






```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
```

```
df=sns.load_dataset('iris')
df.head()
```

	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	

df

	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	
...	
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




```
df['species'].unique()
```

```
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

```
le=LabelEncoder()
```

```
df['species']=df['species'].map({'setosa':0,'versicolor':1,'virginica':2})
```




df

	sepal_length	sepal_width	petal_length	petal_width	species	
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

```
x=df.iloc[:, :-1]
y=df.iloc[:, -1:]
```

y

	species	
0	0	
1	0	
2	0	
3	0	
4	0	
...	...	
145	2	
146	2	
147	2	
148	2	
149	2	

150 rows × 1 columns

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

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1,random_state=42)
```

```
from sklearn.linear_model import LogisticRegression
```

```
model=LogisticRegression()
```

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

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: 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 y = column_or_1d(y, warn=True)
```

```
▼ LogisticRegression  
LogisticRegression()
```

```
y_pred=model.predict(x_test)
```

```
model.predict([[5.1,3.5,1.4,0.2]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names: Feature names are not present  
warnings.warn(  
array([0])
```

◀ ▶

```
model.predict([[5.4,3.7,1.6,0.3]])
```

```
📄 /usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names: Feature names are not present  
warnings.warn(  
array([0])
```

◀ ▶

Start coding or [generate](#) with AI.