

سوال 1

ابتدا فایل را لود میکنیم

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.linear_model import Perceptron
import matplotlib.pyplot as plt
```

In [118]:

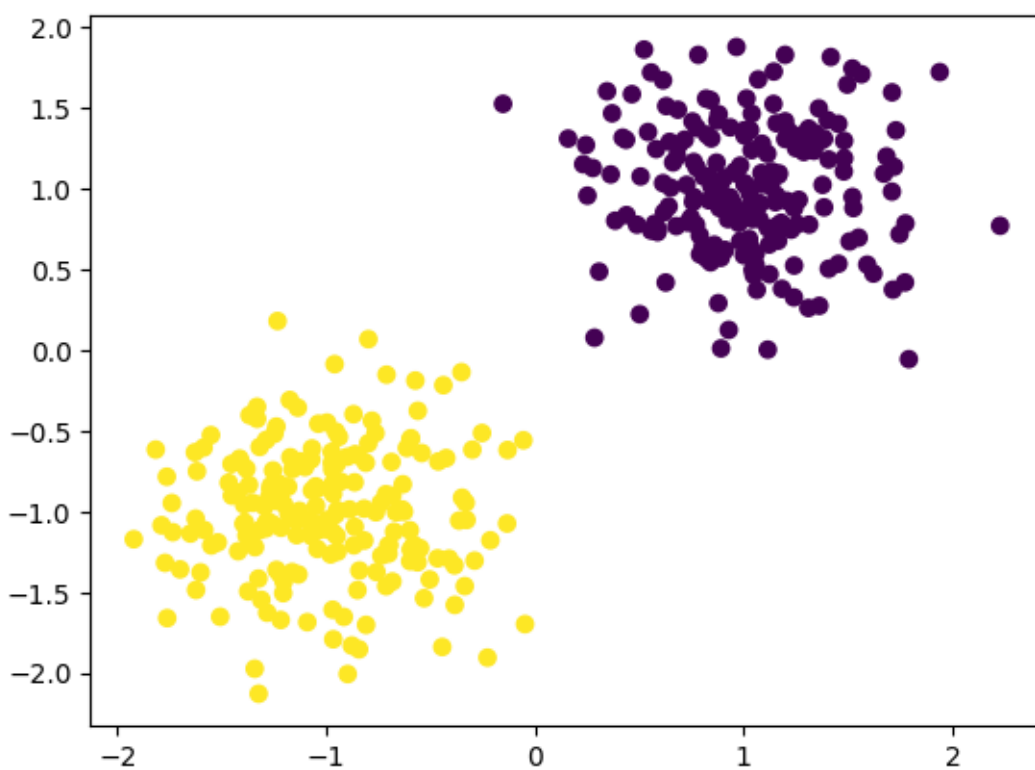
```
!gdown 1PnHCotN9UaydInJw5WWq2uOwtg1A_POa
```

-ستون اول و دوم فایل CSV مربوط به ویژگی ها و ستون سوم آن مربوط به کلاس هر داده است.

	x1	x2	y
0	1.028503	0.973218	-1.0
1	0.252505	0.955872	-1.0

ویژگی های تابع را در x و تارگت را در y ذخیره میکنیم سپس نمودار را رسم میکنیم

```
X = datasets[['x1', 'x2']]
y = datasets['y']
a = datasets['x1']
b = datasets['x2']
plt.scatter(a, b, c=y)
print(X.shape)
plt.show
```



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 23)

clf = Perceptron()

clf.fit(X_train, y_train)
```

سپس داده هارا به دوقسمت ترین و تست تقسیم میکنیم و در ادامه با قایده پرسپترون آن را آموزش میدهم

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 23)

clf = Perceptron()

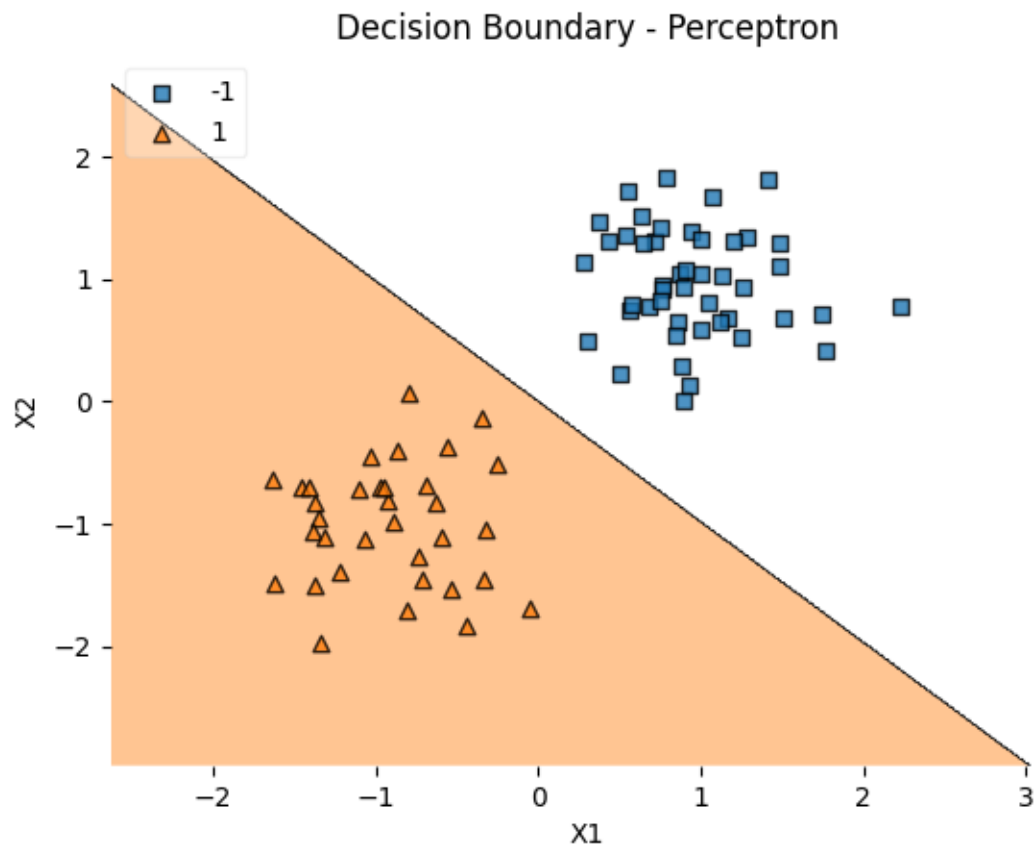
clf.fit(X_train, y_train)
```

نتایج به دست آمده قابل قبول است و اکورسی 1 به دست آمده یعنی داده کاملاً به درستی از هم جدا شده اند

حال باید شکل جدا شدن این داده هارا رسم کنیم .
من اول با استفاده از توابع آماده این کار کردم

```
from mlxtend.plotting import plot_decision_regions

# Plotting decision boundary
plot_decision_regions(X_test.values, y_test.values.astype(np.int), clf=clf,
legend=2)
plt.xlabel('X1')
plt.ylabel('X2')
plt.title('Decision Boundary - Perceptron')
plt.show()
```

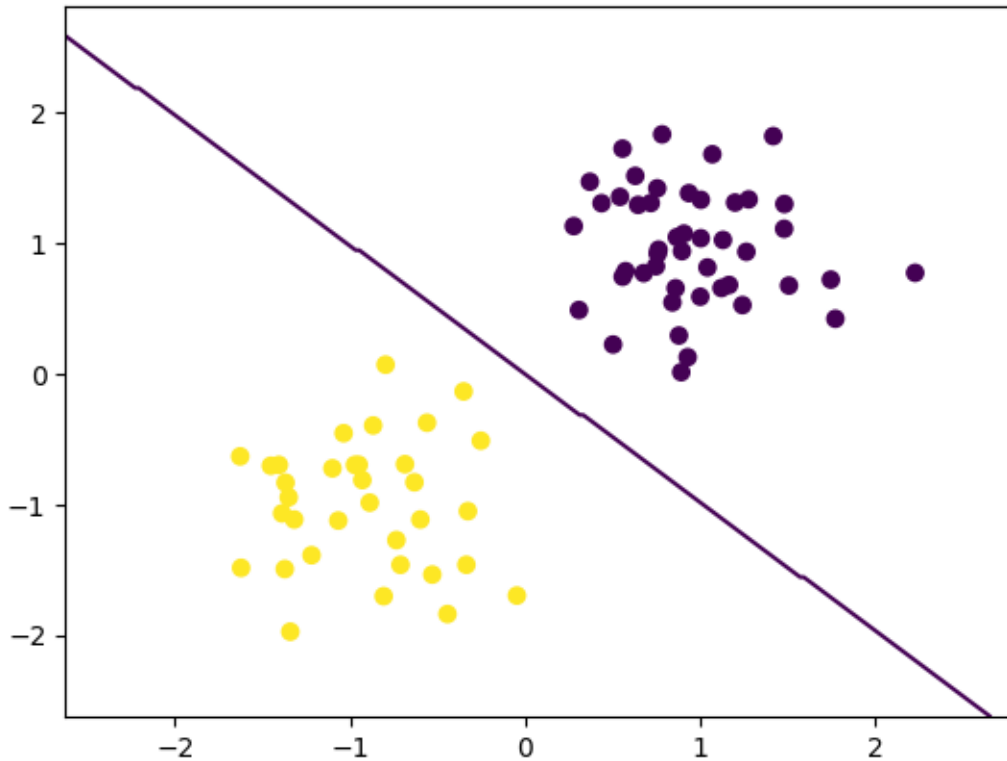


و در ادامه خودم نمودار بدون استفاده از توابع آماده رسم کردم

```
X_test_array = X_test.values
x1_min, x2_min = X_test_array[:, 0].min()-1, X_test_array[:, 0].min()-1
x1_max, x2_max = X_test_array[:, 1].max()+1, X_test_array[:, 1].max()+1

x1m, x2m = np.meshgrid(np.arange(x1_min, x1_max, 0.02),
                        np.arange(x2_min, x2_max, 0.02))
xm = np.stack((x1m.flatten(), x2m.flatten()), axis=1)
print(x1m.shape)
ym=clf.predict(xm)

levels = [-1,0,1]
plt.scatter(X_test_array[:, 0], X_test_array[:, 1], c=y_test)
plt.contour(x1m, x2m, ym.reshape(x1m.shape), levels=[0])
plt.show()
```



قسمت 3

این دفعه با آستانه مدل را آموزش می‌دهیم تا نتایج را ببینیم

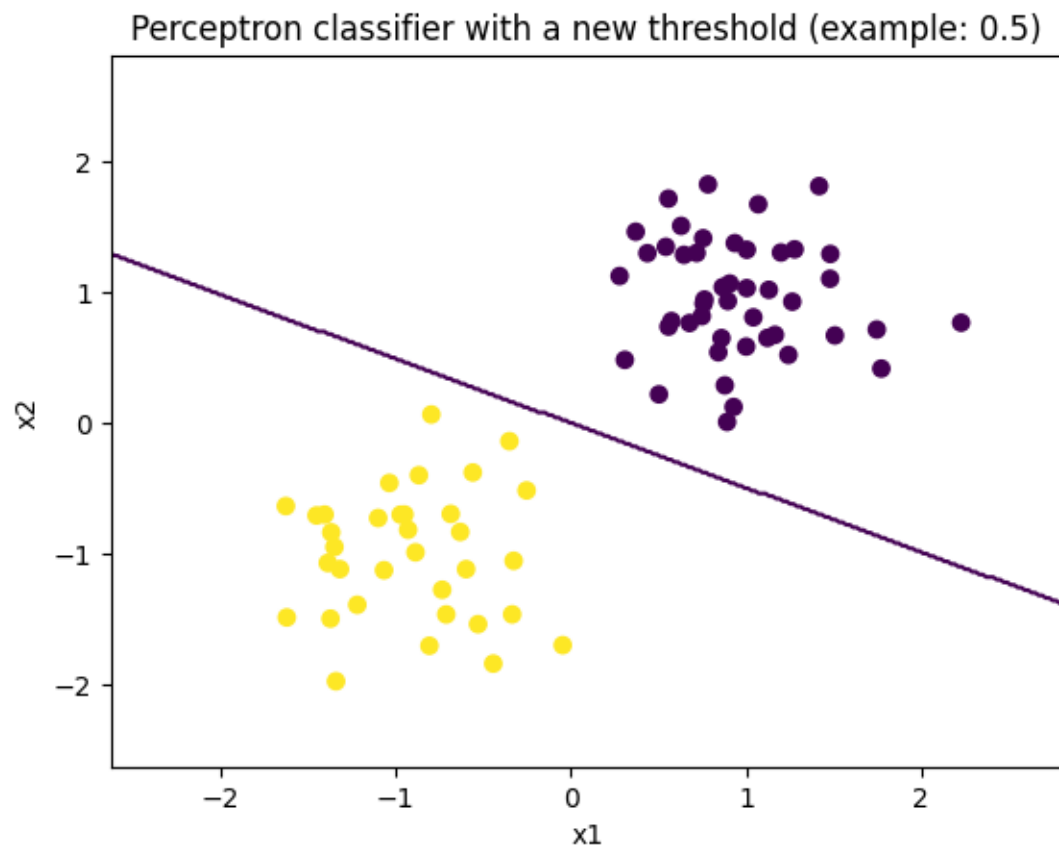
```
clf1 = Perceptron(random_state=23, tol=0.001, max_iter=1000, eta0=0.1,  
verbose=0, n_jobs=-1)  
clf1.fit(X_train, y_train)
```

```
test_accuracy_1 = clf1.score(X_test, y_test)  
test_accuracy_1
```

همانطور که مشاهده میشود باز دقت 1 است و یعنی تمامی داده ها به درستی آموزش دیده اند
حال شکل را رسم میکنیم

```
ym2 = clf1.predict(xm)

plt.contour(x1m, x2m, ym2.reshape(x1m.shape), levels=[0])
plt.scatter(X_test_array[:, 0], X_test_array[:, 1], c=y_test)
plt.xlabel('x1')
plt.ylabel('x2')
plt.title('Perceptron classifier with a new threshold (example: 0.5)')
plt.show()
```



مقایسه خروجی و توضیح تاثیر انتخاب آستانه در پرسپترون:

انتخاب آستانه (threshold) در الگوریتم پرسپترون بر طبقه بندی نتایج تاثیر مهمی دارد. آستانه به عنوان یک مقدار تصمیمگیری استفاده میشود که مشخص میکند که خروجی مدل به عنوان کلاس مثبت یا منفی تشخیص داده شود.

تأثیر انتخاب آستانه بر نتایج طبقه‌بندی به شرح زیر است:

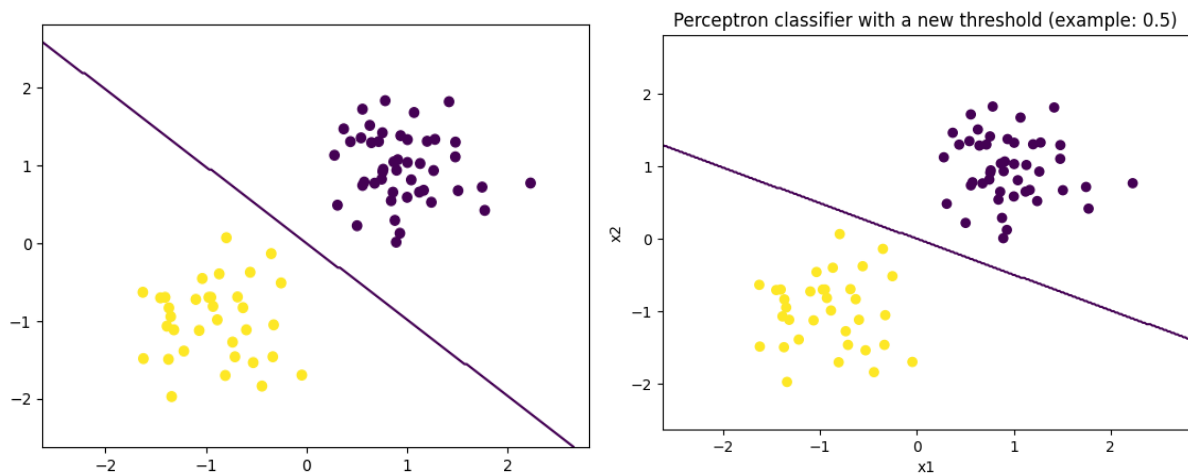
۱. تعیین دقت: انتخاب آستانه میتواند تأثیر مستقیمی بر دقت طبقه‌بندی داشته باشد. زیرا انتخاب آستانه منجر به تعیین تعداد نمونه هایی میشود که به درستی طبقه‌بندی شده‌اند یا نه.

۲. تعیین تعادل بین حساسیت و ویژگی خاصیت: انتخاب آستانه میتواند تعادل بین حساسیت (توانایی تشخیص نمونه های مثبت) و ویژگی خاصیت (توانایی تشخیص نمونه های منفی) را تعیین کند. این موضوع میتواند در مواقعی که ترجیح داده میشود یک نوع خطا را بر نوع دیگر ترجیح دهیم، بسیار مهم باشد.

۳. تأثیر بر توانایی تشخیص: انتخاب آستانه میتواند تأثیر مستقیمی بر توانایی مدل در تشخیص نمونه های مثبت یا منفی داشته باشد. آستانه های مختلف میتوانند تعداد نمونه هایی که به درستی تشخیص داده میشوند را تغییر دهند.

بنابراین، انتخاب آستانه یک جزئی مهم از فرآیند طبقه‌بندی با پرسپترون است و میتواند تأثیر بزرگی بر نتایج نهایی داشته باشد. انتخاب آستانه بهترین کاری است که ممکن است نیاز به تنظیم و تجربه داشته باشد.

در این سوال میبینیم که تفکیک کننده ما به خوبی کار میکند و در هر دو آستانه مدنظر ما، نتیجه به درستی نمایش داده شده و هیچ داده نادرستی نداریم:

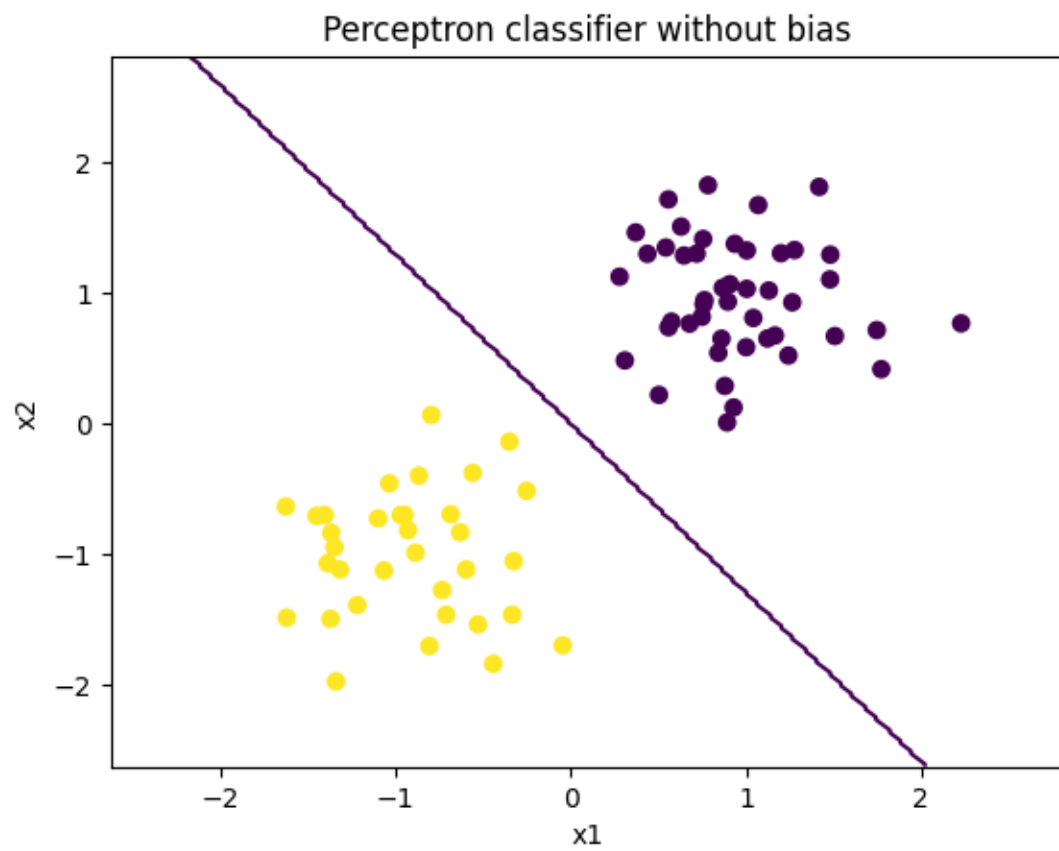


قسمت 4 : حذف بایاس:

```
perceptron_model_without_bias = Perceptron(fit_intercept=False,  
random_state=23, tol=0.001, max_iter=1000, eta0=0.1, verbose=0, n_jobs=-1)  
perceptron_model_without_bias.fit(X_train, y_train)  
test_accuracy_2 = perceptron_model_without_bias.score(X_test, y_test)  
  
test_accuracy_2  
  
ym3 = perceptron_model_without_bias.predict(xm)  
  
plt.contour(x1m, x2m, ym3.reshape(x1m.shape), levels=[0])  
plt.scatter(X_test_array[:, 0], X_test_array[:, 1], c=y_test)
```



```
plt.xlabel('x1')
plt.ylabel('x2')
plt.title('Perc
```



مشاهده میشود داده ها به خوبی از هم جدا شده اند