Breast__cancer__classificatin

June 4, 2022

1 Breast cancer classification using Logistic Regression algorithm

[101]:
```
!jupyter nbconvert --to pdf /content/Breast__cancer__classificatin.ipynb
```

```
[NbConvertApp] Converting notebook /content/Breast__cancer__classificatin.ipynb
to pdf
[NbConvertApp] Support files will be in Breast__cancer__classificatin_files/
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Making directory ./Breast__cancer__classificatin_files
[NbConvertApp] Writing 48633 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 236363 bytes to
/content/Breast__cancer__classificatin.pdf
```

[80]:
```
# importing libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

[81]:
```
# csv file -----> pandas dataframe

df = pd.read_csv('/content/Breast_Cancer_Database.csv')
df.tail()
```

[81]:
```
         id  Clump Thickness  Uniformity of Cell Size  \
678  776715                3                        1
679  841769                2                        1
680  888820                5                       10
681  897471                4                        8
682  897471                4                        8

     Uniformity of Cell Shape  Marginal Adhesion  Single Epithelial Cell Size  \
678                         1                  1                            3
679                         1                  1                            2
680                        10                  3                            7
681                         6                  4                            3
682                         8                  5                            4

     Bare Nuclei  Bland Chromatin  Normal Nucleoli  Mitoses  Class
678            2                1                1        1      2
679            1                1                1        1      2
680            3                8               10        2      4
681            4               10                6        1      4
682            5               10                4        1      4
```

[82]:
```
df.shape
```

[82]:
```
(683, 11)
```

[83]:
```
#re-assining dataframe column class values into 0 and 1

df['Class'] = df['Class'].apply(lambda x: 0 if x == 2 else 1)
df['Class'].value_counts()
```

[83]:
```
0    444
1    239
Name: Class, dtype: int64
```

[84]:
```
df.tail()
```

[84]:
```
         id  Clump Thickness  Uniformity of Cell Size  \
678  776715                3                        1
```

```
679  841769                2                      1
680  888820                5                     10
681  897471                4                      8
682  897471                4                      8

     Uniformity of Cell Shape  Marginal Adhesion  Single Epithelial Cell Size  \
678                         1                  1                            3
679                         1                  1                            2
680                        10                  3                            7
681                         6                  4                            3
682                         8                  5                            4

     Bare Nuclei  Bland Chromatin  Normal Nucleoli  Mitoses  Class
678            2                1                1        1      0
679            1                1                1        1      0
680            3                8               10        2      1
681            4               10                6        1      1
682            5               10                4        1      1
```

[85]:
```
# statistical discription of data

df.describe()
```

[85]:
```
                 id  Clump Thickness  Uniformity of Cell Size  \
count  6.830000e+02       683.000000               683.000000
mean   1.076720e+06         4.442167                 3.150805
std    6.206440e+05         2.820761                 3.065145
min    6.337500e+04         1.000000                 1.000000
25%    8.776170e+05         2.000000                 1.000000
50%    1.171795e+06         4.000000                 1.000000
75%    1.238705e+06         6.000000                 5.000000
max    1.345435e+07        10.000000                10.000000

       Uniformity of Cell Shape  Marginal Adhesion  \
count                683.000000         683.000000
mean                   3.215227           2.830161
std                    2.988581           2.864562
min                    1.000000           1.000000
25%                    1.000000           1.000000
50%                    1.000000           1.000000
75%                    5.000000           4.000000
max                   10.000000          10.000000

       Single Epithelial Cell Size  Bare Nuclei  Bland Chromatin  \
count                   683.000000   683.000000       683.000000
mean                      3.234261     3.544656         3.445095
std                       2.223085     3.643857         2.449697
```

```
min                       1.000000     1.000000          1.000000
25%                       2.000000     1.000000          2.000000
50%                       2.000000     1.000000          3.000000
75%                       4.000000     6.000000          5.000000
max                      10.000000    10.000000         10.000000

       Normal Nucleoli     Mitoses        Class
count       683.000000  683.000000   683.000000
mean          2.869693    1.603221     0.349927
std           3.052666    1.732674     0.477296
min           1.000000    1.000000     0.000000
25%           1.000000    1.000000     0.000000
50%           1.000000    1.000000     0.000000
75%           4.000000    1.000000     1.000000
max          10.000000   10.000000     1.000000
```

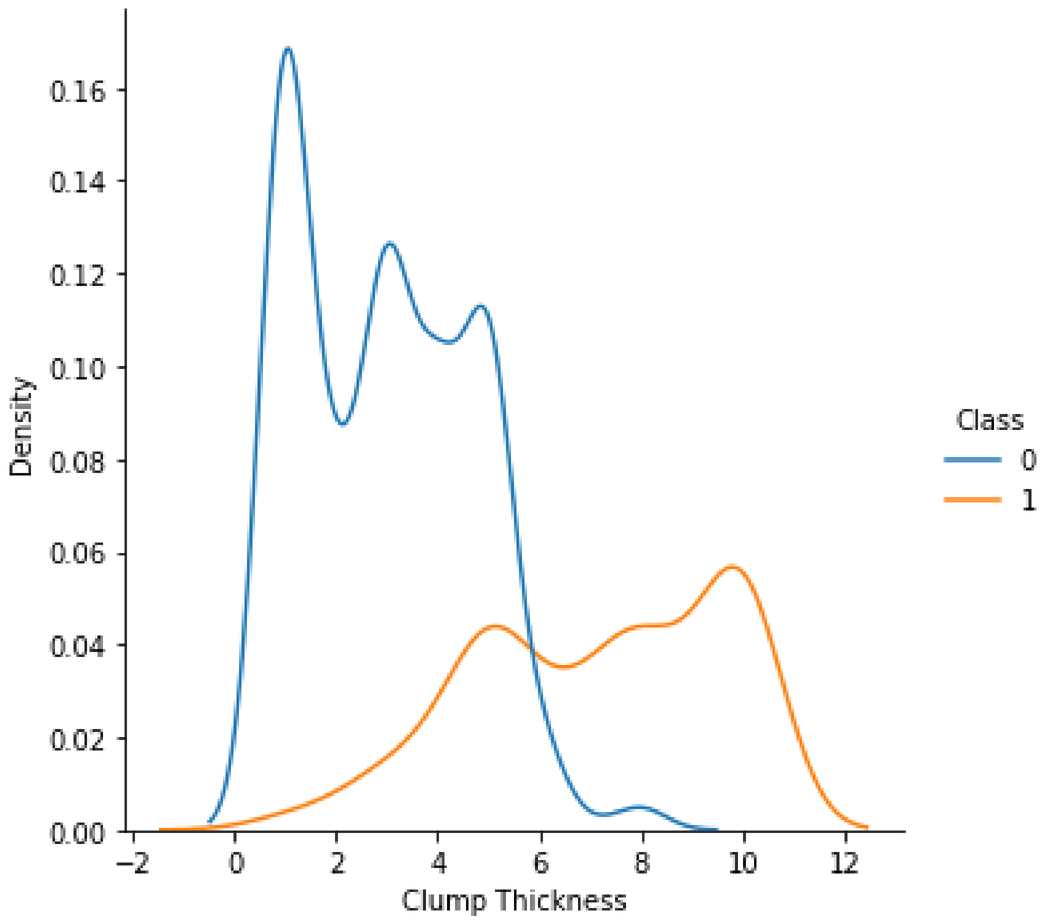
[86]:
```
# checking null values is present or not

df.isnull().sum()
```

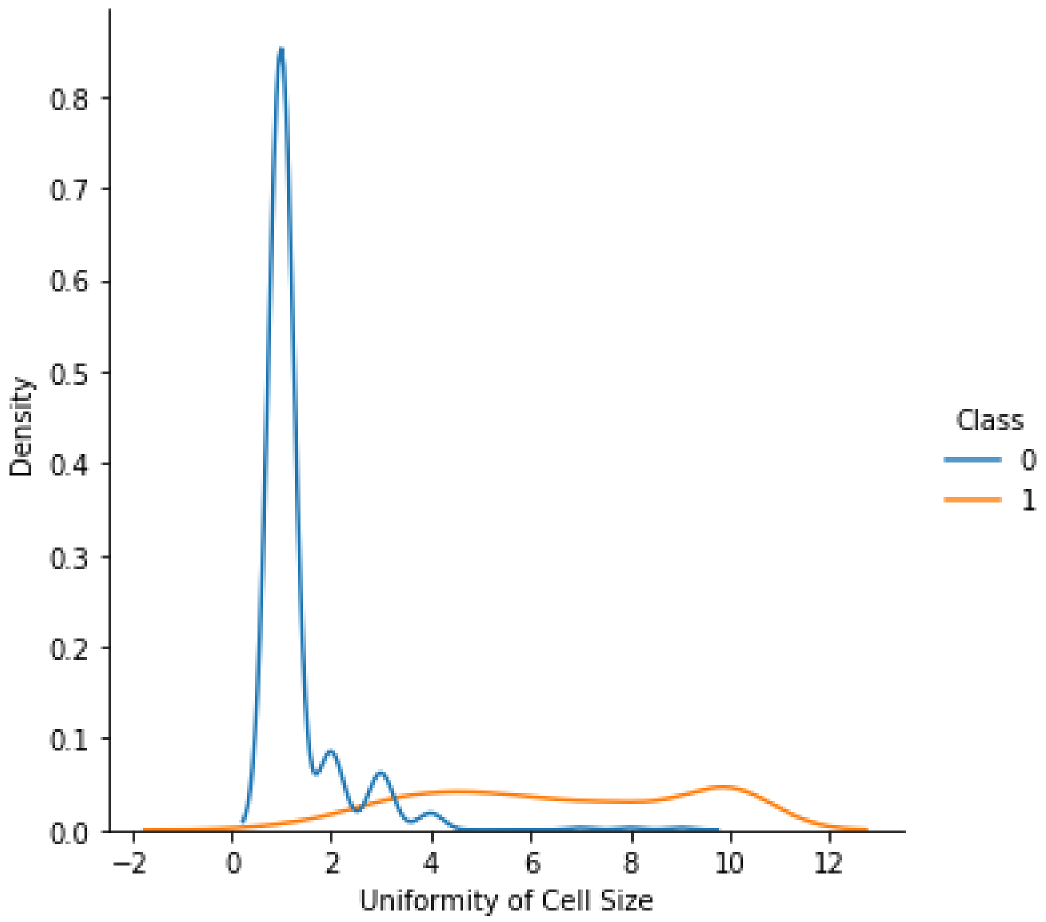
[86]:
```
id                             0
Clump Thickness                0
Uniformity of Cell Size        0
Uniformity of Cell Shape       0
Marginal Adhesion              0
Single Epithelial Cell Size    0
Bare Nuclei                    0
Bland Chromatin                0
Normal Nucleoli                0
Mitoses                        0
Class                          0
dtype: int64
```

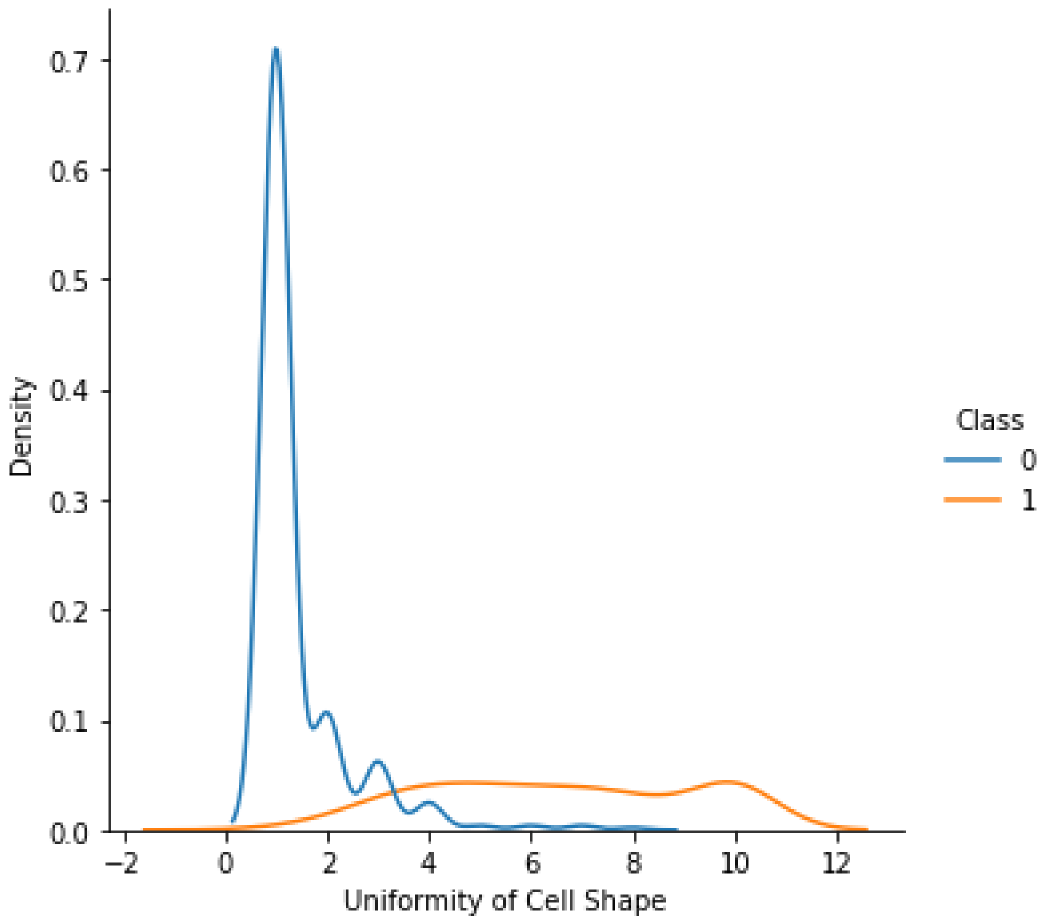
[107]:
```
# Data visualization using pairplot of seaborn
col = list(df.columns)
col = col[1:-1]
for c in col:
  sns.displot(df, x = c, hue = 'Class', kind='kde', color = 'darkblue', label =␣
 ↪c, legend = True,)
  plt.show()

#sns.pairplot(df, hue = 'Class', vars = list(df.columns))
```

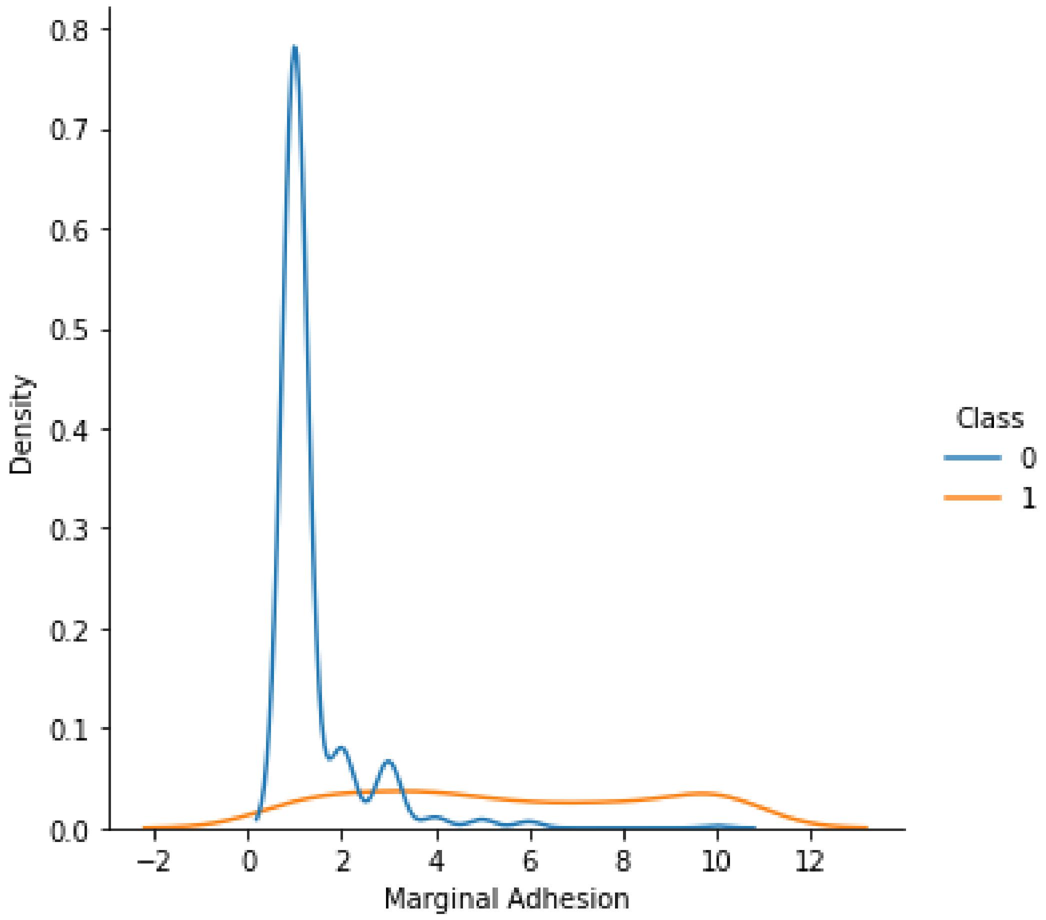


Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
−2
0
2
4
6
8
10
12
Clump Thickness
Class
0
1

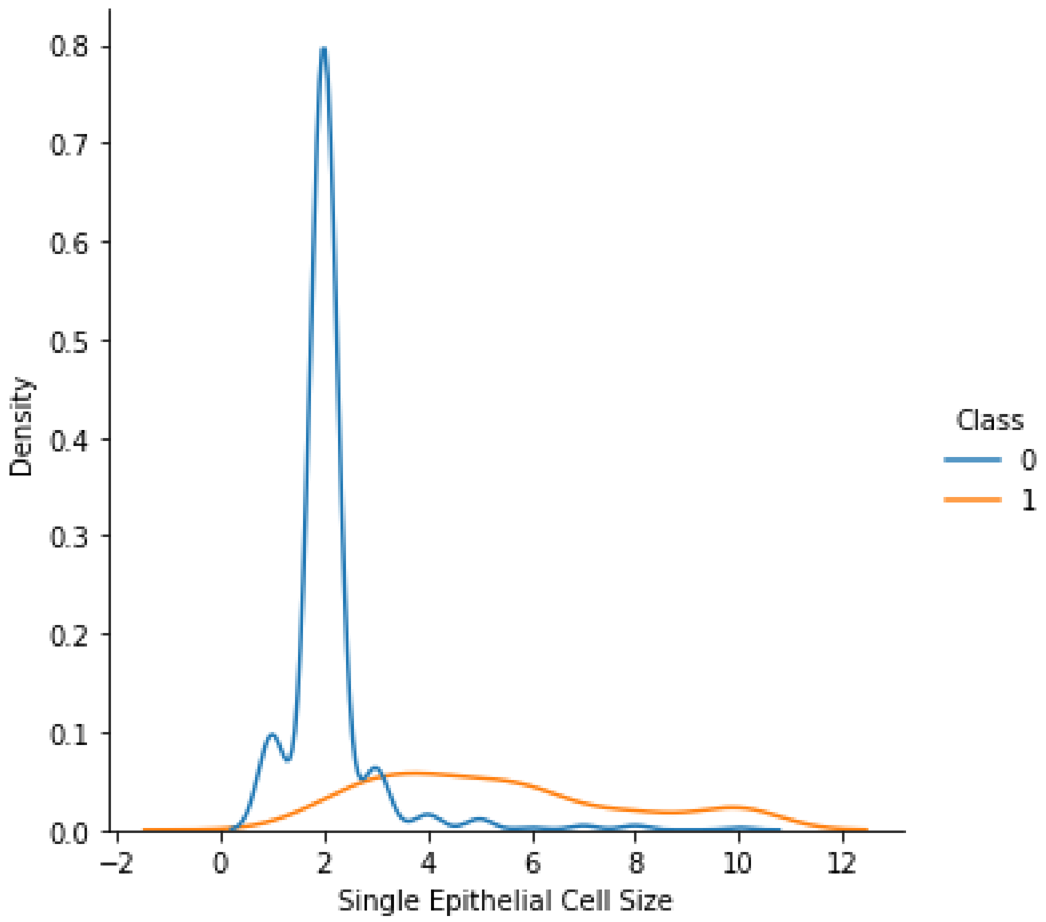

Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
−2
0
2
4
6
8
10
12
Uniformity of Cell Size
Class
0
1



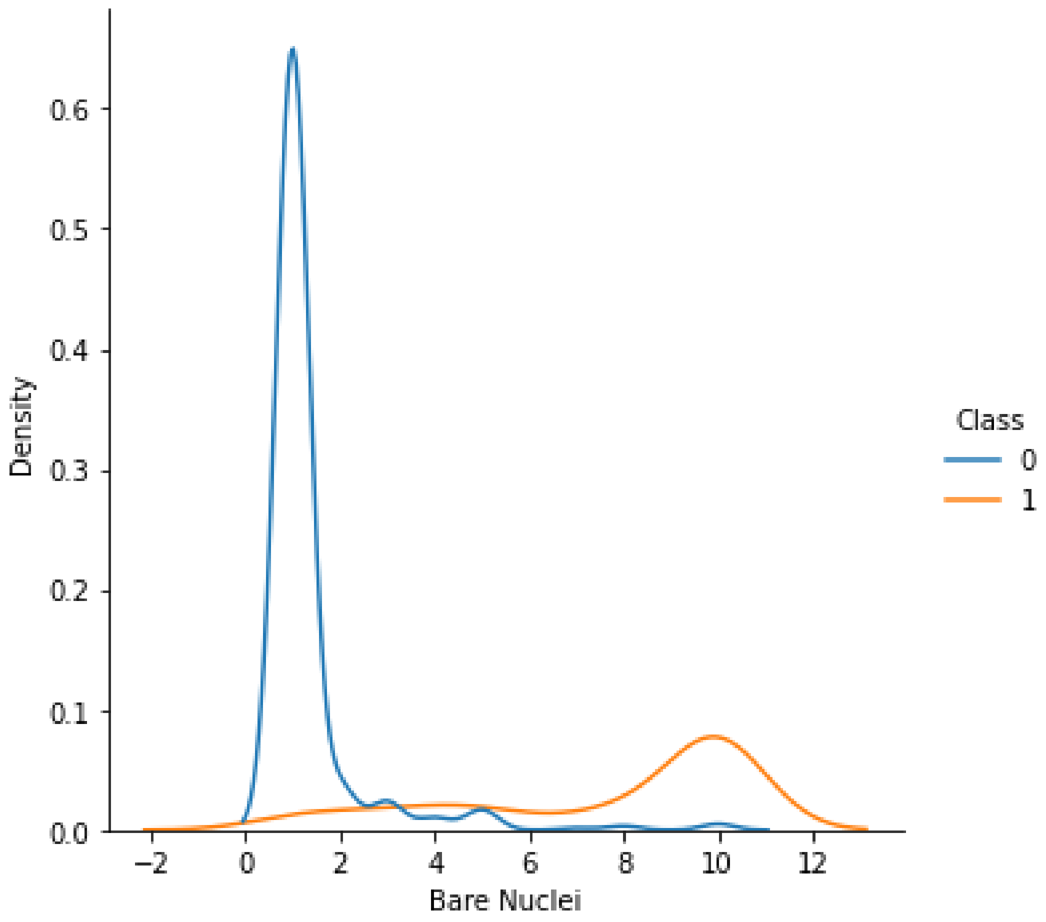
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−2
0
2
4
6
8
10
12
Uniformity of Cell Shape
Class
0
1

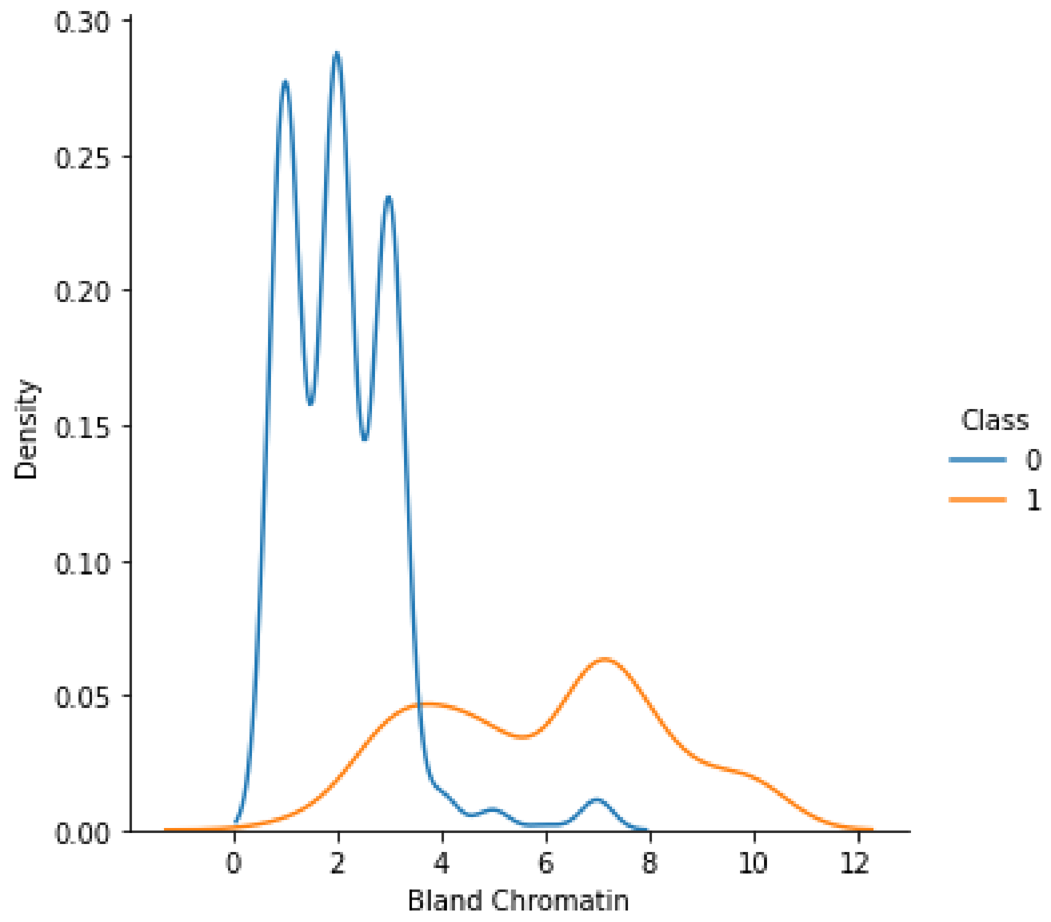


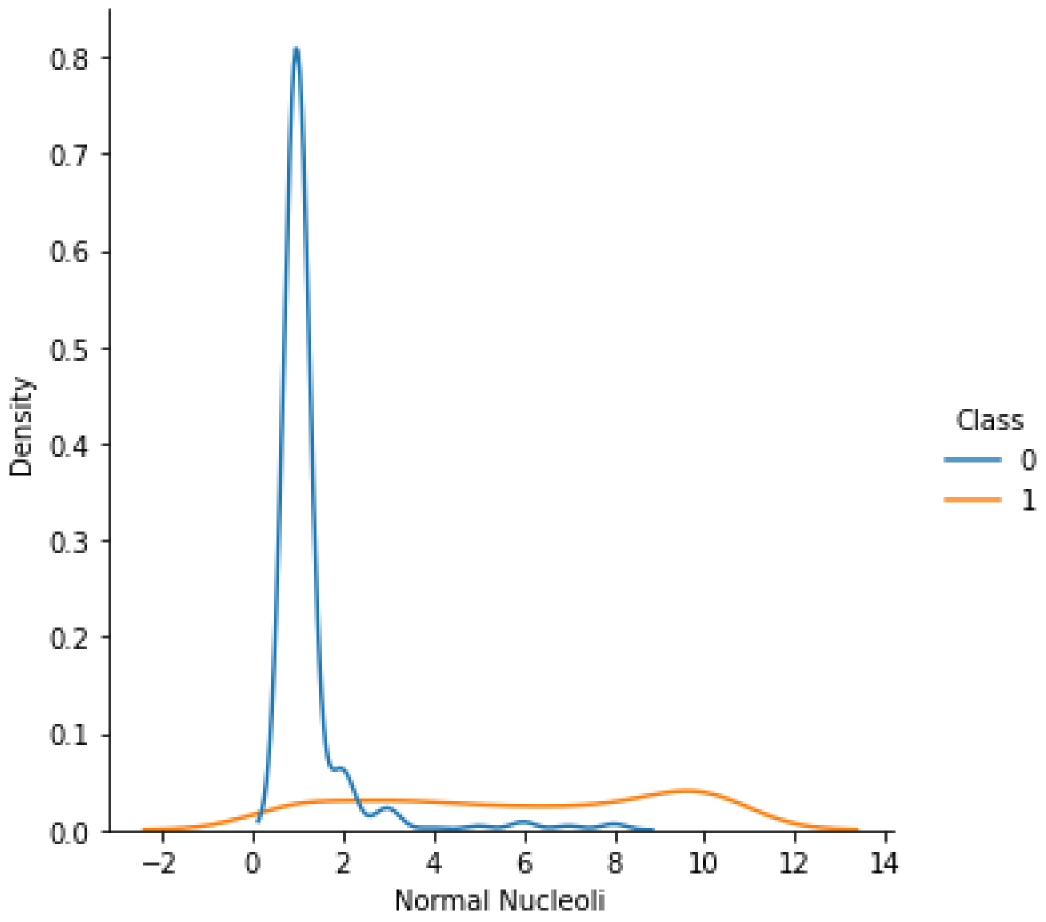
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
−2
0
2
4
6
8
10
12
Marginal Adhesion
Class
0
1



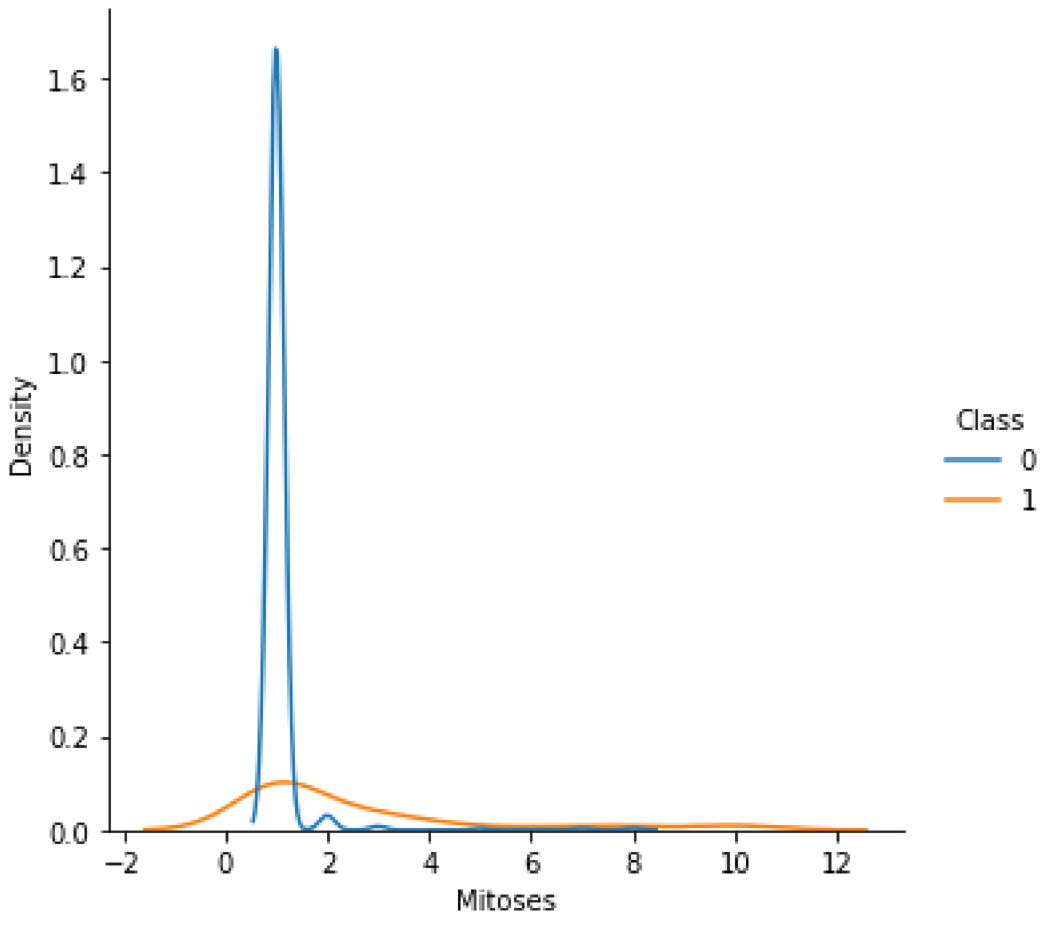
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Density
−2
0
2
4
6
8
10
12
Single Epithelial Cell Size
Class
0
1


Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−2
0
2
4
6
8
10
12
Bare Nuclei
Class
0
1


0.30
0.25
0.20
0.15
0.10
0.05
0.00
Density
0
2
4
6
8
10
12
Bland Chromatin
Class
0
1



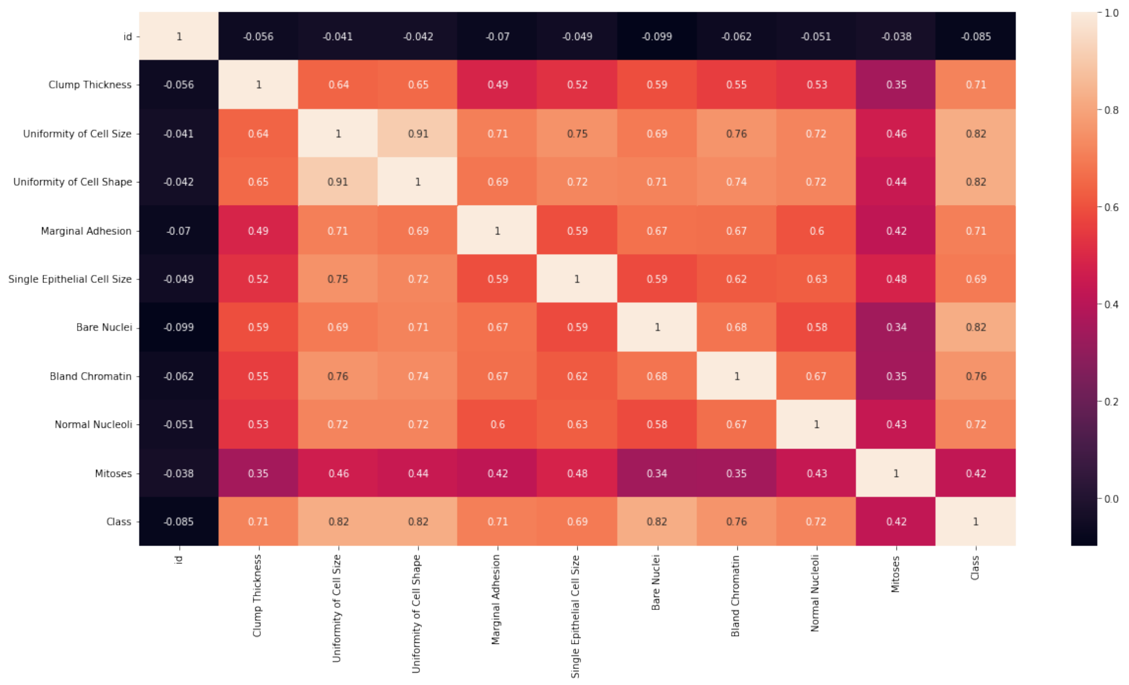
Density
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−2
0
2
4
6
8
10
12
14
Normal Nucleoli
Class
0
1



[88]:
```
# graphical depiction of data

plt.figure(figsize=(20,10))
sns.heatmap(df.corr(),annot = True)
```

[88]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2a9ade9c50>
```



[89]:
```
# dividing data into X and y

X = df.drop(['id','Class'], axis = 1).values
y = df['Class']
```

[90]:
```
X = pd.DataFrame(X)
```

[91]:
```
# splitting X ---> (X_train, X_test) and y ---> (y_train, y_test)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,␣
 ↪random_state=42)
```

2 *Logistic Regression Algorithm*

[92]:
```
model = LogisticRegression()
```

[93]:
```
# fitting data into model

model.fit(X_train,y_train)
```

[93]:
```
LogisticRegression()
```

[94]:
```
# prediction of values or feature elements

predict_value = model.predict(X_train)
```

[95]:
```
# training data accuracy

from sklearn.metrics import accuracy_score
training_score = accuracy_score(y_train, predict_value)
print(training_score)
```

```
0.9706959706959707
```

3 *Accuracy score*

[96]:
```
# testing data accuracy

predict_value = model.predict(X_test)
testing_score = accuracy_score(y_test, predict_value)
testing_score
```

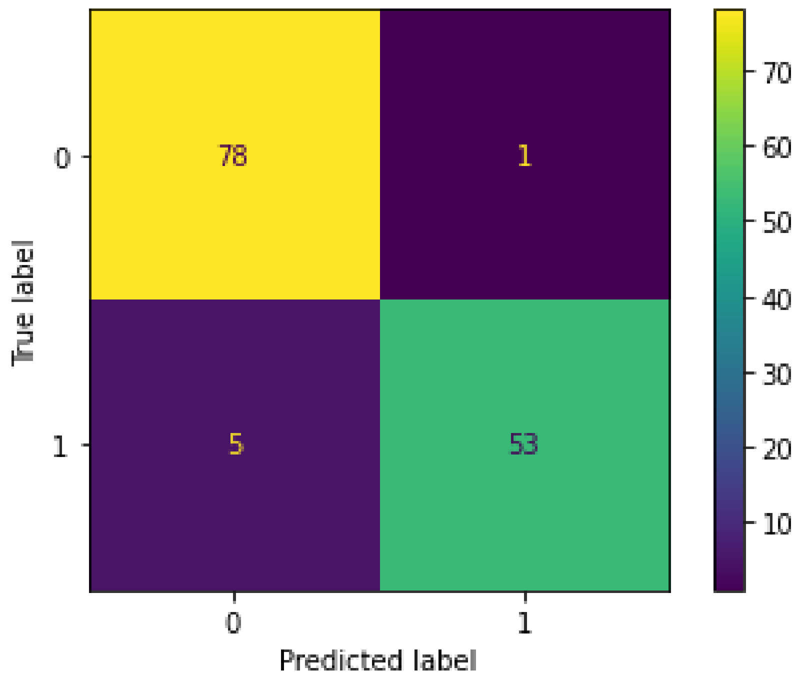
[96]:
```
0.9562043795620438
```

4 *Confusion matrix*

[97]:
```
# confusion matrix visualiztion

ConfusionMatrixDisplay.from_predictions(y_test, predict_value)
```

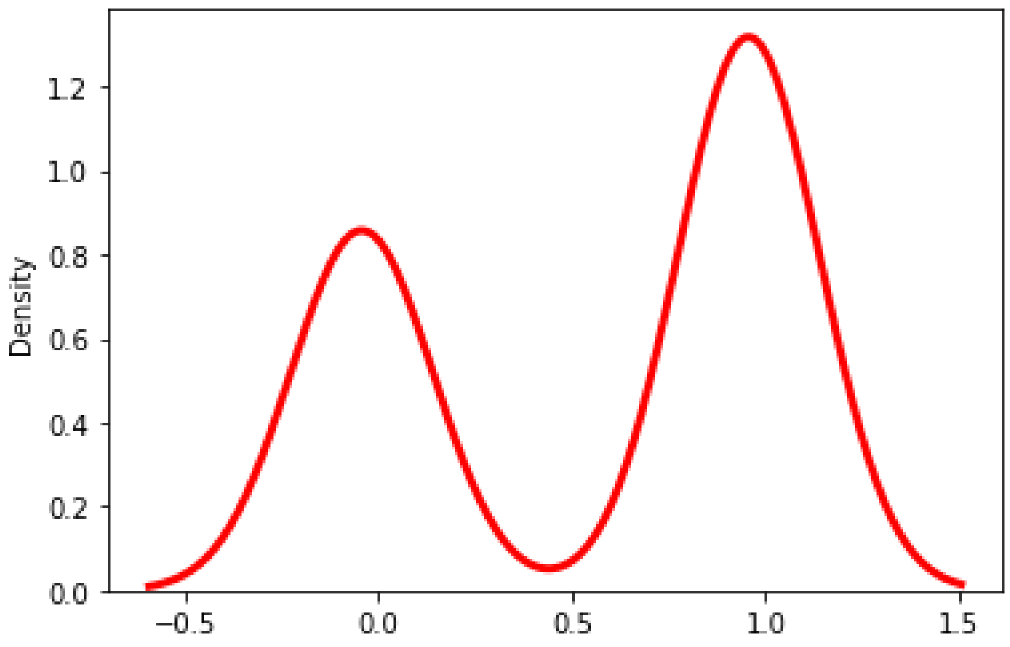
[97]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7f2a9aef4f10>
```



[98]:
```
# distplot or density plot

sns.distplot(testing_score-predict_value, kde = True, hist=False, color='red',␣
↪kde_kws = {'linewidth':3})
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `kdeplot` (an axes-level function for
kernel density plots).
  warnings.warn(msg, FutureWarning)
```



```
[99]: # giving new data into model and make prediction

      data = [float(i) for i in input().strip().split()]
      data = np.array(data)
      data = data.reshape(1,-1)
      prediction = model.predict(data)
```

```
4 8 8 5 4 5 10 4 1
```

```
[100]: # model prediction

       if prediction == 1:
         print("malignant")
       elif prediction == 0:
         print("benign")
```

```
malignant
```

```
[100]:
```