

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
```

```
In [2]: df=sns.load_dataset('iris')
df.head()
```

Out[2]:

	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 [3]: le=LabelEncoder()
```

```
In [4]: df['species']=le.fit_transform(df['species'])
df.head()
```

Out[4]:

	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

```
In [5]: df=df[['sepal_length','petal_length','species']]
```

In [6]: `df.head()`

Out[6]:

	sepal_length	petal_length	species
0	5.1	1.4	0
1	4.9	1.4	0
2	4.7	1.3	0
3	4.6	1.5	0
4	5.0	1.4	0

In [7]: `x=df.iloc[:,0:2]`
`y=df.iloc[:,-1]`

In [8]: `x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)`

In [9]: `clf=LogisticRegression(multi_class='multinomial')`

In [10]: `clf.fit(x_train,y_train)`

Out[10]: `LogisticRegression(multi_class='multinomial')`

In [11]: `y_pred=clf.predict(x_test)`

In [12]: `print(accuracy_score(y_test,y_pred))`

0.9666666666666667

In [13]: `pd.DataFrame(confusion_matrix(y_test,y_pred))`

Out[13]:

	0	1	2
0	11	0	0
1	0	13	0
2	0	1	5

In [14]: `query=np.array([[3.4,2.7]])`
`clf.predict_proba(query)`

Out[14]: `array([[7.73291920e-01, 2.26333807e-01, 3.74272780e-04]])`

In [15]: `clf.predict(query)`

Out[15]: `array([0])`

In [16]: `!pip install mlxtend`

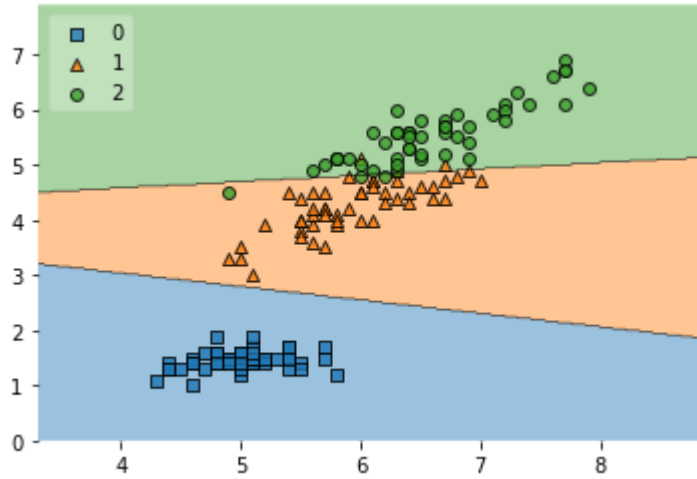
```
Requirement already satisfied: mlxtend in c:\users\user10\anaconda3\lib\site-packages (0.20.0)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\user10\anaconda3\lib\site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: scipy>=1.2.1 in c:\users\user10\anaconda3\lib\site-packages (from mlxtend) (1.7.3)
Requirement already satisfied: joblib>=0.13.2 in c:\users\user10\anaconda3\lib\site-packages (from mlxtend) (1.1.0)
Requirement already satisfied: pandas>=0.24.2 in c:\users\user10\anaconda3\lib\site-packages (from mlxtend) (1.4.2)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\user10\anaconda3\lib\site-packages (from mlxtend) (3.5.1)
Requirement already satisfied: numpy>=1.16.2 in c:\users\user10\anaconda3\lib\site-packages (from mlxtend) (1.21.5)
Requirement already satisfied: setuptools in c:\users\user10\anaconda3\lib\site-packages (from mlxtend) (61.2.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\user10\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\user10\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\user10\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (9.0.1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\user10\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\user10\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\user10\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
Requirement already satisfied: packaging>=20.0 in c:\users\user10\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (21.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\user10\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\user10\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\user10\anaconda3\lib\site-packages (from scikit-learn>=1.0.2->mlxtend) (2.2.0)
```

In [17]: `y`

```
Out[17]: 0      0
         1      0
         2      0
         3      0
         4      0
         ..
        145     2
        146     2
        147     2
        148     2
        149     2
        Name: species, Length: 150, dtype: int32
```

```
In [18]: from mlxtend.plotting import plot_decision_regions  
plot_decision_regions(x.values,y.values,clf,legend=2)
```

Out[18]: <AxesSubplot:>



In []: