

```
from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

!pip install sklearn
```

## ▼ KNN بکارگیری مدل

تعداد نمونه ها کم و در حدود 150 نمونه با 4 خصوصیت است سه کلاس از گل زنیق داریم و مسئله سه کلاسه است

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, classification_report
import numpy as np
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

داده ها را می خوانیم

```
DF = load_iris()
Y = DF.target
X = DF.data
# split X_train , Y_train
# split X_test , Y_test
X_train, X_test, Y_train, Y_test = train_test_split(X, Y)
```

## ▼ بررسی داده های تقسیم شده آموزش و تست

1. داده ها در یک اندازه نیستند بعدا مقیاس شوند.
2. مقدار رنک و مقدار کواریانس ماتریس بررسی شود واریانس ها اگر ماتریس قطری باشد روی قطر قرار دارند و خصوصیت خوب را تعیین می کند رنک خوب با تعداد خصوصیات (4 تا) برابر است

```
# Scaling Data Just Train Data fit
# data are small set
dam_Scaler = StandardScaler()
X_train = dam_Scaler.fit_transform(X_train)
# note that test data must not fit
X_test = dam_Scaler.transform(X_test)

print(np.cov(X.T)) # Cov matrix
print('---Rank---')
print(np.linalg.matrix_rank(np.cov(X.T))) # Rank

[[ 0.68569351 -0.042434  1.27431544  0.51627069]
 [-0.042434  0.18997942 -0.32965638 -0.12163937]
 [ 1.27431544 -0.32965638  3.11627785  1.2956094 ]
 [ 0.51627069 -0.12163937  1.2956094  0.58100626]]
---Rank---
4
```

## ▼ دسته بندی داده ها

```
KNeighborsClassifier()

model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, Y_train)
Y_pred = model.predict(X_test)
```

## ▼ مقایسه Y\_pred , Y\_test

به نوعی میزان خطای تشخیص کلاس را از روی آرایه می‌توانیم ببینیم

```
print(Y_test)
print(Y_pred)
fail_Summary = 0
i=0
for y in Y_test :
    if y != Y_pred[i] :
        fail_Summary+=1
    i+=1
print(f' {fail_Summary} item were different')

[2 1 2 0 1 0 0 2 2 2 0 2 2 2 1 2 1 0 0 2 2 2 1 0 1 1 2 2 1 0 0 2 0 1 2 1 0
 1]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0]
0]
27 item were different
```

## ▼ مشاهده ماتریس تداخل

1. میزان دقت
2. ویژگی با واریانس بالاتر
3. بررسی قطری بودن یا نبودن ماتریس
4. تصمیم‌گیری برای مقیاس بندی فقط داده آموزش

```
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
```

```
[[11  0  0]
 [11  0  0]
 [16  0  0]]
```

	precision	recall	f1-score	support
0	0.29	1.00	0.45	11
1	0.00	0.00	0.00	11
2	0.00	0.00	0.00	16
accuracy			0.29	38
macro avg	0.10	0.33	0.15	38
weighted avg	0.08	0.29	0.13	38

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-c
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-c
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-c
_warn_prf(average, modifier, msg_start, len(result))
```