به نام خدا

پاسخ سوالات تشریحی

تمرین سری دوم درس داده کاوی

جناب آقایان دکتر سعیدرضا خرد پیشه و هادی فراهانی

دانشجو: رضا جمشیدکیانی (۹۸۴۲۲۰۴۴)

فهرست مطالب

۱- ابتدا به بررسی دیتاست با استفاده از پکیج pandas بپردازید.

ابتدا پکیج های لازم را اضافه می کنیم

In [137]:
```
import numpy as np
import pandas as pd
from pandas import DataFrame
import matplotlib.pyplot as plt
from matplotlib import  style
style.use('ggplot')
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error , mean_absolute_error
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
```

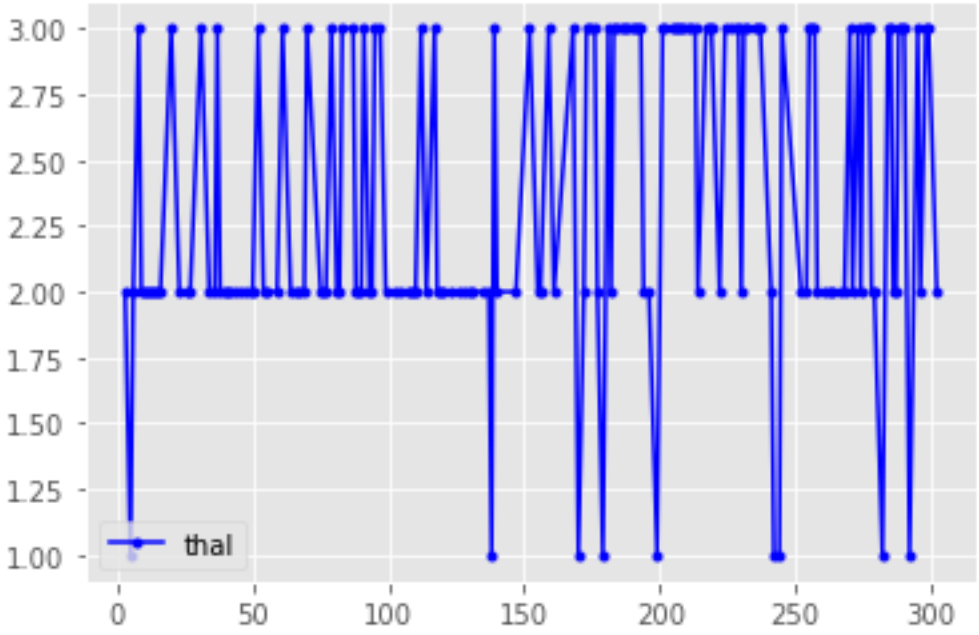
۱-۱ آیا داده پرت در دیتاست وجود دارد؟ در صورت وجود آن ها را حذف کنید.

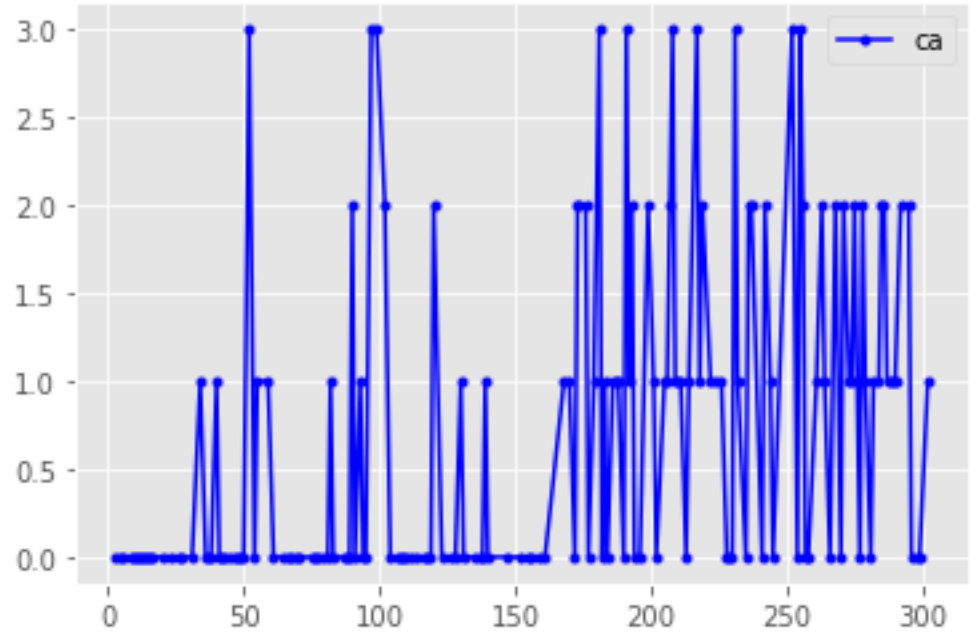
در این بخش دیتاها را پاکسازی می کنیم. برای این کار یک تابع نوشتیم که به صورت دستی میتواند فیلد های مارا پاکسازی نماید همینطور در خط اخر به دنبال داده های نال گشتیم که در این دیتاست چیزی یافت نشد.

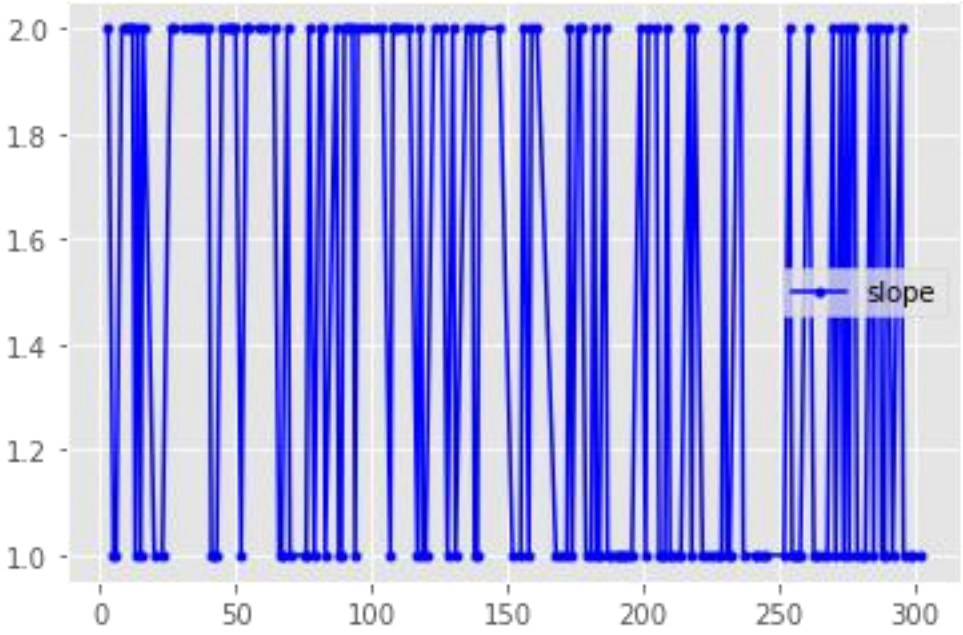
In [138]:
```
df=pd.read_csv('/home/kiani/Documents/python/dm/heart.csv')


def outlier_fixing(dff,tar,minN,maxN):
    dff=dff[dff[tar]<=maxN]
    dff=dff[dff[tar]>=minN]
    ax1=plt.subplot(1,1,1)
    dff[tar].plot(label=tar,color="blue",marker='o')
    plt.legend()
    plt.show()
    return dff


df1=outlier_fixing(df,'age',45,65)
df2=outlier_fixing(df1,'cp',0,3)
df3=outlier_fixing(df2,'trestbps',100,180)
df4=outlier_fixing(df3,'chol',150,350)
df5=outlier_fixing(df4,'thalach',100,180)
df6=(outlier_fixing(df5,'oldpeak',0,4))
df7=(outlier_fixing(df6,'slope',1,2))
df8=outlier_fixing(df7,'ca',0,3)
df9=outlier_fixing(df8,'thal',1,3)
print(len(df9))
print(df.isnull().sum()) #data has no null doc
```

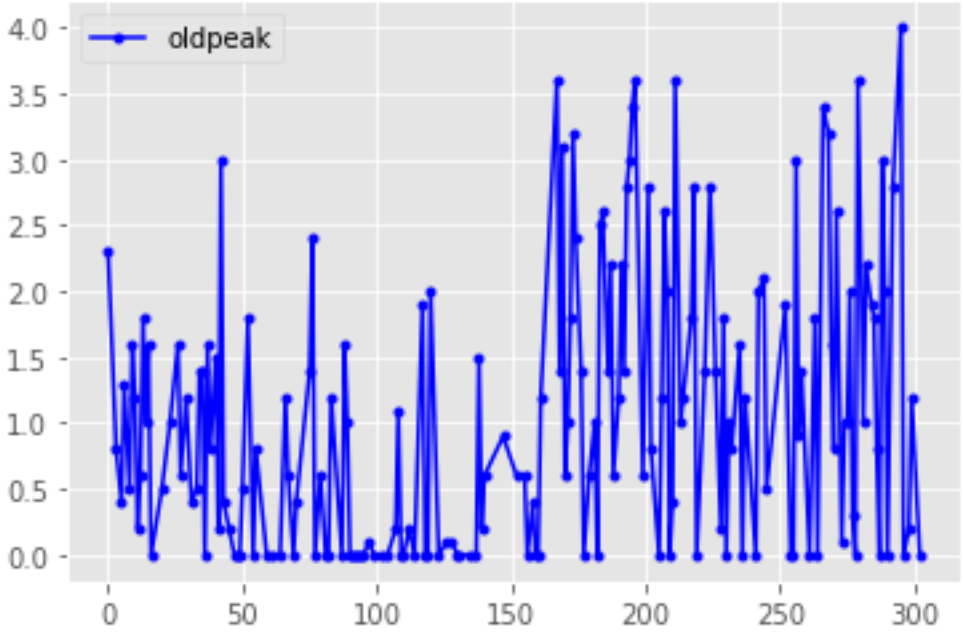
۱-۲ بررسی کنید آیا تعداد نمونه ها در هر کلاس متوازن است؟

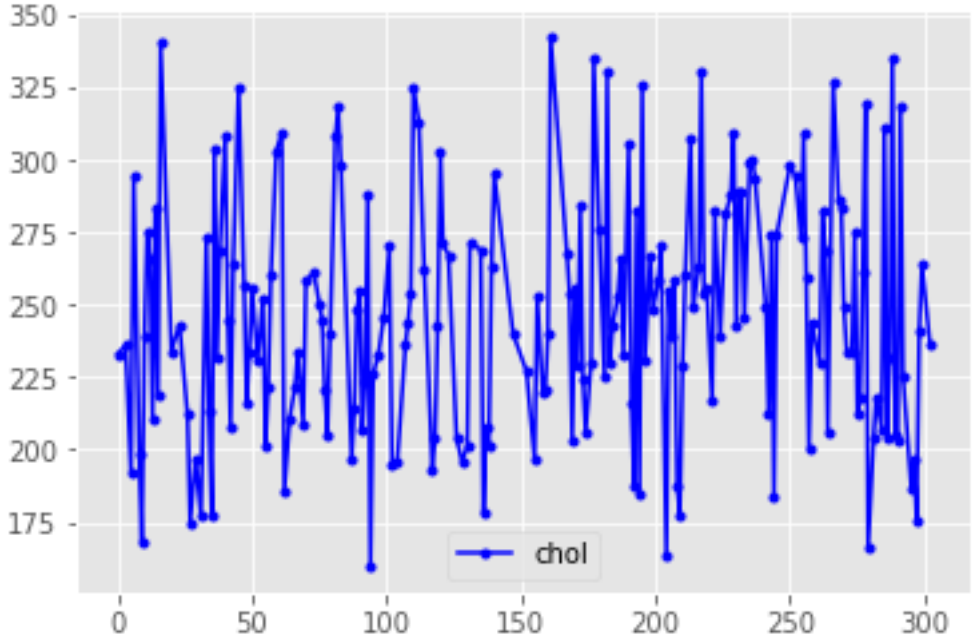
بله کلاس ها دارای توازن می باشد. الگوریتم های طبقه بندی میل به سمت کلاس با برچسب اکثریت دارند درحالی که دادهای حداقلی با درصد کوچکی هم موجود است. درمثال ما چنین پدیده ای ممکن است منجر به خطای پزشکی و هزینه های سنگین ناشی از خطا شود.

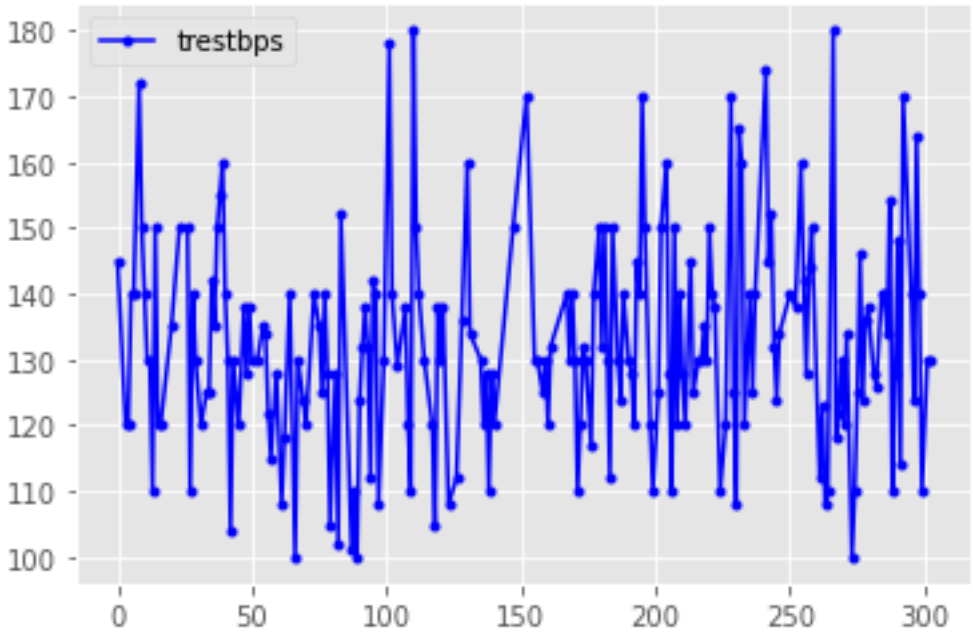

3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
thal
0
50
100
150
200
250
300

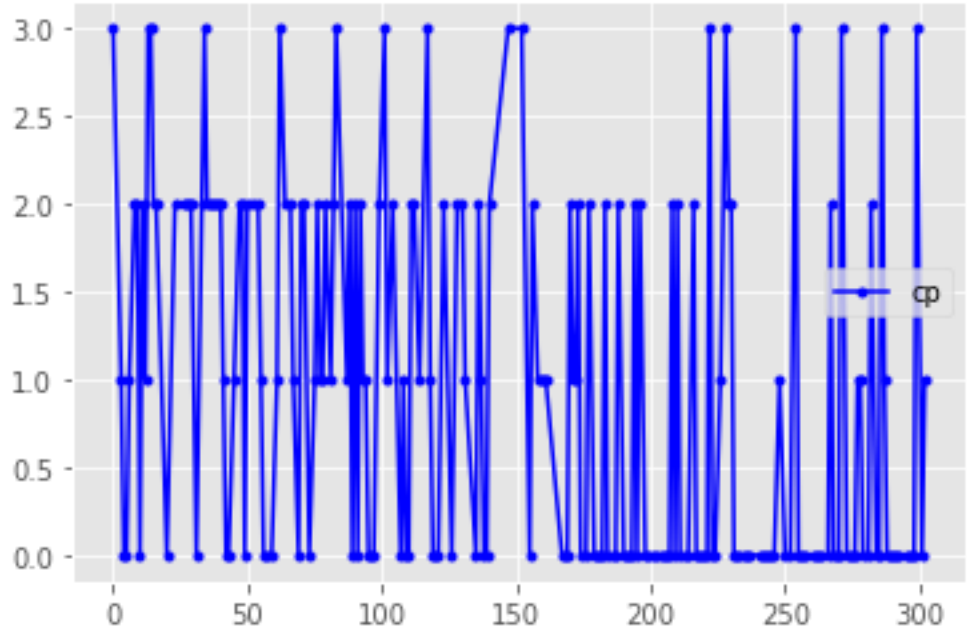

3.0
2.5
2.0
1.5
1.0
0.5
0.0
ca
0
50
100
150
200
250
300

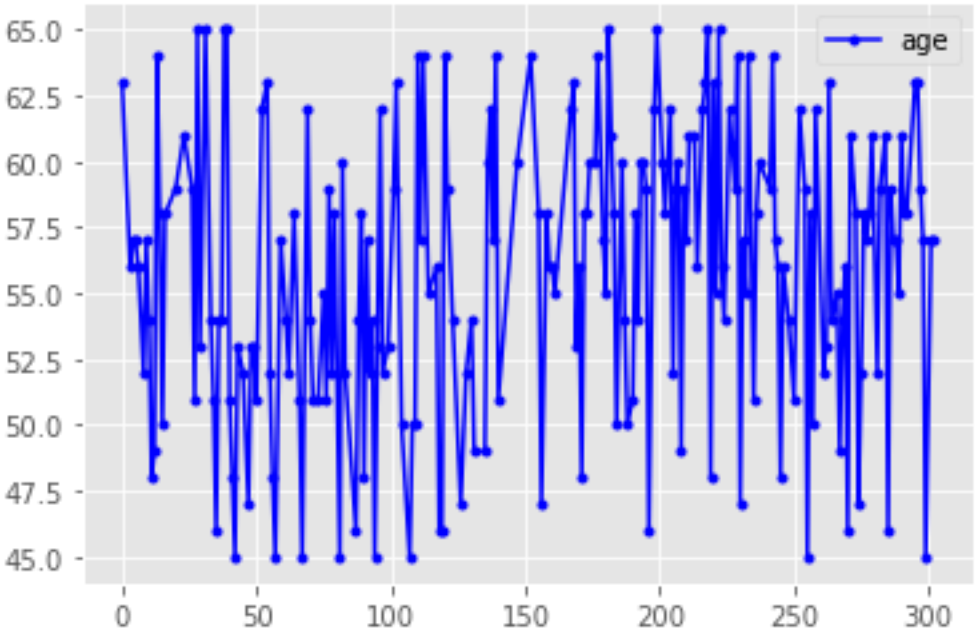

2.0
1.8
1.6
1.4
1.2
1.0
0
50
100
150
200
250
300
slope


4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
50
100
150
200
250
300
oldpeak


350
325
300
275
250
225
200
175
chol
0
50
100
150
200
250
300


180
170
160
150
140
130
120
110
100
trestbps
0
50
100
150
200
250
300





۱- نمونه های موجود در دیتاست را با نسبت ۸۰ به ۲۰ به دو بخش داده های آموزشی و داده های تست تقسیم بندی کنید . برای این کار میتوانید از پکیج sklearn بهره ببرید

در این بخش میتوانیم داده های خود را به دو قسمت تست و ترین تقسیم کنیم. یکبار برای کل دیتا دست و یکبار برای فیچرهای خاص نوشته ایم که به صورت کاستوم میتوان ازهرکدام استفاده کرد.

```
In [163]: #splitting Data
          X = df.drop(['target'] , axis=1).values
          Y = df['target'].values
          x_train , x_test , y_train , y_test = train_test_split(X,Y , test_size=0.20 ,random_state=40 )

In [221]: #for custom data
          X = df.drop(['target','age','sex','fbs','restecg','oldpeak','slope','ca','thal'],axis=1)
          Y = df['target'].values
          x_train , x_test , y_train , y_test = train_test_split(X,Y , test_size=0.20 ,random_state=40 )

In [222]: from sklearn.preprocessing import MinMaxScaler

          s = MinMaxScaler()

          x_train_scaled = s.fit_transform(x_train)

          x_test_scaled = s.transform(x_test)
```

۲- قضیه بیز را بیان کنید.

اگر بخواهیم احتمال وقوع یک رخداد را در صورت وجود یک شرط (شرایط) خاص محاسبه کنیم از احتمال شرطی استفاده می کنیم. با در نظرگرفتن اینکه P(B)>0 (یعنی پیشامد B یک پیشامد محال نباشد)

$$P(A|B)=P(A\cap B)/P(B)$$

به این احتمال، احتمال شرطی می گوییم. (Conditional Probability)

اما اگر فضای نمونه را بر اساس رخداد یا عدم رخداد پیشامد B و B' افراز (تفکیک) کنیم، برای آنکه احتمال A را بدست آوریم، دو حالت وجود دارد. یا B رخ داده یا خیر (B')

حال اگر فضای نمونه توسط B1,B2,….Bj افراز شده باشد بطوری که هر کدام از این افرازها، یک پیشامد محال نباشند P(Bi)>0

در این صورت برای هر پیشامد A داریم:

$$P(B_j|A) = \frac{P(B_j)P(A|B_j)}{P(A)}$$

پس از جایگزاری رابطه احتمال شرطی و قانون ضرب احتمال، به رابطه زیر می رسیم:

$$P(B_j|A) = \frac{P(B_j)P(A|B_j)}{\sum_{i=1}^{n} P(B_i)P(A|B_i)}$$

نام این قضیه، بیز است. مثلا فرض کنید ۹۸٪ از تست های HIV درست است و از طرفی ۲٪ از جوابهای آزمایش منفی ، نادرست است و همچنین ۱٪ از جوابهای آزمایش مثبت نیز نادرست است . با علم به اینکه ۰.۵ درصد از مردم مبتلا هستند اگر جواب آزمایش یک فرد مثبت باشد احتمال مبتلا بودن ایشان چقدر است؟ برای حل اینگونه مسائل از قضیه بیز و رابطه فوق استفاده می کنیم:

$$P(HIV|Positive) = \underbrace{HIV}_{\text{اگر مثبت باشد احتمال}} = \frac{\overbrace{98\%}^{} \quad \overbrace{0.5\%}^{}}{P(Positive \mid HIV) * P(HIV)} = \frac{0.0049}{0.0149} = 32\%$$
$$P(Positive \mid HIV) * P(HIV) + P(Positive \mid HIV^c) * P(HIV^c)$$
$$98\% \quad 0.5\% \quad 1\% \quad 0.995\%$$

۳-۱ مقایسه دسته بند های Gaussian Naive Bayes و Multinomial Naive Bayes و Bernoulli Naive Bayes و کاربردهای آن ها

دسته بندی روش بیز در اغلب موارد به عنوان یک راهکار ساده برای دسته‌بندی و تعیین تشخیص برچسب اشیاء استفاده می گردد. اما برای بکارگیری دسته‌بندی بیز ساده، الگوریتم خاصی وجود ندارد، ولی در عوض خانواده‌ای از الگوریتم‌ها موجود است که با فرض استقلال ویژگی‌ها یا متغیرها نسبت به یکدیگر عمل می نمایند.

روش Gaussian Naive Bayes ساده ترین طبقه بندی Naive Bayes است که در آن باید داده‌ها از نوع پیوسته بوده و با این فرض که داده هر برچسب از یک توزیع ساده گاوسی گرفته شده است اما دسته بند بیز ساده چندجمله‌ای یا Multinomial Naive Bayes به عنوان یک دسته‌بندی متنی مورد استفاده است که برحسب مدل احتمالی یا توزیع چند جمله‌ای، برداری از ویژگی ها در نظر می گیرد که برای ویژگی هایی که ارائه دهنده اعداد گسسته هستند، مناسب تر است. ولی دسته بندی بیز ساده برنولی Bernoulli Naive Bayes که شبیه بیز ساده چند جمله ای است، اما فرضیات متغیرهای بولی هستند. در آن فرض می شود ویژگی ها دودویی باشند (صفر و یک) و به شکلی دسته‌بندی بیز را ایجاد می نماید که بیشترین کاربرد را در دسته‌بندی متن‌های کوتاه دارد، به همین دلیل محبوبیت و دارای کاربرد بیشتری می باشد. طبقه بندی متن، با مدل 'bag of words'، می تواند یک برنامه کاربردی از Naive Bayes برنولی باشد.

۵- پیاده سازی Bayes Naive Gaussian و آموزش آن بر روی داده های آموزشی (۸۰ درصد دیتاست) . نتایج را برای داده های تست (۲۰ درصد باقی دیتاست) بررسی کنید به عبارت دیگر برای داده ورودی بررسی کنید در بخش تست لیبل را پیش بینی کنید .با توجه به این لیبل های واقعی را نیز دارید معیار های زیر گزارش دهید.

ما در اینجا gnb خود را تست و سه داده های لازم را گزاش میکنیم. گزارش در تصویر آمده است.

In [18]:
```
# Gaussian Naive Bayes
from sklearn import datasets
from sklearn import metrics

expected = y_train
predicted = gnb.predict(x_train_scaled)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

```
              precision    recall  f1-score   support

           0       0.78      0.66      0.71       112
           1       0.74      0.84      0.79       130

    accuracy                           0.76       242
   macro avg       0.76      0.75      0.75       242
weighted avg       0.76      0.76      0.75       242

[[ 74  38]
 [ 21 109]]
```

٦- با استفاده از پکیج sklearn و GaussianNB یک مدل بسازید و بر روی داده های آموزشی ، ترین کنید سپس بر روی داده های تست همانند سوال ۵ سه معیار را گزارش دهید.

مدل ایجاد شده و بر روی داده های خود تست میکنیم.کد را میتوایند در تصویر زیر ببینید

In [223]:
```
gnb = GaussianNB()
gnb.fit(x_train_scaled,y_train)

GN_train_score = gnb.score(x_train_scaled,y_train)

GN_test_score = gnb.score(x_test_scaled,y_test)
#prediction
y_pred_GN=classifier.predict(x_test_scaled)
```

۸- کلاسیفایر SVM را با استفاده از پکیج sklearn بر سه فیچر مطرح شده در سوال (٤ (با استفاده از داده های آموزشی ترین کنید . سپس بر روی داده های تست سه معیار Recall ، score F۱ ، Precisionرا گزارش کنید.

In [23]:
```
from sklearn import metrics

expected = y_train
predicted = svm.predict(x_train_scaled)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

```
              precision    recall  f1-score   support

           0       0.83      0.68      0.75       112
           1       0.76      0.88      0.81       130

    accuracy                           0.79       242
   macro avg       0.79      0.78      0.78       242
weighted avg       0.79      0.79      0.78       242

[[ 76  36]
 [ 16 114]]
```

۹- حداقل دو حالت مختلف را برای کرنل در SVM ساخته شده با پکیج در نظر بگیرید و نتایج آن را گزارش دهید . آیا کرنل های مختلف نتایج مختلفی ارایه دادند ؟ به صورت کلی علت استفاده از کرنل ها در SVM چیست ؟ توضیح دهید.

دو نمونه svm با دو کرنل مختلف و نتایج ان را مشاهده میکنید. بله نتایج مختلفی ارایه شد که در تصویر مشاهده مینمایید.به طور کلی وظیفه کرنل ها در svm دریافت ورودی های مختلف و تبدیل ان به فرم مورد نیاز است .لگوریتم های مختلف SVM ، از انواع مختلف توابع کرنل استفاده می کنند. این توابع می توانند انواع متفاوتی داشته باشند. به عنوان مثال خطی ، غیرخطی ، چند جمله ای ، تابع پایه شعاعی (RBF) و سیگموئید.

In [22]:
```
svm = SVC(kernel="linear")
svm.fit(x_train_scaled,y_train)
y_pred_SVM=svm.predict(x_test_scaled)
#Training Score
Train_Score = svm.score(x_train_scaled,y_train)
#Test Score
test_score = svm.score(x_test_scaled,y_test)
#Conf_Matrix
cm=confusion_matrix(y_pred_SVM,y_test)
print(cm)
```

```
[[19  7]
 [ 7 28]]
```

In [21]:
```
svm = SVC(kernel="rbf")
svm.fit(x_train_scaled,y_train)
y_pred_SVM=svm.predict(x_test_scaled)
#Training Score
Train_Score = svm.score(x_train_scaled,y_train)
#Test Score
test_score = svm.score(x_test_scaled,y_test)
#Conf_Matrix
cm=confusion_matrix(y_pred_SVM,y_test)
print(cm)
```

```
[[17  4]
 [ 9 31]]
```

۱۰- دسته بند SVM را با استفاده از پکیج sklearn بسازید و با در نظر گرفتن کلیه فیچرهای دیتاست بر روی داده های آموزشی ترین کنید سپس نتایج را بر روی داده های تست، ارزیابی کنید.

سوال ۹ و ۱۰ تقریباً با هم انجام شد.

۱۵- تفاوت بین روش های کلاس بندی پارامتری و غیرپارامتری را به صورت خلاصه بیان کنید هر کدام بهتر است در چه مواقعی استفاده شوند ؟

کلاس بندی پارامتریک آنهایی هستند که فرضهایی را در مورد پارامترهای توزیع جمعیتی که نمونه از آنها گرفته می شود ، ارائه می دهند. این اغلب فرض است که داده های جمعیت به طور معمول توزیع می شود. آزمون های غیر پارامتری "بدون توزیع" هستند و به همین ترتیب می توانند برای متغیرهای غیر نرمال استفاده شوند.در یک مدل پارامتری ، تعداد پارامترها با توجه به اندازه نمونه ثابت می شود. در یک مدل غیر پارامتری ، تعداد (موثر) پارامترها می توانند با اندازه نمونه رشد کنند.

در یک رگرسیون OLS ، تعداد پارامترها همیشه به طول β خواهد بود به علاوه یک واریانس. یک شبکه عصبی با معماری ثابت و بدون تحلیل رفتن وزن یک مدل پارامتریک است. اما اگر تجزیه وزن دارید ، مقدار پارامتر پوسیدگی که با اعتبار سنجی متقابل انتخاب می شود ، با داده های بیشتر ، به طور کلی کوچکتر می شود. این می تواند به عنوان افزایش تعداد موثر پارامترها با افزایش اندازه نمونه تفسیر شود.

۱۶- معیار Matthews Correlation Coefficient(MCC) چیست و در چه جاهایی استفاده میشود.

ضریب همبستگی Matthews (MCC) یا ضریب phi در یادگیری ماشین به عنوان معیاری برای کیفیت طبقه بندی های باینری (دو کلاسه) ، که توسط برایان دبلیو ماتیوز ، بیوشیمی دان در سال ۱۹۷۵ معرفی شده است ، استفاده می شود

ضریب مثبت و منفی درست و نادرست را در نظر می گیرد و به طور کلی به عنوان یک معیار متعادل در نظر گرفته می شود سوال شانزده :که حتی اگر کلاس ها از اندازه های بسیار متفاوت باشند می تواند مورد استفاده قرار گیرد. در اصل ضریب

همبستگی بین طبقه بندی باینری مشاهده و پیش بینی شده است. مقداری بین −۱ و +۱ برمی گرداند. ضریب ۱ نشان دهنده یک پیش بینی کامل است ، ۰ چیزی بهتر از پیش بینی تصادفی نیست و −۱ نشان دهنده اختلاف نظر کلی بین پیش بینی و مشاهده است. با این حال ، اگر MCC برابر با −۱ ، ۰ یا ۱+ نباشد ، این یک شاخص قابل اعتماد نیست که یک پیش بینی شبیه حدس تصادفی است زیرا MCC به مجموعه داده وابسته است. MCC برای جدول احتمالی ۲ × ۲ با آمار مربع کای ارتباط نزدیک دارد