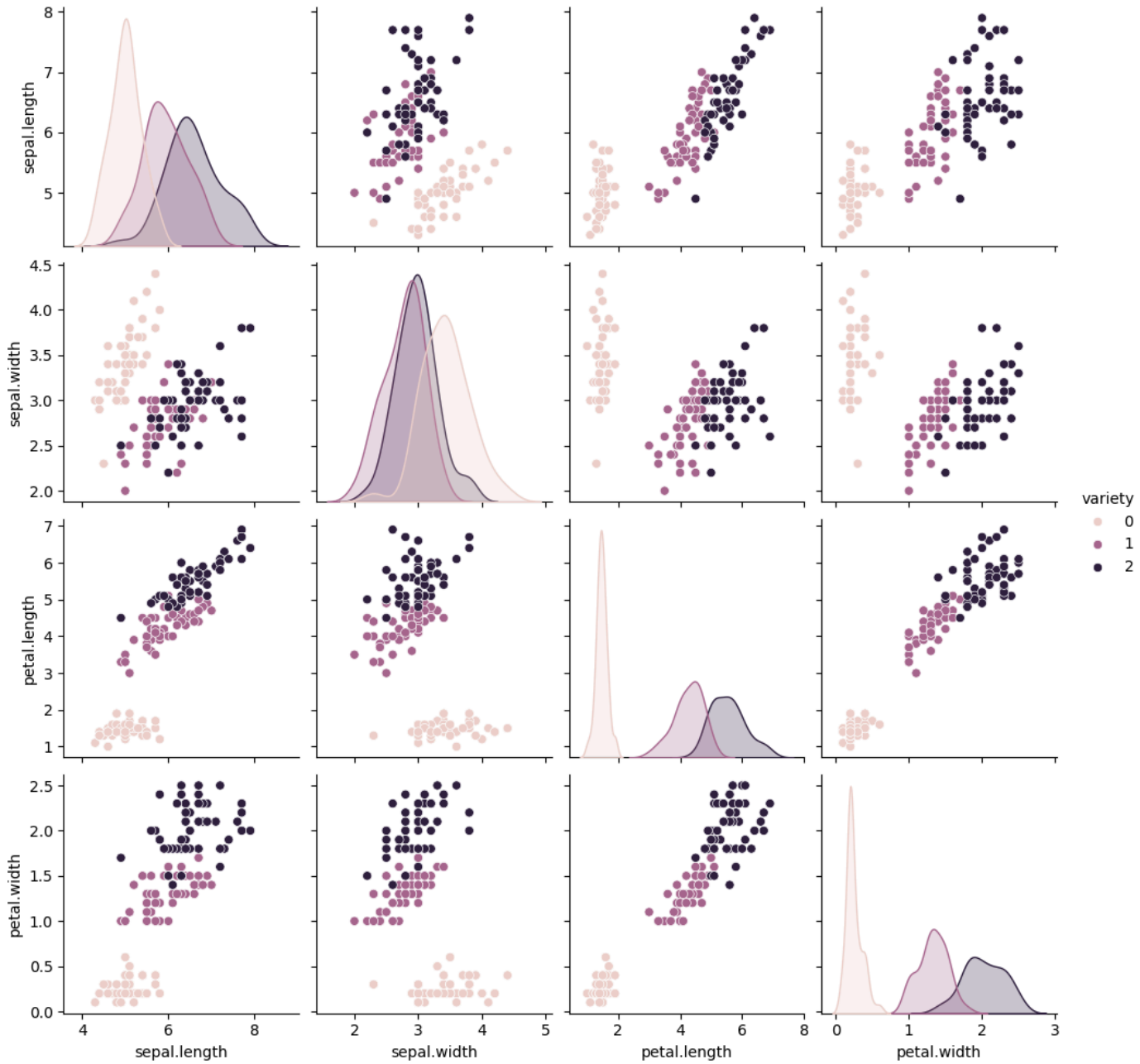
```
xtest
```

	sepal.length	sepal.width	petal.length	petal.width
80	5.5	2.4	3.8	1.1
67	5.8	2.7	4.1	1.0
6	4.6	3.4	1.4	0.3
103	6.3	2.9	5.6	1.8
118	7.7	2.6	6.9	2.3
127	6.1	3.0	4.9	1.8
111	6.4	2.7	5.3	1.9
141	6.9	3.1	5.1	2.3
0	5.1	3.5	1.4	0.2
104	6.5	3.0	5.8	2.2
89	5.5	2.5	4.0	1.3
149	5.9	3.0	5.1	1.8
30	4.8	3.1	1.6	0.2
144	6.7	3.3	5.7	2.5
52	6.9	3.1	4.9	1.5
117	7.7	3.8	6.7	2.2
142	5.8	2.7	5.1	1.9
62	6.0	2.2	4.0	1.0
131	7.9	3.8	6.4	2.0
143	6.8	3.2	5.9	2.3
107	7.3	2.9	6.3	1.8
147	6.5	3.0	5.2	2.0
51	6.4	3.2	4.5	1.5
83	6.0	2.7	5.1	1.6
124	6.7	3.3	5.7	2.1
135	7.7	3.0	6.1	2.3
69	5.6	2.5	3.9	1.1
113	5.7	2.5	5.0	2.0
81	5.5	2.4	3.7	1.0
121	5.6	2.8	4.9	2.0
137	6.4	3.1	5.5	1.8

```
sns.pairplot(dt, hue='variety')
```

60	5.0	2.0	3.5	1.0
74	6.4	2.9	4.3	1.3
2	4.7	3.2	1.3	0.2
39	5.1	3.4	1.5	0.2
106	4.9	2.5	4.5	1.7
138	6.0	3.0	4.8	1.8

```
<seaborn.axisgrid.PairGrid at 0x7b8408561250>
```




```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()

lr.fit(xtrain,ytrain)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
ypred = lr.predict(xtest)

from sklearn.metrics import accuracy_score

accuracy_score(ytest,ypred)
```

```
0.8947368421052632
```

```
import pickle

pickle.dump(lr,open('iris.pkl','wb'))
```

Start coding or generate with AI.