Ungraded Lab: Logistic Regression using Scikit-Learn

Goals

In this lab you will:

- Train a logistic regression model using scikit-learn.

Dataset

Let's start with the same dataset as before.

In [1]:
```
import numpy as np

X = np.array([[0.5, 1.5], [1,1], [1.5, 0.5], [3, 0.5], [2, 2], [1, 2
y = np.array([0, 0, 0, 1, 1, 1])
```

Fit the model

The code below imports the logistic regression model (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.l from scikit-learn. You can fit this model on the training data by calling `fit` function.

In [2]:
```
from sklearn.linear_model import LogisticRegression

lr_model = LogisticRegression()
lr_model.fit(X, y)
```

Out[2]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_interc
ept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=10
0,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, v
erbose=0,
                   warm_start=False)
```

Make Predictions

You can see the predictions made by this model by calling the `predict` function.

In [3]:
```
y_pred = lr_model.predict(X)

print("Prediction on training set:", y_pred)
```

```
Prediction on training set: [0 0 0 1 1 1]
```

Calculate accuracy

You can calculate this accuracy of this model by calling the `score` function.

In [4]:
```
print("Accuracy on training set:", lr_model.score(X, y))
```

```
Accuracy on training set: 1.0
```