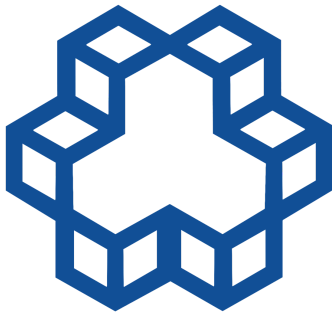


به نام خدا



نام و نام خانوادگی : علی اصغر قندی منور
شماره دانشجویی: ۹۸۲۴۴۲۳
درس: مبانی سیستم های هوشمند
استاد: دکتر مهدی علیاری
مینی پروژه ۱

در ابتدا، ما از کتابخانه `sklearn.datasets` برای تولید داده‌های مصنوعی و کتابخانه `matplotlib.pyplot` دومی برای نمایش داده‌ها در نمودار استفاده می‌کنیم.

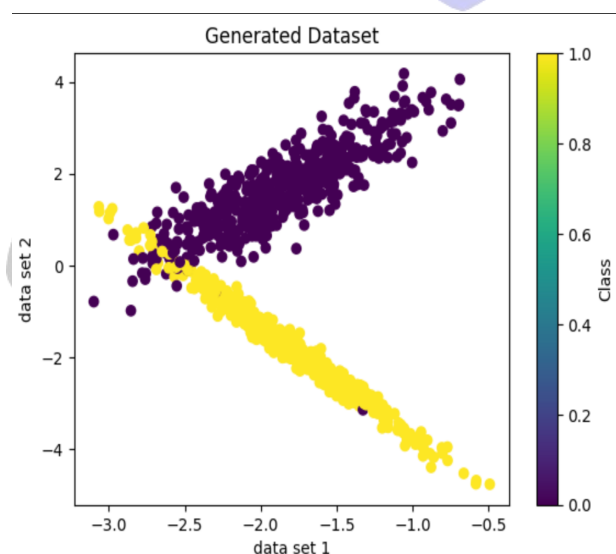
با استفاده از تابع `make_classification` از کتابخانه `sklearn.datasets`، یک مجموعه داده تصادفی تولید می‌کنیم. این مجموعه داده شامل 1000 نمونه (`n_samples=1000`) است که دارای 2 ویژگی (`n_features=2`) هستند. از این دو ویژگی، هر دو ویژگی معنادار هستند (`n_informative=2`) و هیچ ویژگی تکراری وجود ندارد (`n_redundant=0`).

هر کلاس تنها یک خوشه دارد (`n_clusters_per_class=1`) و تعداد کلاس‌ها دو عدد است (`n_classes=2`). همچنین فاصله بین دو کلاس (`class_sep`) برابر با 1.8 تعیین شده است.

تابع `make_classification` نیز با استفاده از یک عدد تصادفی `random_state=23` که دو رقم آخر شماره دانشجویی است داده‌ها را تولید و امکان تکرار پذیری نمونه‌ها را فراهم می‌کند.

سپس داده‌های تولید شده را در متغیرهای `X` و `y` ذخیره می‌کنیم. متغیر `X` حاوی ویژگی‌های داده‌ها و متغیر `y` حاوی برچسب‌های کلاس‌ها است.

با استفاده از تابع `plt.scatter` از کتابخانه `matplotlib.pyplot`، داده‌ها را در نمودار پراکندگی نمایش می‌دهیم.



```
from sklearn.datasets import make_classification
import matplotlib.pyplot as plt

X, y = make_classification(
    n_samples=1000,
    n_features=2,
    n_informative=2,
    n_redundant=0,
    n_clusters_per_class=1,
    n_classes=2,
    class_sep=1.8,
    random_state=23
)

plt.scatter(X[:, 0], X[:, 1], c=y)
plt.xlabel('data set 1')
plt.ylabel('data set 2')
plt.title('Generated Dataset')
plt.colorbar(label='Class')
plt.show()
```

```

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression, Perceptron, SGDClassifier

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

models = [
    LogisticRegression(random_state=23),
    Perceptron(random_state=23, penalty='l2'),
    SGDClassifier(random_state=23, loss='hinge')
]

def train_and_evaluate(model, X_train, X_test, y_train, y_test):
    model_name = type(model).__name__
    model.fit(X_train, y_train)
    model.predict(X_test)
    train_score = model.score(X_train, y_train)
    test_score = model.score(X_test, y_test)
    print(f"{model_name} train score is: {train_score}")
    print(f"{model_name} test score is: {test_score}\n")

for model in models:
    train_and_evaluate(model, X_train, X_test, y_train, y_test)

```

این کد نشان دهنده از آموزش و ارزیابی سه مدل مختلف یادگیری ماشین با استفاده از کتابخانه‌های scikit-learn است.

از کتابخانه‌های scikit-learn، اعم از train_test_split برای تقسیم داده به دو مجموعه آموزش و تست، و همچنین LogisticRegression، Perceptron و SGDClassifier برای تعریف مدل‌های یادگیری ماشین استفاده می‌شود.

داده‌ها که در متغیرهای X و y ذخیره شده‌اند، با استفاده از تابع train_test_split به دو مجموعه آموزش (X_train و y_train) و تست (X_test و y_test) تقسیم می‌شوند. در اینجا، 20٪ از داده‌ها به عنوان مجموعه تست اختصاص داده شده است.

سه مدل یادگیری ماشین، به نام‌های Logistic Regression، Perceptron و Stochastic Gradient Descent (SGD) Classifier تعریف شده‌اند و در یک لیست به نام models ذخیره می‌شوند.

تابع train_and_evaluate بر چسب‌های مربوط به آموزش و تست را می‌گیرد. این تابع مدل را آموزش می‌دهد، پیش‌بینی‌ها را بر روی داده‌های تست انجام می‌دهد، امتیاز دقت مدل را برای داده‌های آموزش و تست محاسبه می‌کند، و امتیازها را چاپ می‌کند.

در نهایت، یک حلقه for برای اعمال تابع train_and_evaluate به هر یک از مدل‌ها در لیست models استفاده می‌شود. این حلقه مدل‌ها را آموزش می‌دهد و امتیاز دقت آن‌ها را برای داده‌های آموزش و تست نمایش می‌دهد.

دانشگاه صنعتی خواجه نصیرالدین طوسی

```
LogisticRegression train score is: 0.96625
LogisticRegression test score is: 0.935

Perceptron train score is: 0.92875
Perceptron test score is: 0.92

SGDClassifier train score is: 0.9675
SGDClassifier test score is: 0.94
```

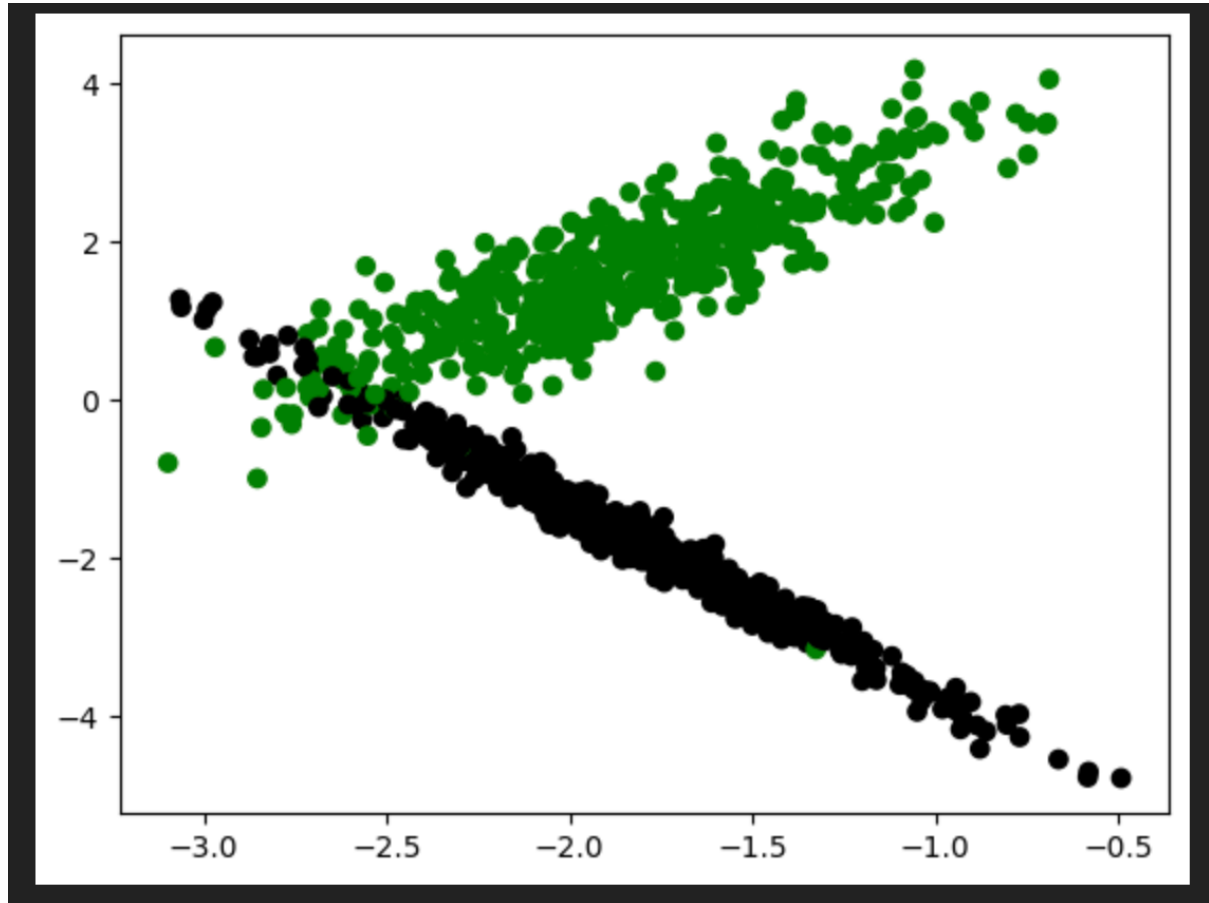
(1-3)

با استفاده از تابع `plt.scatter` از کتابخانه `matplotlib`، داده‌ها را در نمودار پراکنندگی نمایش می‌دهیم. همچنین با استفاده از پارامتر `[c=colors[y]`، رنگ‌های نقاط بر اساس برچسب‌های کلاس‌ها تعیین می‌شود. به عبارت دیگر، نقاط کلاس 0 با رنگ 'سبز' و نقاط کلاس 1 با رنگ 'سیاه' نمایش داده می‌شوند.

در نهایت، با دستور `plt.show()` نمودار نمایش داده می‌شود.

دانشگاه صنعتی خواجه نصیرالدین طوسی

```
colors = np.array(['green', 'black'])  
plt.scatter(X[:, 0], X[:, 1], c=colors[y])  
plt.show()
```



دانشگاه صنعتی خواجه نصیرالدین طوسی

(1-4)

```

from sklearn.linear_model import LogisticRegression
from mlxtend.plotting import plot_decision_regions
import matplotlib.pyplot as plt
import numpy as np

model1_3 = LogisticRegression()
model1_3.fit(X, y)

y_pred = model1_3.predict(X)
misclassified = np.where(y != y_pred)

plot_decision_regions(X, y, clf=model1_3, legend=2)
misclassified_points = X[misclassified]

ax = plt.gca()
ax.scatter(misclassified_points[:, 0], misclassified_points[:, 1],
          facecolors='none', edgecolors='r', marker='o', label='Misclassified')

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.legend()
plt.show()

```

این کد یک مدل رگرسیون لجستیک را آموزش داده و نمودار تصمیم برای دسته‌بندی داده‌ها را رسم می‌کند. همچنین نقاطی که توسط مدل اشتباه دسته‌بندی شده‌اند را با رنگ قرمز نمایش می‌دهد.

یک مدل رگرسیون لجستیک با نام `model1_3` تعریف و ایجاد می‌کنیم.

مدل با داده‌های `X` و `y` آموزش داده می‌شود.

پیش‌بینی‌های مدل بر روی داده‌ها انجام می‌شود و در متغیر `y_pred` ذخیره می‌شود.

با استفاده از تابع `np.where` نقاطی که توسط مدل اشتباه دسته‌بندی شده‌اند (تفاوت بین برچسب‌های واقعی و پیش‌بینی شده) در متغیر `misclassified` ذخیره می‌شود.

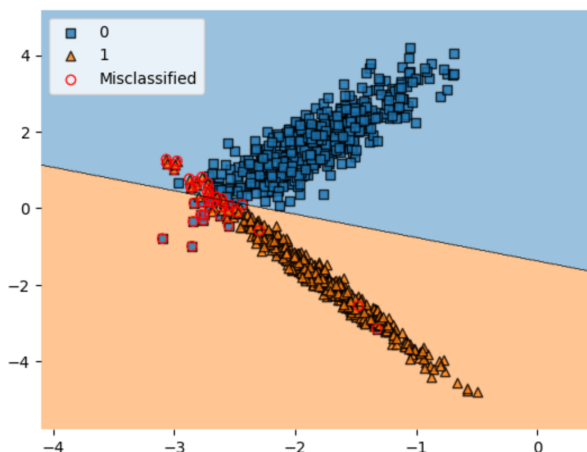
با استفاده از تابع `plot_decision_regions`، نمودار تصمیم برای دسته‌بندی داده‌ها ترسیم می‌شود و با پارامتر `clf=model1_3`، مدل رگرسیون لجستیک برای تصمیم‌گیری از آن استفاده می‌شود.

نقاطی که توسط مدل اشتباه دسته‌بندی شده‌اند، با استفاده از متغیر `misclassified_points` از داده‌های `X` استخراج می‌شوند. با استفاده از تابع `plt.gca()`، محورها و ستون‌های بالا و راست از نمودار حذف می‌شوند تا نمودار ساده‌تر و خواناتر باشد.

نقاط اشتباه دسته‌بندی شده با رنگ قرمز و با شکل دایره (`'marker='o'`) نمایش داده می‌شوند.

با استفاده از تابع `plt.legend()`، یک نمایش توضیحی به نمودار اضافه می‌شود تا بتوان نقاط اشتباه دسته‌بندی شده را از دیگر نقاط تشخیص داد.

در نهایت، با دستور `plt.show()` نمودار نمایش داده می‌شود.



```

from sklearn.linear_model import LogisticRegression
from mlxtend.plotting import plot_decision_regions
import matplotlib.pyplot as plt
import numpy as np

model1_3 = LogisticRegression()
model1_3.fit(X, y)

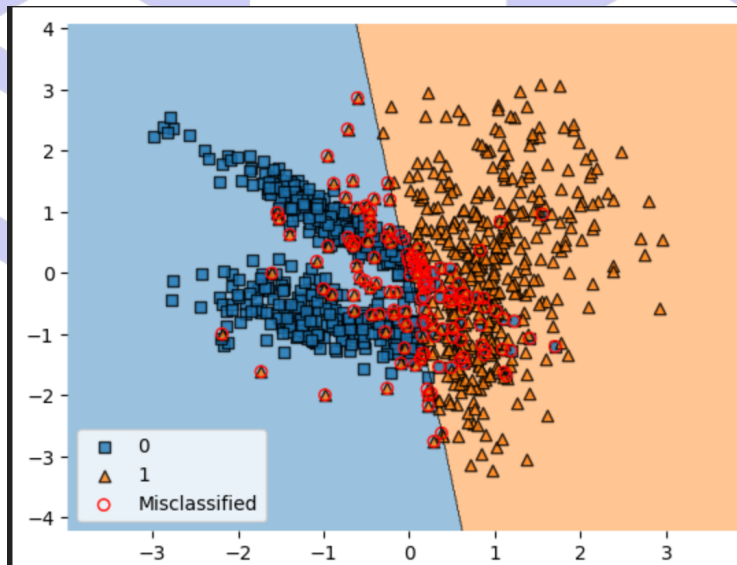
y_pred = model1_3.predict(X)
misclassified = np.where(y != y_pred)

plot_decision_regions(X, y, clf=model1_3, legend=2)
misclassified_points = X[misclassified]

ax = plt.gca()
ax.scatter(misclassified_points[:, 0], misclassified_points[:, 1],
           facecolors='none', edgecolors='r', marker='o', label='Misclassified')

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.legend()
plt.show()

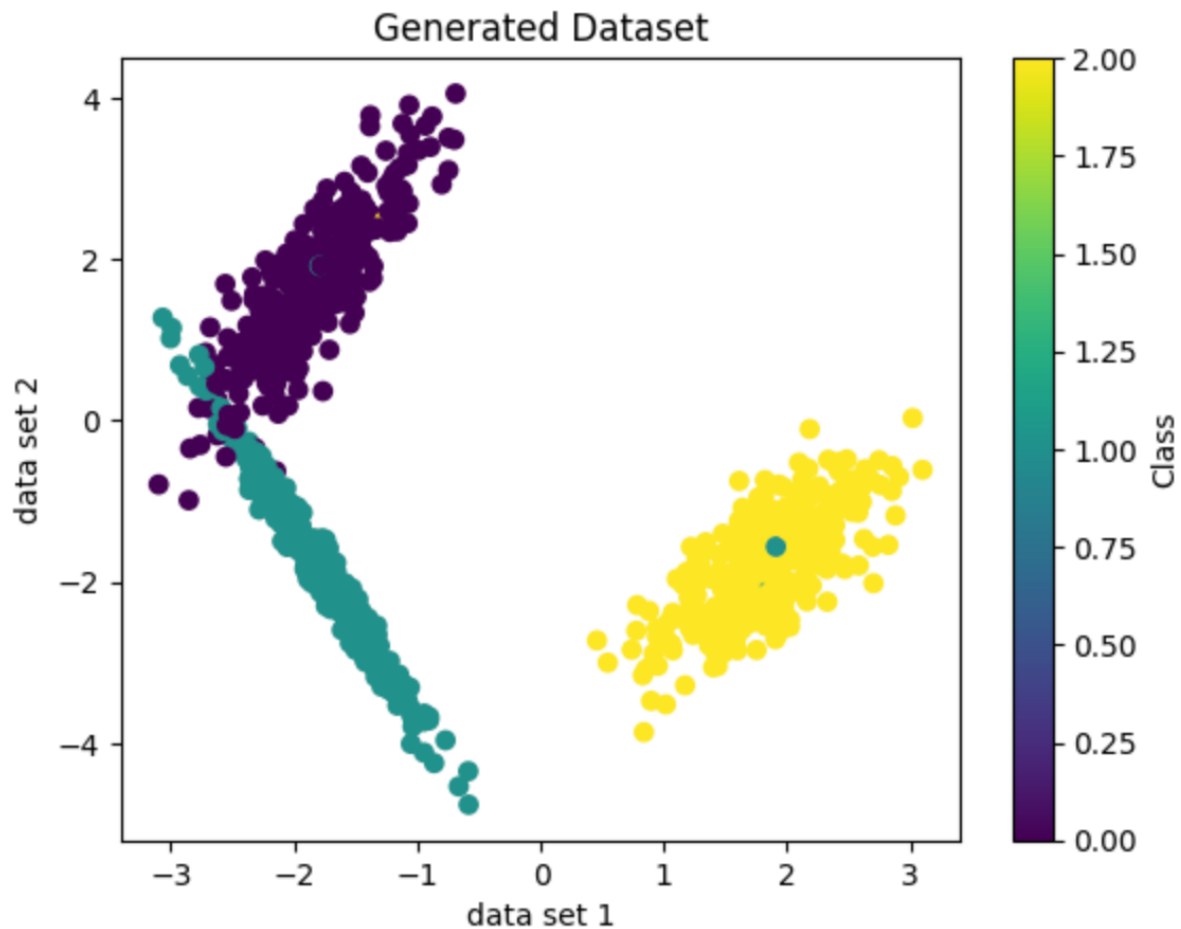
```



دانشگاه صنعتی خواجه نصیرالدین طوسی

```
X, y = make_classification(  
    n_samples=1000,  
    n_features=2,  
    n_informative=2,  
    n_redundant=0,  
    n_clusters_per_class=1,  
    n_classes=3,  
    class_sep=1.8,  
    random_state=23  
)  
  
plt.scatter(X[:, 0], X[:, 1], c=y)  
plt.xlabel('data set 1')  
plt.ylabel('data set 2')  
plt.title('Generated Dataset')  
plt.colorbar(label='Class')  
plt.show()
```

دانشگاه صنعتی خواجه نصیرالدین طوسی



```

from sklearn.model_selection import train_test_split
from mlxtend.plotting import plot_decision_regions
from sklearn.linear_model import LogisticRegression, Perceptron, SGDClassifier

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

models = [
    LogisticRegression(random_state=23),
    Perceptron(random_state=23, penalty='l2'),
    SGDClassifier(random_state=23, loss='hinge')
]

def train_and_evaluate(model, X_train, X_test, y_train, y_test):
    model_name = type(model).__name__
    model.fit(X_train, y_train)
    model.predict(X_test)
    train_score = model.score(X_train, y_train)
    test_score = model.score(X_test, y_test)
    print(f"{model_name} train score is: {train_score}")
    print(f"{model_name} test score is: {test_score}\n")

for model in models:
    train_and_evaluate(model, X_train, X_test, y_train, y_test)

```


LogisticRegression train score is: 0.97
LogisticRegression test score is: 0.975

Perceptron train score is: 0.955
Perceptron test score is: 0.945

SGDClassifier train score is: 0.9725
SGDClassifier test score is: 0.975

```
from sklearn.linear_model import LogisticRegression
from mlxtend.plotting import plot_decision_regions
import matplotlib.pyplot as plt
import numpy as np

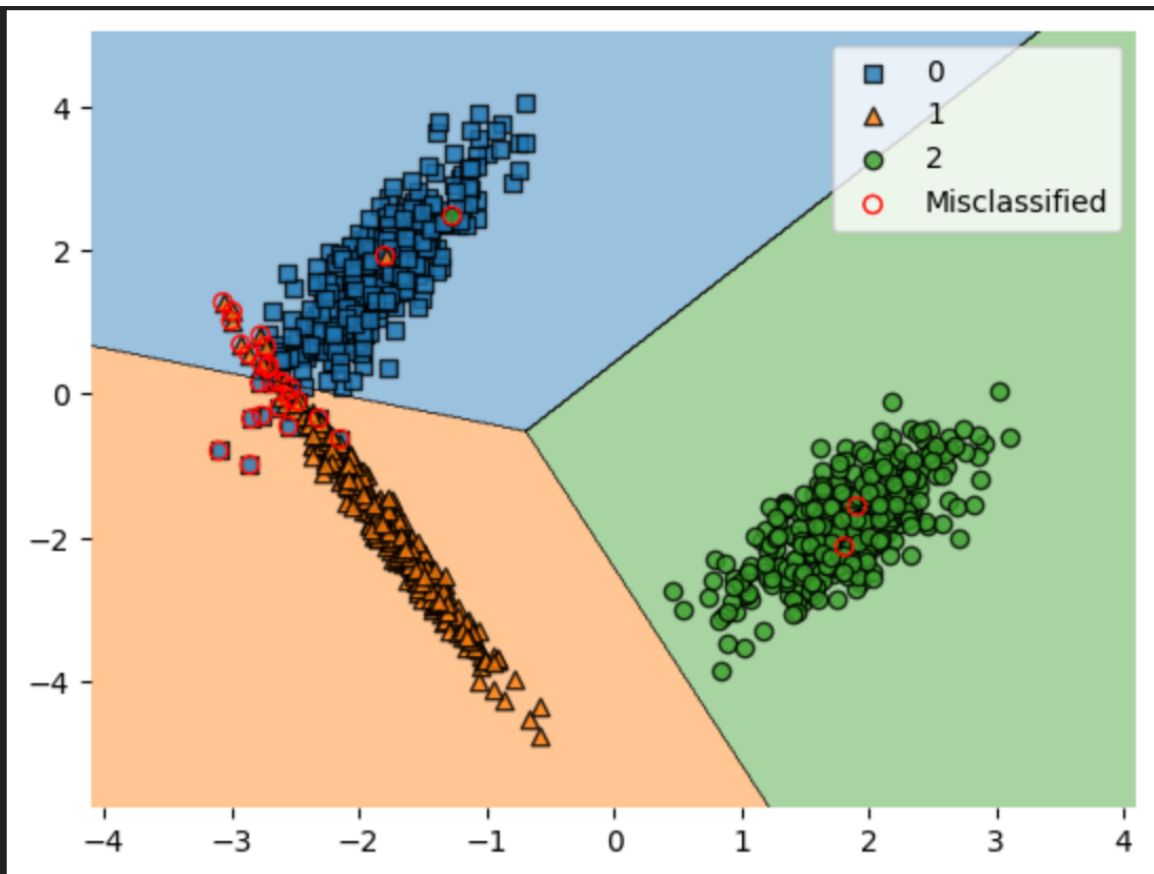
model1_3 = LogisticRegression()
model1_3.fit(X, y)

y_pred = model1_3.predict(X)
misclassified = np.where(y != y_pred)

plot_decision_regions(X, y, clf=model1_3, legend=2)
misclassified_points = X[misclassified]

ax = plt.gca()
ax.scatter(misclassified_points[:, 0], misclassified_points[:, 1],
           facecolors='none', edgecolors='r', marker='o', label='Misclassified')

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.legend()
plt.show()
```



```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import gdown
url = 'https://drive.google.com/file/d/1loB4ihlR60ej-6TRNyyBy8A94DmvRSG7'
output = 'data_banknote_authentication.txt'
gdown.download(url, output, quiet=False)

df = pd.read_csv("data_banknote_authentication.txt")

```

```

/usr/local/lib/python3.10/dist-packages/gdown/parse_url.py:35: UserWarning: You specified a Google Drive link that is not the correct link to download a file. You might want to try
warnings.warn(
Downloading...
From: https://drive.google.com/file/d/1loB4ihlR60ej-6TRNyyBy8A94DmvRSG7
To: /content/data_banknote_authentication.txt
6.86kB [00:00, 8.91MB/s]

```

```

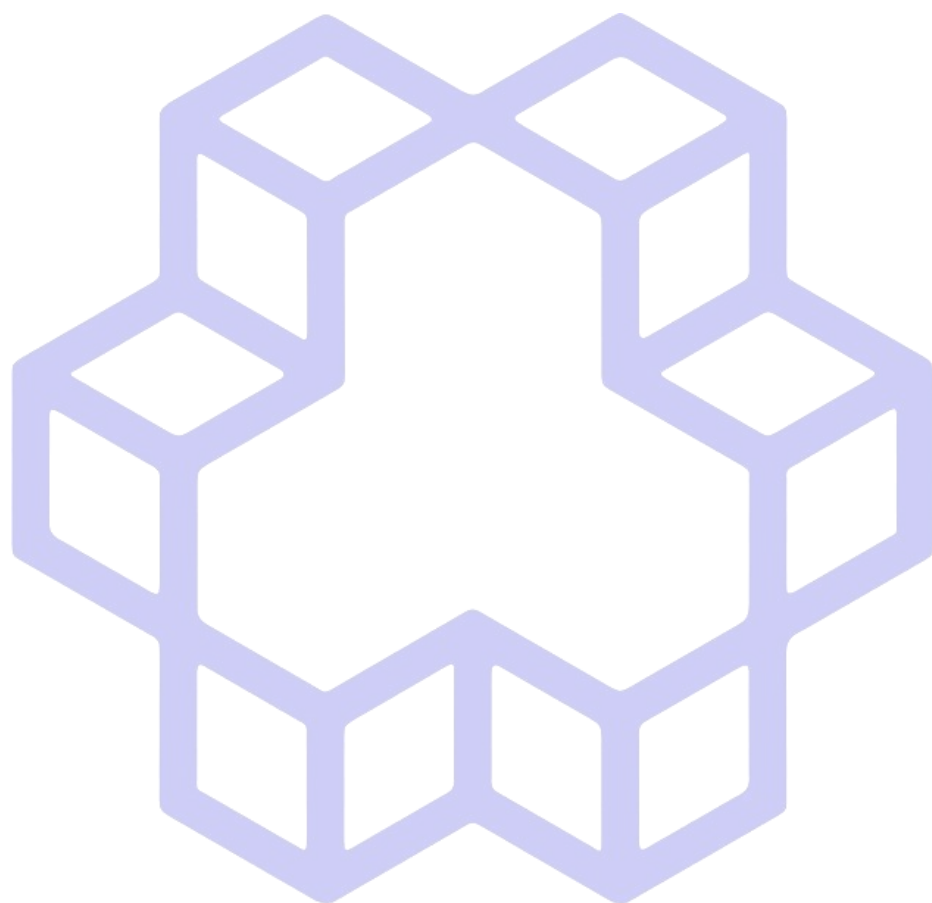
from sklearn.model_selection import train_test_split

df_shuffled = df.sample(frac=1, random_state=23).reset_index(drop=True)

X = df_shuffled.iloc[:, :-1].values
y = df_shuffled.iloc[:, -1].values.reshape(-1, 1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)

```



دانشگاه صنعتی خواجه نصیرالدین طوسی