CSL7670 : Fundamentals of Machine Learning

Lab Report

Name:	**SOUHITYA KUNDU**
Roll Number:	**M20PH209**
Program:	**MSc-MTech(Physics & Materials Sc.)**

Chapter 1

Lab-7

1.1 Objective

The objective of this whole assignment is to learn about Bayesian techniques, learn about the Confusion matrix and all the things which is related to performance metrics. In the homework assignment I have shown the definitions and also calculated the Precision,Recall and Accuracy value for given problem

1.2 Problem-1

The main objective is to understand Bayesian Learning approach by proper application in a dataset.

- First, I have divided the dataset in the form of 70:20:10 for training, validation, and testing
- Second I computed the mean and covariance matrices for both the given categories.

Solution 1(a):

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


# In[2]:


df = pd.read_csv("data␣-␣Sheet2.csv")


# In[3]:


df.head(5)


# In[4]:
```

```


X = df.drop(["Class␣Label"],axis = 1)
y = df["Class␣Label"]


# In[5]:


X,y
#1998 rows


# In[6]:


from sklearn.model_selection import train_test_split


# In[7]:


X_train,X_test1,y_train,y_test1 = train_test_split(X,y,test_size = 0.3, random_state = 42)
X_validation,X_test,y_validation,y_test = train_test_split(X_test1,y_test1 ,test_size = 0.33,random_state = 42)


# In[8]:


X_test.shape


# In[9]:


from sklearn.naive_bayes import GaussianNB

GNB = GaussianNB()

GNB.fit(X_train, y_train)

# Output Prediction
predicted = GNB.predict(X_test)


# In[10]:


print(f"Actual_value:{y_test}")
print(f"Predicted_value:{predicted}")


```

```
# In[11]:


from sklearn.metrics import accuracy_score,f1_score

accuracy = accuracy_score(predicted, y_test)
f1 = f1_score(predicted, y_test, average="weighted")

print(f"Accuracy:␣{accuracy}")
print(f"F1␣Score:␣{f1}")


# In[12]:


# For both the categories compute: (i) Mean (ii) Covariance
# Matrix.
# df[(df["Feature-1"]) & (df["Class Label"]==0)] This wont work because 1 st feature is non boolean type rather float
df1 = df[(~df["Feature-1"].isnull()) & (df["Class␣Label"] == 0)]
df1.mean()


# In[13]:


df2 = df[(~df["Feature-1"].isnull()) & (df["Class␣Label"] == 1)]
df2.mean()


# In[14]:


df3 = df[(~df["Feature-2"].isnull()) & (df["Class␣Label"] == 0)]
df3.mean()


# In[15]:


df4 = df[(~df["Feature-2"].isnull()) & (df["Class␣Label"] == 1)]
df4.mean()


# In[16]:


np.cov(df1["Feature-1"],df1["Class␣Label"])


# In[17]:


np.cov(df2["Feature-1"],df2["Class␣Label"])
```

```


# In[18]:


np.cov(df3["Feature-2"],df3["Class␣Label"])


# In[19]:


np.cov(df4["Feature-2"],df4["Class␣Label"])


# In[ ]:
```

Accuracy:	0.8939393939393939	
F1	Score:	0.8944517640169813

The Mean and Covariance Matrices for the following is given below:

- I have calculated the mean based on different features
- While calculation different classes have been considered separately

Feature-1(Class Label =0)	
Feature-1	150.179968
Feature-2	502.764772
Class Label	0.0
Feature-1(Class Label =1)	
Feature-1	45.173738
Feature-2	200.346970
Class Label	1.000000
Feature-2(Class Label =0)	
Feature-1	150.179968
Feature-2	502.764772
Class Label	0.000000
Feature-2(Class Label =1)	
Feature-1	45.173738
Feature-2	200.346970
Class Label	1.000000

Covariance Matrix 1(Feature -1 with Class 0)

```
[7274.99116657,                                  0.   ],
[                                                0.   ,    0.   ]
```

Covariance Matrix 2(Feature -1 with Class 1)

```
[675.40971162,                                   0.   ],
[                                                0.   ,    0.   ]
```

Covariance Matrix 3(Feature -2 with Class 0)

```
[82422.61505572,                                 0.   ],
[                                                0.   ,    0.   ]
```

Covariance Matrix 4(Feature -2 with Class 1)

```
[13504.08218254,                                 0.   ],
[                                                0.   ,    0.   ]
```