**تمرین 1 (ساخت مدل کلاسبندی در Fisher Iris DataSet)**

در این تمرین، ما به بررسی مجموعه داده‌ی گل‌های زنبق (Fisher Iris) پرداختیم و با استفاده از ویژگی‌های چهارگانه‌ی این مجموعه داده['sepal length (cm) ','sepal width (cm)','petal length (cm)','petal width (cm)']، اقدام به دسته‌بندی آن در سه کلاس متفاوت کردیم ['setosa', 'versicolor', 'virginica']. مراحل کار به ترتیب در ادامه ارائه می‌گردد.

**1. ساخت DataFrame:**

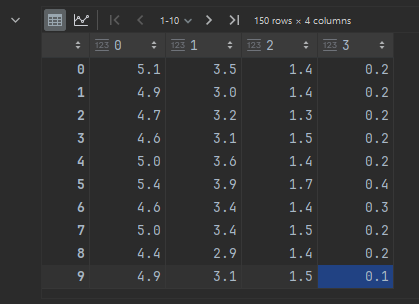
برای این منظور، ابتدا مجموعه‌ی داده را با استفاده از کتابخانه‌ی Pandas به یک DataFrame تبدیل کردیم. روش کار به صورت زیر می‌باشد:

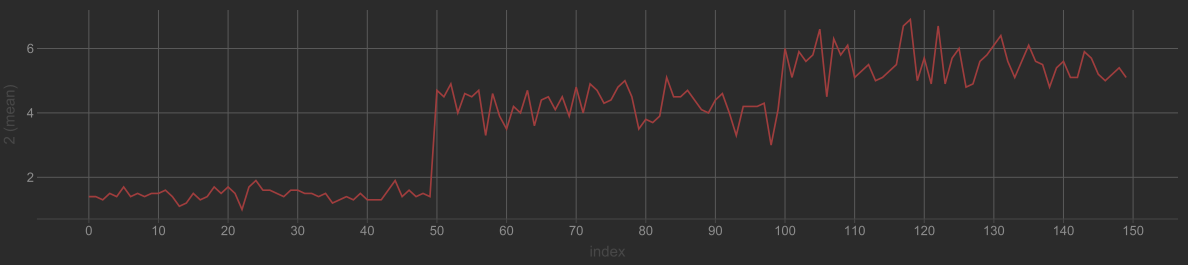
1. data = datasets.load\_iris()

2. data["data"]

3.

Output:





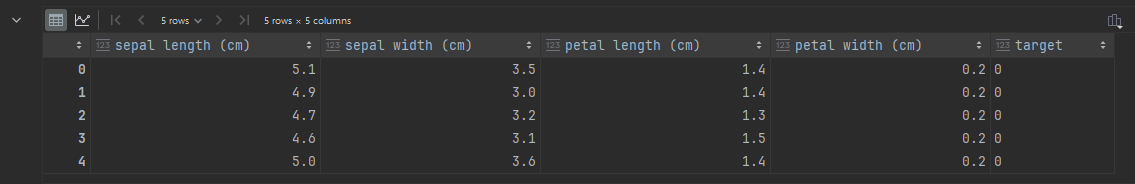
1. df = pd.DataFrame(data["data"] , columns = data["feature\_names"])

2. df["target"] = data["target"]

3. df.head()

4.

Output:



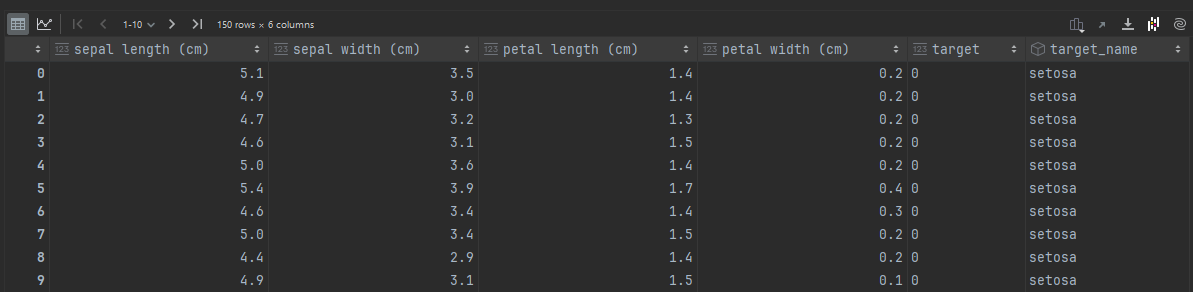
1. # Create New Column With the Species Name

2. df["target\_name"] = df["target"].map({0 : "setosa" , 1 : "versicolor" , 2 : "virginica"})

3. df

4.

Output:



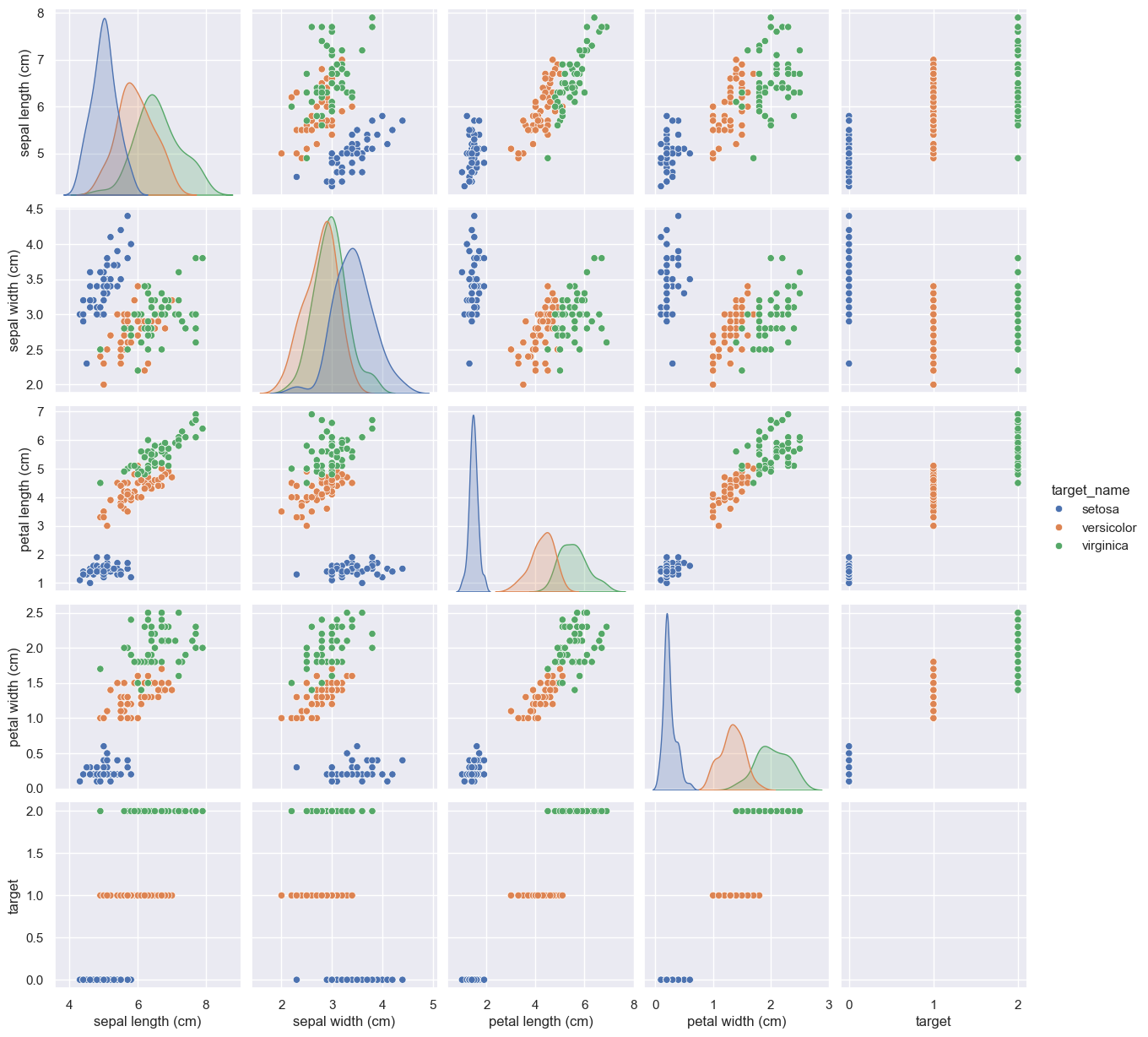
**2. رسم نمودار روابط:**

در این بخش، با استفاده از کتابخانه‌ی Seaborn اقدام به ترسیم نمودارهای تناسب بین ویژگی‌ها و کلاس‌ها نمودیم. روش کار به صورت زیر می‌باشد:

1. sns.pairplot(df , hue = "target\_name")

2.

Output:



**3. تقسیم بندی داده‌ها به دو بخش آموزش و آزمون (Train-Test Split):**

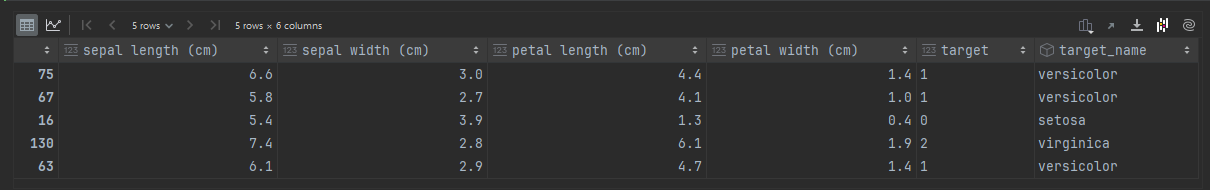
در این بخش، برای ساخت مدل مورد نظر، داده‌ها را به دو بخش آموزش و آزمون تقسیم می‌کنیم. روش کار به صورت زیر می‌باشد:

1. df\_train , df\_test = train\_test\_split(df , test\_size = 0.25)

2. df\_train.head()

3.

Output:



**4. آماده‌سازی مدل مورد نظر:**

در این بخش، ما مدل مورد نظرمان را آماده‌سازی می‌کنیم. روش کار به صورت زیر می‌باشد:

1. x\_train = df\_train.drop(columns = ["target" , "target\_name"]).values

2. y\_train = df\_train["target"].values

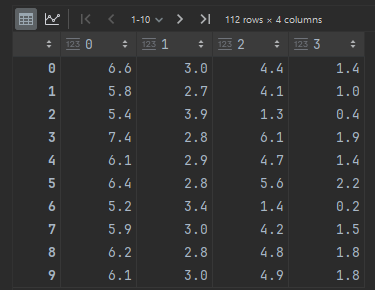
3.

در ادامه، با استفاده از دستورهای زیر، خروجی‌های x\_train و y\_train را در قالب جدول دریافت می‌کنیم:

1. x\_train

2.

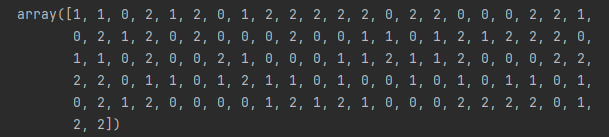
Output:



1. y\_train

2.

Output:



**5. ساخت یک مدل دستی:**

در این بخش، با تعریف یک تابع، یک مدل دستی پیش‌بینی را طراحی می‌کنیم. روش کار به صورت زیر می‌باشد:

1. def single\_feature\_predictions(petal\_length):  
2. '''Predicts The Iris Species Given The Petal Length'''

3. if petal\_length < 2.5:  
4. return 0

5. elif petal\_length < 4.8:  
6. return 1

7. else:  
8. return 2

9.

سپس این تابع را بر روی مدل آموزش‌دیده اجرا می‌کنیم:

1. manual\_y\_predictions = np.array([single\_feature\_predictions(val) for val in x\_train[: , 2]])

2.

با استفاده از دستور زیر، میزان دقّت مدل آموزش‌دیده را به دست می‌آوریم:

1. manual\_model\_accuracy = np.mean(manual\_y\_predictions == y\_train)

2. print(f'Manual Model Accuracy: {manual\_model\_accuracy \* 100 : .2f}%')

3.

Output:

Manual Model Accuracy: 96.43%

سپس با استفاده از دستورهای زیر، معیارهای ارزیابی را محاسبه می‌کنیم:

1. accuracy = accuracy\_score(y\_train , manual\_y\_predictions)  
2. precision = precision\_score(y\_train , manual\_y\_predictions , average = 'weighted')  
3. recall = recall\_score(y\_train , manual\_y\_predictions , average = 'weighted')  
4. f1 = f1\_score(y\_train , manual\_y\_predictions , average = 'weighted')

5.

1. report = classification\_report(y\_train , manual\_y\_predictions , target\_names =

2. data["target\_names"])  
3. print(f"Accuracy = {accuracy \* 100 : .2f}%")  
4. print(f"Precision = {precision \* 100 : .2f}%")  
5. print(f"Recall = {recall \* 100 : .2f}%")  
6. print(f"F1 - Score = {f1 \* 100 : .2f}%")  
7. print("\nClassification Report:\n" , report)

8.

Output:

Accuracy = 96.43%

Precision = 96.77%

Recall = 96.43%

F1 - Score = 96.40%

Classification Report:

precision recall f1-score support

setosa 1.00 1.00 1.00 41

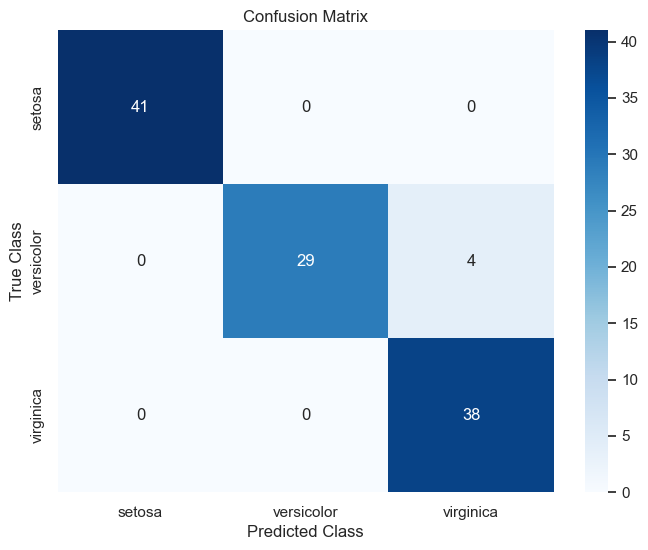
versicolor 1.00 0.88 0.94 33

virginica 0.90 1.00 0.95 38

accuracy 0.96 112

macro avg 0.97 0.96 0.96 112

weighted avg 0.97 0.96 0.96 112



**6. ساخت مدل Logistic Regression:**

در این بخش، ما یک مدل Logistic Regression طراحی کرده و سپس آن را ارزیابی می‌کنیم.

1. model = LogisticRegression(max\_iter = 200)

2. model.fit(x\_train , y\_train)

3.

1. model.score(x\_train , y\_train)

2.

Output:

0.9821428571428571

نتایج ارزیابی عبارت‌اند از:

Accuracy = 92.86%

Precision = 94.29%

Recall = 92.86%

F1 - Score = 92.90%

Classification Report:

precision recall f1-score support

setosa 1.00 1.00 1.00 9

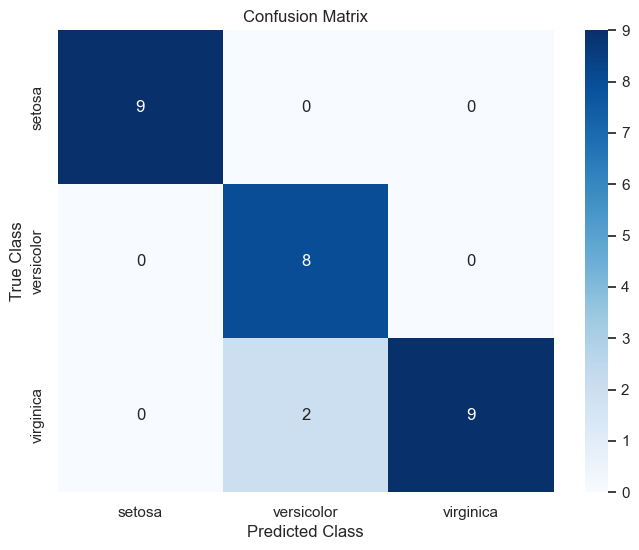
versicolor 0.80 1.00 0.89 8

virginica 1.00 0.82 0.90 11

accuracy 0.93 28

macro avg 0.93 0.94 0.93 28

weighted avg 0.94 0.93 0.93 28



نتایج ارزیابی نهایی عبارت‌اند از:

