

LAB 03 Task - 2

In [3]:

```
from sklearn import preprocessing
import numpy as np
import pandas as pd
from sklearn.naive_bayes import GaussianNB, MultinomialNB
from sklearn import datasets
from sklearn.model_selection import train_test_split
```

Load the breast cancer wisconsin dataset

```
data = datasets.load_breast_cancer()
print(data)
```

[illegible]

```

===== \n
Max\n      ===== \n      Min
an):      radius (me
          6.981 28.11\n      texture (mean):
9.71 39.28\n      perimeter (mean):      43.79 188.5\n      ar
ea (mean):      143.5 2501.0\n      smoothness (mean):
0.053 0.163\n      compactness (mean):      0.019 0.345\n      co
ncavity (mean):      0.0 0.427\n      concave points (mean):
0.0 0.201\n      symmetry (mean):      0.106 0.304\n      fr
actal dimension (mean):      0.05 0.097\n      radius (standard erro
r):      0.112 2.873\n      texture (standard error):      0.3
6 4.885\n      perimeter (standard error):      0.757 21.98\n      area
(standard error):      6.802 542.2\n      smoothness (standard erro
r):      0.002 0.031\n      compactness (standard error):      0.002
0.135\n      concavity (standard error):      0.0 0.396\n      concave p
oints (standard error):      0.0 0.053\n      symmetry (standard error):
0.008 0.079\n      fractal dimension (standard error):      0.001 0.03\n      rad
ius (worst):      7.93 36.04\n      texture (worst):
12.02 49.54\n      perimeter (worst):      50.41 251.2\n      ar
ea (worst):      185.2 4254.0\n      smoothness (worst):
0.071 0.223\n      compactness (worst):      0.027 1.058\n      co
ncavity (worst):      0.0 1.252\n      concave points (wors
t):      0.0 0.291\n      symmetry (worst):      0.
156 0.664\n      fractal dimension (worst):      0.055 0.208\n      ====
===== \n\n      :Missing Attribute Va
lues: None\n\n      :Class Distribution: 212 - Malignant, 357 - Benign\n\n
:Creator: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian\n\n
:Donor: Nick Street\n\n      :Date: November, 1995\n\n\nThis is a copy of UCI ML
Breast Cancer Wisconsin (Diagnostic) datasets.\nhttps://goo.gl/U2Uwz2\n\nFea
tures are computed from a digitized image of a fine needle\naspirate (FNA) o
f a breast mass. They describe\ncharacteristics of the cell nuclei present
in the image.\n\nSeparating plane described above was obtained using\nMultis
urface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree\nConstruction Via
Linear Programming." Proceedings of the 4th\nMidwest Artificial Intelligence
and Cognitive Science Society,\npp. 97-101, 1992], a classification method w
hich uses linear\nprogramming to construct a decision tree. Relevant featur
es\nwere selected using an exhaustive search in the space of 1-4\nfeatures a
nd 1-3 separating planes.\n\nThe actual linear program used to obtain the se
parating plane\nin the 3-dimensional space is that described in:\n[K. P. Ben
nett and O. L. Mangasarian: "Robust Linear\nProgramming Discrimination of Tw
o Linearly Inseparable Sets",\nOptimization Methods and Software 1, 1992, 23
-34].\n\n\nThis database is also available through the UW CS ftp server:\n\nft
p ftp.cs.wisc.edu\ncd math-prog/cpo-dataset/machine-learn/WDBC/\n\n.. topi
c:: References\n\n      - W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nucle
ar feature extraction \n      for breast tumor diagnosis. IS&T/SPIE 1993 Inte
rnational Symposium on \n      Electronic Imaging: Science and Technology, vo
lume 1905, pages 861-870,\n      San Jose, CA, 1993.\n      - O.L. Mangasarian,
W.N. Street and W.H. Wolberg. Breast cancer diagnosis and \n      prognosis v
ia linear programming. Operations Research, 43(4), pages 570-577, \n      Jul
y-August 1995.\n      - W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machin
e learning techniques\n      to diagnose breast cancer from fine-needle aspir
ates. Cancer Letters 77 (1994) \n      163-171.', 'feature_names': array(['me
an radius', 'mean texture', 'mean perimeter', 'mean area',
'mean smoothness', 'mean compactness', 'mean concavity',
'mean concave points', 'mean symmetry', 'mean fractal dimension',
'radius error', 'texture error', 'perimeter error', 'area error',
'smoothness error', 'compactness error', 'concavity error',
'concave points error', 'symmetry error',
'fractal dimension error', 'worst radius', 'worst texture',
'worst perimeter', 'worst area', 'worst smoothness',
'worst compactness', 'worst concavity', 'worst concave points',
'worst symmetry', 'worst fractal dimension'], dtype='<U23'), 'filenam

```

```
e': 'c:\\users\\drash\\appdata\\local\\programs\\python\\python37\\lib\\site-packages\\sklearn\\datasets\\data\\breast_cancer.csv'}
```

In [7]:

```
data = datasets.load_breast_cancer()
feature = data.data
target = data.target
print("Feature:",feature)
print("Target:",target)
```

```
Feature: [[1.799e+01 1.038e+01 1.228e+02 ... 2.654e-01 4.601e-01 1.189e-01]
[2.057e+01 1.777e+01 1.329e+02 ... 1.860e-01 2.750e-01 8.902e-02]
[1.969e+01 2.125e+01 1.300e+02 ... 2.430e-01 3.613e-01 8.758e-02]
...
[1.660e+01 2.808e+01 1.083e+02 ... 1.418e-01 2.218e-01 7.820e-02]
[2.060e+01 2.933e+01 1.401e+02 ... 2.650e-01 4.087e-01 1.240e-01]
[7.760e+00 2.454e+01 4.792e+01 ... 0.000e+00 2.871e-01 7.039e-02]]
Target: [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0
1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 0 1 0 0
1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1 0 1 1 1
1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 1 0 1 1
1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 0 1 1 0 0 0 1 0
1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1
1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1
1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 1
1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0
0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 0 1 1 1 1
1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 0 1 1
0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 0 1 0 1 0 1 1 1 1 0 1 1 0 1 0 0
1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 0 1]
```

In [9]:

```
data_training,data_testing,target_training,target_testing = train_test_split(feature,target
```

In [11]:

```
gnb = MultinomialNB()
#Train the model using the training sets
gnb.fit(data_training, target_training)
#Predict the response for test dataset
target_pred = gnb.predict(data_testing)
```

In [15]:

```
#Accuracy
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(target_testing, target_pred))
```

Accuracy: 0.8982456140350877

In [16]:

```
#Precision and Recall
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score

precision = precision_score(target_testing, target_pred)
recall = recall_score(target_testing, target_pred)

print('precision: {}'.format(precision))
print('recall: {}'.format(recall))
```

```
precision: 0.8871794871794871
recall: 0.9611111111111111
```

In []: