

```
In [1]: import numpy as np
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

```
In [2]: #loading dataset
data=np.loadtxt('ex2data1.txt',delimiter=',')
```

```
In [3]: x=data[:,0:2]
y=data[:,2:3]
```

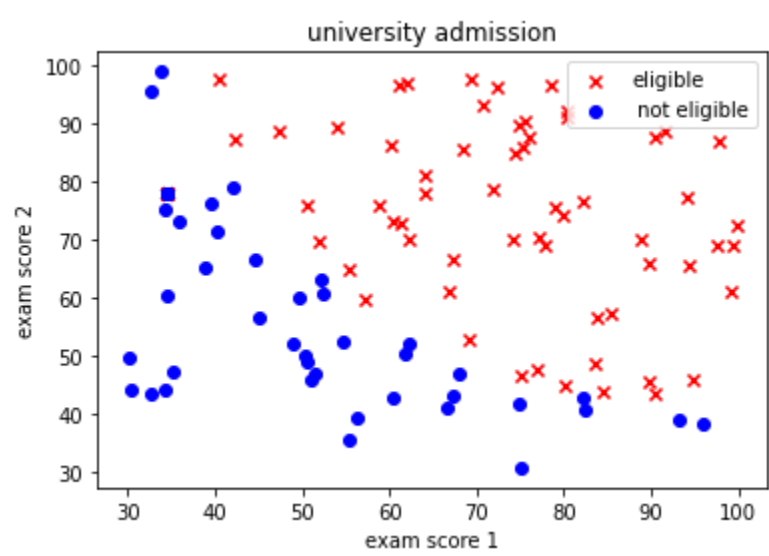
```
In [4]: x.shape
```

Out[4]: (100, 2)

```
In [5]: get_admission=np.where(y==1)
no_admission=np.where(y==0)
no_admission
```

Out[5]: (array([ 0, 1, 2, 5, 10, 11, 14, 17, 20, 22, 23, 27, 28, 29, 32, 34, 35,
36, 38, 39, 41, 43, 44, 45, 53, 54, 55, 57, 61, 62, 63, 64, 65, 67,
70, 78, 79, 86, 89, 92], dtype=int64),
array([0, 0,
0, 0], dtype=int64))

```
In [6]: get_admission=np.where(y==1)
no_admission=np.where(y==0)
plt.scatter(x[get_admission,0],x[get_admission,1],label='eligible',c='r',marker='x')
plt.scatter(x[no_admission,0],x[no_admission,1],label=' not eligible',c='b',marker='o')
plt.title("university admission")
plt.xlabel("exam score 1")
plt.ylabel("exam score 2")
plt.legend()
plt.show()
```



```
In [7]: #splitting the data into train and test
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
In [8]: #call the algorithm
k=LogisticRegression()
```

```
In [9]: #fit method to load the train datasets into the called algorithm
k.fit(x_train,y_train)

C:\Users\SAI RUPA\anaconda3\lib\site-packages\sklearn\utils\validation.py:73: 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 ravel().
    return f(**kwargs)
```

Out[9]: LogisticRegression()

```
In [10]: y_pred=k.predict(x_test) #predicted data
y_pred
```

Out[10]: array([0., 0., 0., 0., 1., 1., 0., 1., 0., 1., 0., 0., 0., 1., 0., 1., 0.,
1., 0., 1.])

```
In [11]: y_test.reshape(1,20) #actual data
```

Out[11]: array([[1., 0., 0., 0., 1., 1., 1., 1., 0., 1., 0., 0., 0., 1., 1., 1.,
0., 1., 1., 1.]])

```
In [12]: #accuracy
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)
```

Out[12]: 0.8

```
In [13]: #check for the new data
x_n=np.array([[50,75]])
y_n=k.predict(x_n)
print('for a student with scores 45 and 85,we predict an admission of:',y_n)

for a student with scores 45 and 85,we predict an admission of: [1.]
```

```
In [ ]:
```

```
In [ ]:
```