In [5]:
```
import pandas as pd
```

In [9]:
```
df = pd.read_csv('iris.data')
df.head()
```

Out[9]:

	pl	pw	sl	sw	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
pl         0
pw         0
sl         0
sw         0
species    0
dtype: int64
```

In [11]:
```
x = df.drop(columns='species',axis=1)
```

In [13]:
```
y = df['species']
```

In [10]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

In [44]:
```
x_train , x_test, y_train, y_test = train_test_split(x,y,test_size=0.3, random_sta
```

In [45]:
```
model = LogisticRegression()
```

In [46]:
```
model.fit(x_train,y_train)
pred_y = model.predict(x_test)
print(pred_y)
```

```
['Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-virginica'
 'Iris-versicolor']

/home/student/anaconda3/lib/python3.9/site-packages/sklearn/linear_model/_logisti
c.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-le
arn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
(https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

In [2]:
```
from sklearn.metrics import accuracy_score
```

In [48]:
```
acc = accuracy_score(y_test,pred_y)
print(acc)
```

```
0.9777777777777777
```

In [49]:
```
model.score(x_test,y_test)
```

Out[49]:
```
0.9777777777777777
```

In [50]:
```
from sklearn.preprocessing import StandardScaler
```

In [51]:
```
#std = StandardScaler()
#x_test = std.fit_transform(x_test)
```

In [52]:
```
# y_test = std.fit_transform(y_test)
```

In [53]:
```
# gauzy(input = noramlly distributed) ,
# bernaulli(y/n.1/0) ,naivy(continuus) , multinomial naivy(categorical),
```

In [13]:
```
from sklearn.naive_bayes import GaussianNB
```

In [55]:
```
model1 = GaussianNB()
```

In [56]:
```
NBmodel = model1.fit(x_train,y_train)
y_pred1 = model1.predict(x_test)
```

In [57]:
```
acc1 = accuracy_score(y_test,y_pred1)
```

In [58]:
```
print(acc1)
```

```
0.9333333333333333
```

In [84]:
```
model1.predict([[5.1,3.5,1.4,0.2]])
```

```
/home/student/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarn
ing: X does not have valid feature names, but GaussianNB was fitted with feature
names
  warnings.warn(
```

Out[84]:
```
array(['Iris-setosa'], dtype='<U15')
```

In [63]:
```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
```

In [20]:
```
from sklearn.metrics import recall_score ,precision_score
```

In [72]:
```
print("recall Score : ", recall_score(y_test,pred_y,average='micro'))
```

```
recall Score :  0.9777777777777777
```

In [69]:
```
from sklearn.metrics import classification_report
```

In [70]:
```
print(classification_report(pred_y,y_test))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        14
Iris-versicolor       0.94      1.00      0.97        17
 Iris-virginica       1.00      0.93      0.96        14

       accuracy                           0.98        45
      macro avg       0.98      0.98      0.98        45
   weighted avg       0.98      0.98      0.98        45
```

In [77]:
```
new_data = [[5.5,3.5,1.2,0.2],[6.7,3.4,4.3,1.2],[6.2,2.6,5.8,1.8]]
new_peredict = NBmodel.predict(new_data)
print(new_peredict)
```

```
['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']

/home/student/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarn
ing: X does not have valid feature names, but GaussianNB was fitted with feature
names
  warnings.warn(
```

In [81]:
```
new_data = [[6.7,3.4,4.3,1.2]]
new_peredict = NBmodel.predict(new_data)
print(new_peredict)
```

```
['Iris-versicolor']

/home/student/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarn
ing: X does not have valid feature names, but GaussianNB was fitted with feature
names
  warnings.warn(
```

In [83]:
```
NBmodel.predict([[5.1,3.5,1.4,0.2]])
```

```
/home/student/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarn
ing: X does not have valid feature names, but GaussianNB was fitted with feature
names
  warnings.warn(
```

Out[83]:
```
array(['Iris-setosa'], dtype='<U15')
```

naivy based alogrithm for diabetes dataset

In [6]:
```
df = pd.read_csv('diabetes.csv')
df.head()
```

Out[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcom
0	6	148	72	35	0	33.6	0.627	50	
1	1	85	66	29	0	26.6	0.351	31	
2	8	183	64	0	0	23.3	0.672	32	
3	1	89	66	23	94	28.1	0.167	21	
4	0	137	40	35	168	43.1	2.288	33	

In [7]:
```
x = df.drop('Outcome',axis=1)
```

In [8]:
```
y = df['Outcome']
```

In [11]:
```
x_train , x_test, y_train, y_test = train_test_split(x,y,test_size=0.3,random_stat
```

In [14]:
```
model2 = GaussianNB()
```

In [15]:
```
dia_NBmodel = model2.fit(x_train,y_train)
```

In [16]:
```
pred_y3 = model2.predict(x_test)
```

In [17]:
```
predict_this_data = [[6,148,72,35,0,33.6,0.627,50],[2,85,63,22,0,26.6,0.351,29]]
predicted = dia_NBmodel.predict(predict_this_data)
print(predicted)
```

```
[1 0]

/home/student/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarn
ing: X does not have valid feature names, but GaussianNB was fitted with feature
names
  warnings.warn(
```

In [19]:
```
dia_acc = accuracy_score(y_test,pred_y3)
print(dia_acc)
```

```
0.7835497835497836
```

In [21]:
```
print("Diabetes Recall Score : ",recall_score(pred_y3,y_test,average='micro'))
```

```
Diabetes Recall Score :  0.7835497835497836
```

In [ ]:

In [ ]: