WM_LAB_FAT_19BCE0521

June 8, 2021

1 Student Details

[1]:
```
#Student Names: Aman Anand
#Student Reg. No.: 19BCE0521
#Course and Slot: Web Mining - CSE3024 (L39+L40)
#Faculty: Shashank Mouli Satapathy
```

1.1 Imported Libraries

[2]:
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
```

1.2 Question 1

Consider the dataset given in the file Breast-Cancer.txt.

- Apply Multi-nomial Naïve-Bayes classifier using dataset given to prepare a model. Use Laplase estimator while evaluating.

- Test the accuracy of the model using the following test data.

*** (recurrence-events),50-59,premeno,25-29,0-2,no,1,left,left_low,no**

*** (recurrence-events),60-69,ge40,25-29,0-2,no,3,right,left_low,no**

*** (recurrence-events),50-59,premeno,15-19,0-2,no,2,left,left_low,no**

*** (recurrence-events),40-49,premeno,40-44,0-2,no,1,left,left_low,no**

*** (no-recurrence-events),50-59,ge40,35-39,0-2,no,2,left,left_low,no**

- Prepare confusion matrix and display.

- Evaluate the performance of the classifier using F-Measure, Specificity and sensitivity analysis.

[3]:
```
df1 = pd.read_csv("Breast-Cancer.txt")
df1.to_csv('BreastCancer.csv', index = None)
```

[4]:
```
df1
```

[4]:

	Class	Age	MenoPause	Tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiate
0	no-recurrence-events	30-39	premeno	30-34	0-2	no	3	left	left_low	no
1	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	right	right_up	no
2	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	left	left_low	no
3	no-recurrence-events	60-69	ge40	15-19	0-2	no	2	right	left_up	no
4	no-recurrence-events	40-49	premeno	0-4	0-2	no	2	right	right_low	no
..	…	…	…	…	…	…	…	…	…	…
276	recurrence-events	30-39	premeno	30-34	0-2	no	2	left	left_up	no
277	recurrence-events	30-39	premeno	20-24	0-2	no	3	left	left_up	yes
278	recurrence-events	60-69	ge40	20-24	0-2	no	1	right	left_up	no
279	recurrence-events	40-49	ge40	30-34	3-5	no	3	left	left_low	no
280	recurrence-events	50-59	ge40	30-34	3-5	no	3	left	left_low	no

[281 rows x 10 columns]

[5]:
```
#Creating feature(x) and target(y) attributes
X = df1.drop("irradiate", axis=1)
Y = df1["irradiate"]
```

[6]:
```
#Encoding X to numeric datatypes
le = LabelEncoder()

#Iterate over all the values of each column and extract their dtypes
for col in X.columns:
    #Compare if the dtype is object
```

```
    if X[col].dtype=='object':
        #Use LabelEncoder to do the numeric transformation
        X[col]=le.fit_transform(X[col])
```

[7]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.
↪3,random_state=1)
```

[8]:
```
#Naive Bayes

#Create a Multinomial Classifier
mnb = MultinomialNB()

#Train the model using the training sets
mnb.fit(X_train, Y_train)

#Predict the response for test dataset
Y_pred = mnb.predict(X_test)
print("\n------Multinomial Naive Bayes------\n")

# Predicted values
print("Predicted values:",Y_pred)

#Confusion Matrix
print("\nConfusion Matrix: \n\n",confusion_matrix(Y_test, Y_pred))

#Model Accuracy, how often is the classifier correct?
print("\nAccuracy:",metrics.accuracy_score(Y_test, Y_pred)*100)

# Report containing precision,recall,f1-score,support
print("\nReport--> \n",classification_report(Y_test, Y_pred))
```

```
------Multinomial Naive Bayes------

Predicted values: ['no' 'no' 'no' 'no' 'no' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no'
'no' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no'
 'no' 'no' 'no' 'yes' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no'
 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'yes'
 'no' 'no' 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'yes' 'no' 'no' 'no'
 'yes' 'no' 'no' 'no' 'no' 'no' 'no' 'no' 'yes' 'no' 'no' 'no' 'yes' 'no'
 'yes']

Confusion Matrix:

 [[60 10]
 [11  4]]
```

```
Accuracy: 75.29411764705883

Report-->
                precision    recall  f1-score   support

           no       0.85      0.86      0.85        70
          yes       0.29      0.27      0.28        15

    accuracy                           0.75        85
   macro avg       0.57      0.56      0.56        85
weighted avg       0.75      0.75      0.75        85
```

[9]:
```
cm = confusion_matrix(Y_test, Y_pred)
TP = cm[1][1]
TN = cm[0][0]
FP = cm[0][1]
FN = cm[1][0]

#Calculate Specificity
conf_sensitivity = (TP / float(TP + FN))

#Calculate Sensitivity
conf_specificity = (TN / float(TN + FP))
```

[10]:
```
print(f'Sensitivity: {round(conf_sensitivity,2)}')
print(f'Specificity: {round(conf_specificity,2)}')
```

```
Sensitivity: 0.27
Specificity: 0.86
```

[11]:
```
#Testing accuracy of developed model:
newdata =␣
 ↪[['recurrence-events','50-59','premeno','25-29','0-2','no',1,'left','left_low','no'],
          ␣
 ↪['recurrence-events','60-69','ge40','25-29','0-2','no',3,'right','left_low','no'],
          ␣
 ↪['recurrence-events','50-59','premeno','15-19','0-2','no',2,'left','left_low','no'],
          ␣
 ↪['recurrence-events','40-49','premeno','40-44','0-2','no',1,'left','left_low','no'],
          ␣
 ↪['no-recurrence-events','50-59','ge40','35-39','0-2','no',2,'left','left_low','no']]

df2 = pd.
 ↪DataFrame(newdata,columns=['Class','Age','MenoPause','Tumor-size','inv-nodes','node-caps','
                                    'breast','breast-quad','irradiate'])

df2
```

[11]:
```
                  Class    Age MenoPause Tumor-size inv-nodes node-caps  \
0     recurrence-events  50-59   premeno      25-29       0-2        no
1     recurrence-events  60-69      ge40      25-29       0-2        no
2     recurrence-events  50-59   premeno      15-19       0-2        no
3     recurrence-events  40-49   premeno      40-44       0-2        no
4  no-recurrence-events  50-59      ge40      35-39       0-2        no

   deg-malig breast breast-quad irradiate
0          1   left    left_low        no
1          3  right    left_low        no
2          2   left    left_low        no
3          1   left    left_low        no
4          2   left    left_low        no
```

[12]:
```
#Creating feature(x) and target(y) attributes form new testing dataset
X2 = df2.drop("irradiate", axis=1)
Y2 = df2["irradiate"]
```

[13]:
```
#Encoding X2 to numeric datatypes
le = LabelEncoder()

#Iterate over all the values of each column and extract their dtypes
for col in X2.columns:
    #Compare if the dtype is object
    if X2[col].dtype=='object':
        #Use LabelEncoder to do the numeric transformation
        X2[col]=le.fit_transform(X2[col])
```

[14]:
```
#Predict the response for test dataset
Y_pred2 = mnb.predict(X2)
print("\n------ Multinomial Naive Bayes------\n")

# Predicted values
print("Predicted values:",Y_pred2)

#Confusion Matrix
print("\nConfusion Matrix: \n\n",confusion_matrix(Y2, Y_pred2))

#Model Accuracy, how often is the classifier correct?
print("\nAccuracy:",metrics.accuracy_score(Y2, Y_pred2)*100)

# Report containing precision,recall,f1-score,support
print("\nReport--> \n",classification_report(Y2, Y_pred2))
```

```
------ Multinomial Naive Bayes------

Predicted values: ['no' 'no' 'no' 'no' 'no']
```

```
Confusion Matrix:

 [[5]]

Accuracy: 100.0

Report-->
              precision    recall  f1-score   support

          no       1.00      1.00      1.00         5

    accuracy                           1.00         5
   macro avg       1.00      1.00      1.00         5
weighted avg       1.00      1.00      1.00         5
```