```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

```
In [2]: df=pd.read_csv(r'C:\Users\vansh\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Anacon
        da3 (64-bit)\heart_data.csv')
```

```
In [3]: df.isnull().sum()
```

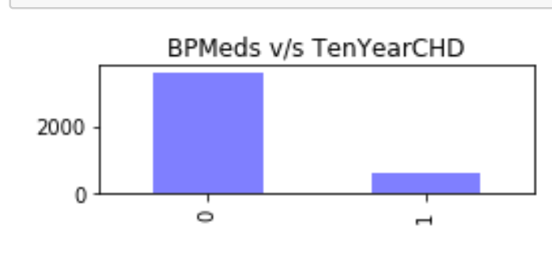
```
Out[3]: male                 0
        age                  0
        education          105
        currentSmoker        0
        cigsPerDay          29
        BPMeds              53
        prevalentStroke      0
        prevalentHyp         0
        diabetes             0
        totChol             50
        sysBP                0
        diaBP                0
        BMI                 19
        heartRate            1
        glucose            388
        TenYearCHD           0
        dtype: int64
```

```
In [5]: df.tail(5)
```

Out[5]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	di
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	
4238	1	40	3.0	0	0.0	0.0	0	1	0	185.0	141.0	
4239	0	39	3.0	1	30.0	0.0	0	0	0	196.0	133.0	

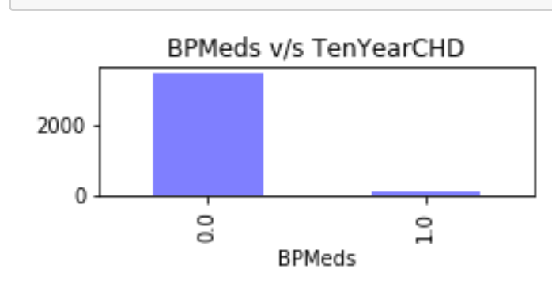
```
In [6]: plt.subplot2grid((3,3),(0,0),colspan=2)
        df.TenYearCHD.value_counts().plot(kind='bar',alpha=0.5,color='b')
        plt.title('BPMeds v/s TenYearCHD')
        plt.show()
```

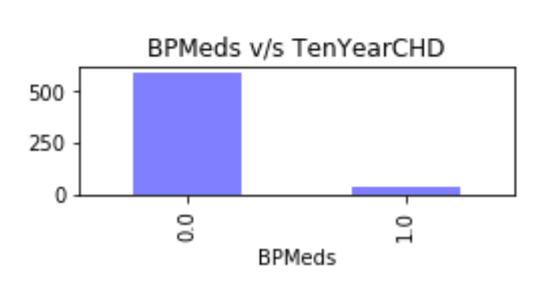


tenYearCHD = 1 means people who have heart risk.

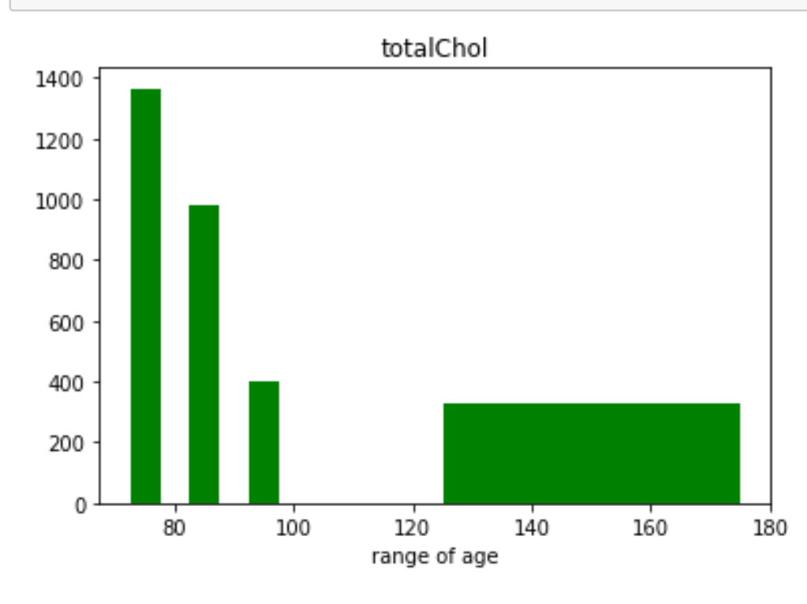
visualizing the relationship between BPMeds and TenYearCHD

```
In [7]: plt.subplot2grid((3,3),(0,0),colspan=2)
        df.BPMeds[df.TenYearCHD == 0].value_counts().plot(kind='bar',alpha=0.5,color='b')
        plt.title('BPMeds v/s TenYearCHD')
        plt.xlabel('BPMeds')
        plt.show()

        plt.subplot2grid((3,3),(1,0),colspan=2)
        df.BPMeds[df.TenYearCHD == 1].value_counts().plot(kind='bar',alpha=0.5,color='b')
        plt.title('BPMeds v/s TenYearCHD')
        plt.xlabel('BPMeds')
        plt.show()
```



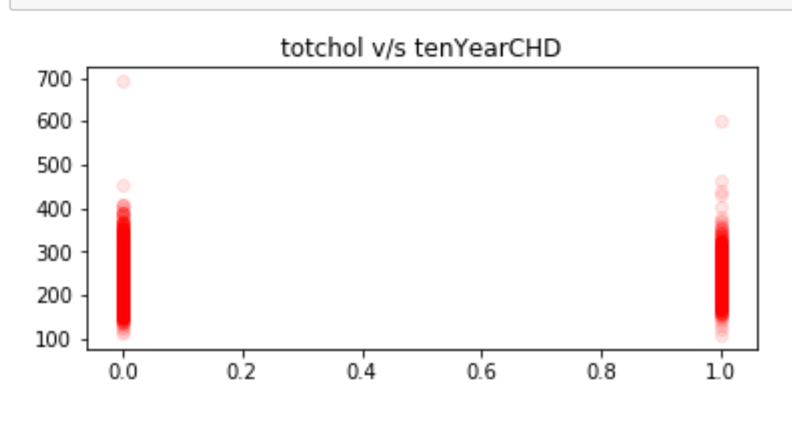


```
In [8]: bins=[70,80,90,100,200]
        plt.hist(df.glucose,bins,histtype='bar',rwidth=.5,color='g')
        plt.xlabel('range of age')
        plt.title('totalChol')
        plt.show()
```

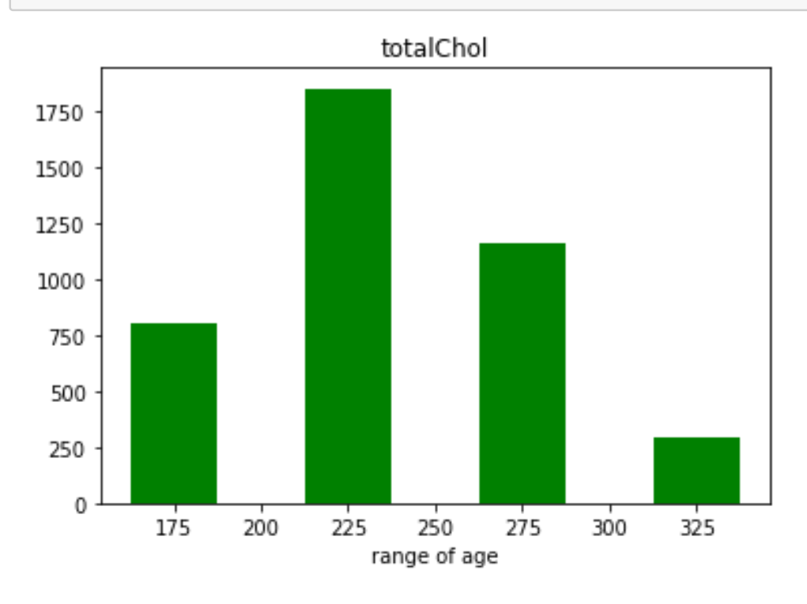


visualizing the relation between tatchol and tenYearCHD

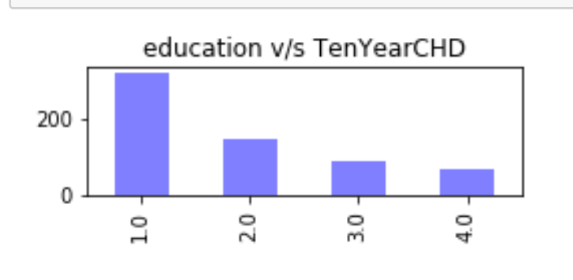
```
In [9]: plt.subplot2grid((3,3),(0,0),colspan=3,rowspan=2)
        plt.scatter(df.TenYearCHD,df.totChol,alpha=0.1,color='red')
        plt.title('totchol v/s tenYearCHD')
        plt.show()
```

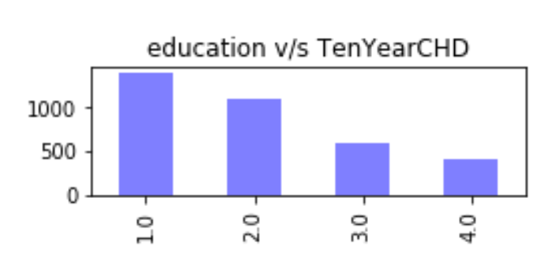


```
In [10]: bins=[150,200,250,300,350]
         plt.hist(df.totChol,bins,histtype='bar',rwidth=.5,color='g')
         plt.xlabel('range of age')
         plt.title('totalChol')
         plt.show()
```



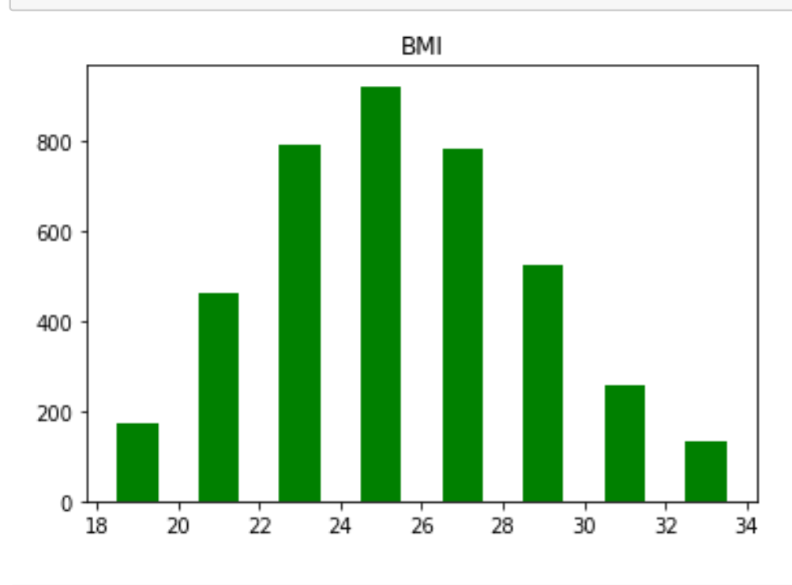
```
In [11]: plt.subplot2grid((3,3),(0,0),colspan=2)
         df.education[df.TenYearCHD == 1].value_counts().plot(kind='bar',alpha=0.5,color='b')
         plt.title('education v/s TenYearCHD')
         plt.show()

         plt.subplot2grid((3,3),(1,0),colspan=2)
         df.education[df.TenYearCHD == 0].value_counts().plot(kind='bar',alpha=0.5,color='b')
         plt.title('education v/s TenYearCHD')
         plt.show()
```





visualizing BMI

```
In [12]: bins=[18,20,22,24,26,28,30,32,34]
         plt.hist(df.BMI,bins,histtype='bar',rwidth=.5,color='g')
         plt.title('BMI')
         plt.show()
```



```
In [13]: x_train=df.drop('TenYearCHD',axis=1)
         y_train=df.iloc[:,15]
```

```
In [14]: y_train.head()
```

```
Out[14]: 0    0
         1    0
         2    0
         3    1
         4    0
         Name: TenYearCHD, dtype: int64
```

spliting the data into test and trainset

```
In [15]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(x_train,y_train,test_size=0.2,random_state=1)
```

```
In [16]: x_train.shape
```

```
Out[16]: (3392, 15)
```

```
In [17]: x_test.shape
```

```
Out[17]: (848, 15)
```

```
In [18]: x_train.isnull().sum()
```

```
Out[18]: male                 0
         age                  0
         education           82
         currentSmoker        0
         cigsPerDay          27
         BPMeds              36
         prevalentStroke      0
         prevalentHyp         0
         diabetes             0
         totChol             38
         sysBP                0
         diaBP                0
         BMI                 15
         heartRate            1
         glucose            305
         dtype: int64
```

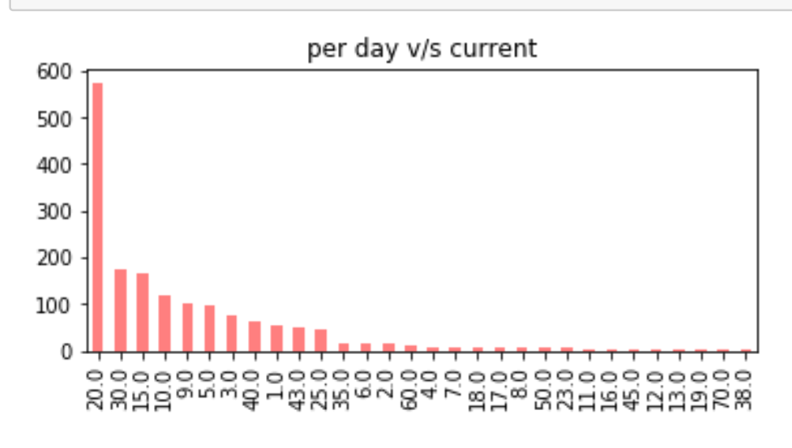
```
In [19]: x_train.head(5)
```

Out[19]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	di
2161	1	48	1.0	1	1.0	0.0	0	0	0	259.0	130.0	
133	0	56	2.0	1	20.0	0.0	0	0	0	246.0	128.0	
3124	0	51	3.0	0	0.0	0.0	0	1	0	198.0	142.5	
2473	0	66	1.0	1	1.0	1.0	0	1	0	261.0	154.0	
3879	1	40	4.0	1	40.0	0.0	0	0	0	334.0	120.0	

relationship between perdatcigs and currentday

```
In [20]: plt.subplot2grid((3,3),(0,0),colspan=3,rowspan=2)
         x_train.cigsPerDay[x_train.currentSmoker == 1].value_counts().plot(kind='bar',alpha=0.5,colo
         r='red')
         plt.title("per day v/s current")
         plt.show()
```



now in this below step ,there is filling the missing value of cigsPerDay field.

```
In [21]: x_train.fillna({'cigsPerDay':20.0},inplace=True)
         x_test.fillna({'cigsPerDay':20.0},inplace=True)
```

merge feature currentSmoker and cigsPerDay

```
In [22]: x_train['number_current_cigs']=x_train['currentSmoker']*x_train['cigsPerDay']
         x_test['number_current_cigs']=x_test['currentSmoker']*x_test['cigsPerDay']
```

drop feature currentSmoker and cigsPerDay

```
In [23]: x_train.drop('currentSmoker',axis=1,inplace=True)
         x_train.drop('cigsPerDay',axis=1,inplace=True)
         x_test.drop('currentSmoker',axis=1,inplace=True)
         x_test.drop('cigsPerDay',axis=1,inplace=True)
```

```
In [24]: x_train.shape
```

```
Out[24]: (3392, 14)
```

filling the missing values of BPMeds

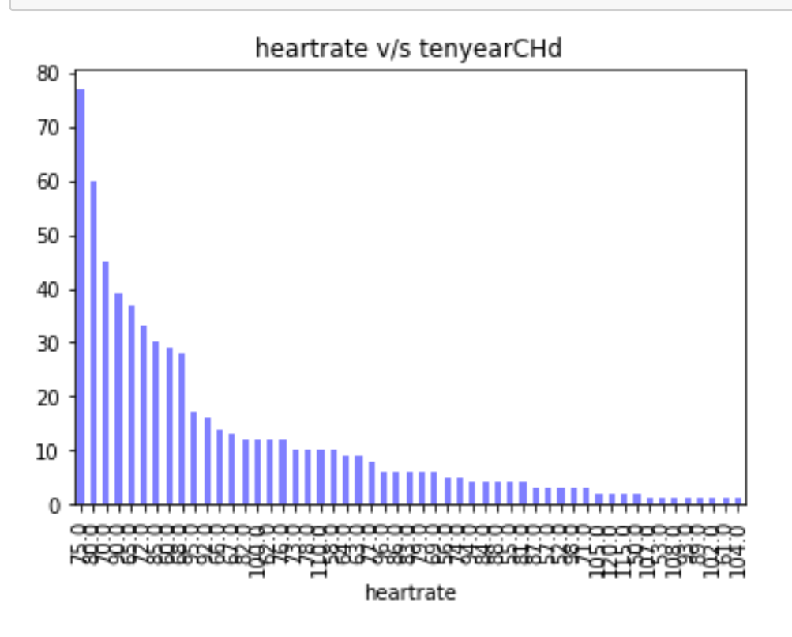
```
In [25]: x_train.fillna({'BPMeds':0.0},inplace=True)
         x_test.fillna({'BPMeds':0.0},inplace=True)
```

filling the missing value of totChol

```
In [26]: x_train.fillna({'totChol':225},inplace=True)
         x_test.fillna({'totChol':225},inplace=True)
```

visulaizing the relationship between heartrate and TenYearCHD

```
In [27]: plt.subplot2grid((6,6),(0,0),colspan=6,rowspan=6)
         df.heartRate[df.TenYearCHD == 1].value_counts().plot(kind='bar',alpha=0.5,color='b')
         plt.title('heartrate v/s tenyearCHd')
         plt.xlabel('heartrate')
         plt.show()
```



filling the missing value of heartRate field

```
In [28]: x_train.fillna({'heartRate':75.0},inplace=True)
         x_test.fillna({'heartRate':75.0},inplace=True)
```

```
In [29]: x_train.isnull().sum()
```

```
Out[29]: male                     0
         age                      0
         education               82
         BPMeds                   0
         prevalentStroke          0
         prevalentHyp             0
         diabetes                 0
         totChol                  0
         sysBP                    0
         diaBP                    0
         BMI                     15
         heartRate                0
         glucose                305
         number_current_cigs      0
         dtype: int64
```

filling the missing the value of glucose feature

```
In [30]: x_train.fillna({'glucose':225},inplace=True)
         x_test.fillna({'glucose':225},inplace=True)
```

filling the missing value of education field

```
In [31]: x_train.fillna({'education':1.0},inplace=True)
         x_test.fillna({'education':1.0},inplace=True)
```

filling the missing value of BMI

```
In [32]: x_train.fillna({'BMI':25.0},inplace=True)
         x_test.fillna({'BMI':25.0},inplace=True)
```

```
In [37]: from sklearn.svm import SVC
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.model_selection import cross_val_score
         from sklearn.model_selection import KFold
         from sklearn.linear_model import LogisticRegression
         from sklearn import preprocessing

         models=[]
         models.append(('knn',KNeighborsClassifier(n_neighbors=5)))
         models.append(('SVM',SVC(kernel='rbf',C=.1)))
         models.append(('log',LogisticRegression()))
         # cross validation
         result=[]
         for name,model in models:
             kf=KFold(n_splits=10)
             cv_result=cross_val_score(model,x_train,y_train,cv=kf)
             result.append(cv_result)
             msg="%s: %f :{%f}" %(name,cv_result.mean(),cv_result.std())
             print(msg)
         print(result)
```

```
knn: 0.833122 :{0.021175}
```

```
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value
of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled fe
atures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

```
SVM: 0.848458 :{0.018682}
```

```
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

```
log: 0.853172 :{0.021004}
[array([0.85588235, 0.86176471, 0.81120944, 0.84660767, 0.79646018,
       0.84955752, 0.82890855, 0.8259587 , 0.84660767, 0.80825959]), array([0.86176471, 0.864
70588, 0.83775811, 0.85250737, 0.82300885,
       0.87905605, 0.84070796, 0.820059  , 0.86725664, 0.83775811]), array([0.87941176, 0.867
64706, 0.83185841, 0.85250737, 0.8259587 ,
       0.89085546, 0.85250737, 0.8259587 , 0.86135693, 0.84365782])]
```

```
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

as we can see , logistic regression gave highest accuracy

```
In [39]: from sklearn.metrics import confusion_matrix
         model1=LogisticRegression()
         model1.fit(x_train,y_train)
         pred=model1.predict(x_test)
         cm=confusion_matrix(pred,y_test)
         print(cm)
```

```
[[716 125]
 [  2   5]]
```

```
E:\New folder (2)\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
ult solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```