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

```
Mounted at /content/gdrive
```

```
!pip install sklearn
```

بکارگیری مدل KNN

تعداد نمونه ها کم و در حدود 150 نمونه با 4 خصوصیت است سه کلاس از گل زنبق داریم و مسئاله 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 (4تا) برابر است

```
# Scaling Data Just Train Data fit
# data are small set
dam_Scaler = StandardScaler()
#X_tarin=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

KneighborsClassifire()

```
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 می توانیم 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')
```

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

مشاهده ماتریس تداخل

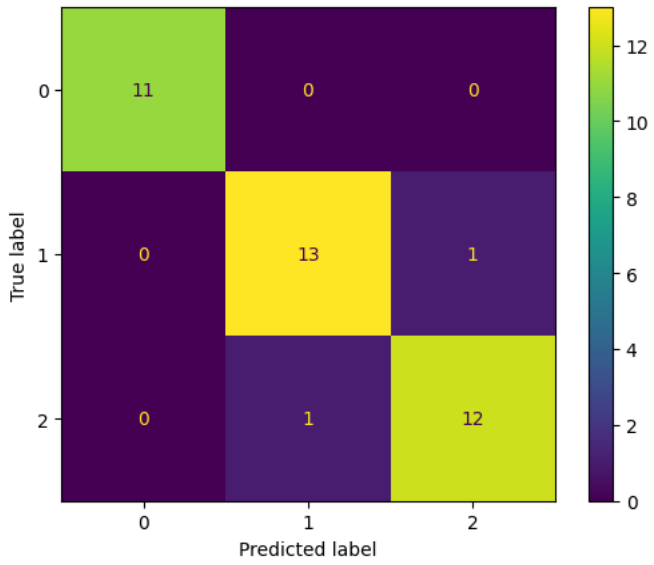
1. میزان دقت
2. ویژه گی با واریانس بالاتر
3. بررسی قطری بودن یا نبودن ماتریس
4. تصمیم گیری برای مقیاس بندی فقط داده آموزش

```
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
cm = confusion_matrix(Y_test, Y_pred)
cm_display = ConfusionMatrixDisplay(cm).plot()
```

```
[[11  0  0]
 [ 0 13  1]
 [ 0  1 12]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        11
           1       0.93      0.93      0.93        14
           2       0.92      0.92      0.92        13

    accuracy                           0.95        38
   macro avg       0.95      0.95      0.95        38
weighted avg       0.95      0.95      0.95        38
```



✓ 0s completed at 8:57 PM