Third Year B. Tech (EL & CE)

Semester: VI **Subject:** Data Science for Engineering

Name: Shreerang Mhatre **Class: TY**

Roll No: 52 **Batch: A2**

Experiment No: 06

Name of the Experiment: **Classification using naive bayes model**

Performed on: 25/04/2024

Submitted on: 25/04/2024

Problem Statement:

Aim: Write a python program to build a model to classify the type of cancer. The data has two types of cancer classes: malignant (harmful) and benign (not harmful).
Perform following steps:

- Load the Data (The dataset is available in the scikit-learn library).
- Exploring Data: Prints features, Shape, Size, labels, head records, data types, outliers etc.
- Split the data into train and test set.
- Select the classification model.
- Fit the model on train data.
- Predict the outcome on test data.
- Evaluate the performance of model: Confusion Matrix, accuracy, F1, precision,
- Check of Tuning Hyperparameters of the model to improve performance.

localhost:8888/notebooks/DSE%20Practicals/EXP%20-%206/Shreerang%20Mhatre%20Exp6.ipynb

jupyter Shreerang Mhatre Exp6 Last Checkpoint: Last Tuesday at 10:18 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = pd.read_csv('Breast_Cancer_data.csv')
```

In [3]:
```
data.describe()
```

Out[3]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.627417
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.483918
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.000000
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.000000
50%	13.370000	18.840000	86.240000	551.100000	0.095870	1.000000
75%	15.780000	21.800000	104.100000	782.700000	0.105300	1.000000
max	28.110000	39.280000	188.500000	2501.000000	0.163400	1.000000

In [4]:
```
data.head()
```

Out[4]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
0	17.99	10.38	122.80	1001.0	0.11840	0
1	20.57	17.77	132.90	1326.0	0.08474	0
2	19.69	21.25	130.00	1203.0	0.10960	0

Type here to search 37°C ENG 05:27 PM 25-04-2024

localhost:8888/notebooks/DSE%20Practicals/EXP%20-%206/Shreerang%20Mhatre%20Exp6.ipynb

jupyter Shreerang Mhatre Exp6 Last Checkpoint: Last Tuesday at 10:18 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

75%	15.780000	21.800000	104.100000	782.700000	0.105300	1.000000
max	28.110000	39.280000	188.500000	2501.000000	0.163400	1.000000

In [4]:
```
data.head()
```

Out[4]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
0	17.99	10.38	122.80	1001.0	0.11840	0
1	20.57	17.77	132.90	1326.0	0.08474	0
2	19.69	21.25	130.00	1203.0	0.10960	0
3	11.42	20.38	77.58	386.1	0.14250	0
4	20.29	14.34	135.10	1297.0	0.10030	0

In [5]:
```
data.shape
```

Out[5]: (569, 6)

In [6]:
```
data["diagnosis"].hist()
```

Out[6]: <AxesSubplot:>

Type here to search 37°C ENG 05:27 PM 25-04-2024

localhost:8888/notebooks/DSE%20Practicals/EXP%20-%206/Shreerang%20Mhatre%20Exp6.ipynb

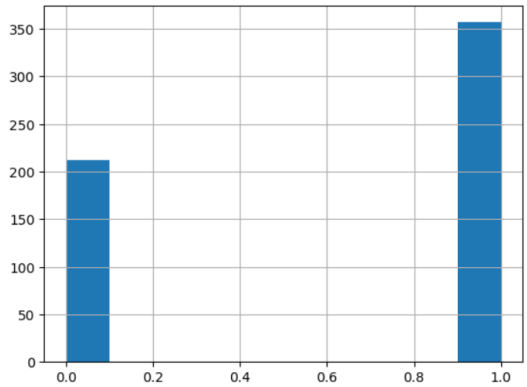
jupyter Shreerang Mhatre Exp6 Last Checkpoint: Last Tuesday at 10:18 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

Out[5]: (569, 6)

In [6]:
```
data["diagnosis"].hist()
```

Out[6]: <AxesSubplot:>



In [7]:
```
corr = data.iloc[:,:-1].corr(method="pearson")
```

In [8]:
```
corr
```

Type here to search 37°C ENG 05:27 PM 25-04-2024

localhost:8888/notebooks/DSE%20Practicals/EXP%20-%206/Shreerang%20Mhatre%20Exp6.ipynb

jupyter Shreerang Mhatre Exp6 Last Checkpoint: Last Tuesday at 10:18 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

In [7]:
```
corr = data.iloc[:,:-1].corr(method="pearson")
```

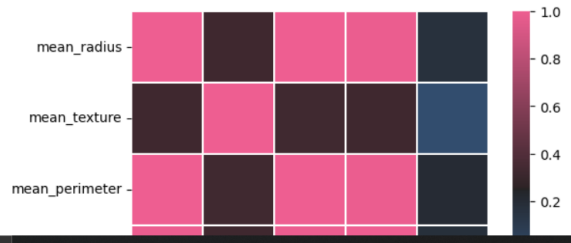
In [8]:
```
corr
```

Out[8]:

	mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness
mean_radius	1.000000	0.323782	0.997855	0.987357	0.170581
mean_texture	0.323782	1.000000	0.329533	0.321086	-0.023389
mean_perimeter	0.997855	0.329533	1.000000	0.986507	0.207278
mean_area	0.987357	0.321086	0.986507	1.000000	0.177028
mean_smoothness	0.170581	-0.023389	0.207278	0.177028	1.000000

In [9]:
```
cmap = sns.diverging_palette(250,354,80,60,center='dark',as_cmap=True)
sns.heatmap(corr, vmax=1, vmin=-.5, cmap=cmap, square=True, linewidth=.2)
```

Out[9]: <AxesSubplot:>



Type here to search 37°C ENG 05:28 PM 25-04-2024

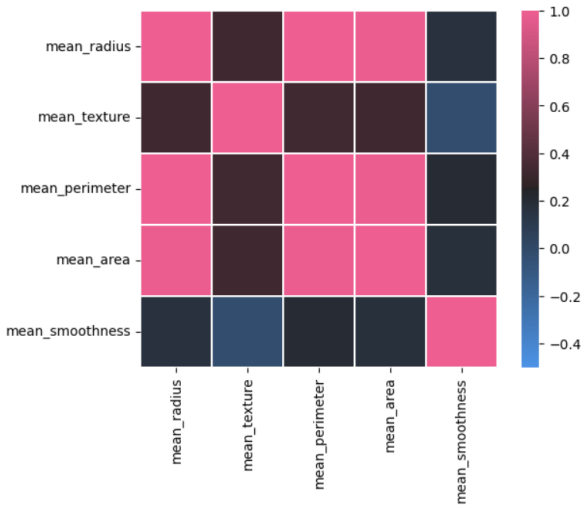
localhost:8888/notebooks/DSE%20Practicals/EXP%20-%206/Shreerang%20Mhatre%20Exp6.ipynb

jupyter Shreerang Mhatre Exp6 Last Checkpoint: Last Tuesday at 10:18 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

In [9]:
```
cmap = sns.diverging_palette(250,354,80,60,center='dark',as_cmap=True)
sns.heatmap(corr, vmax=1, vmin=-.5, cmap=cmap, square=True, linewidth=.2)
```

Out[9]: <AxesSubplot:>



localhost:8888/notebooks/DSE%20Practicals/EXP%20-%206/Shreerang%20Mhatre%20Exp6.ipynb

jupyter Shreerang Mhatre Exp6 Last Checkpoint: Last Tuesday at 10:18 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

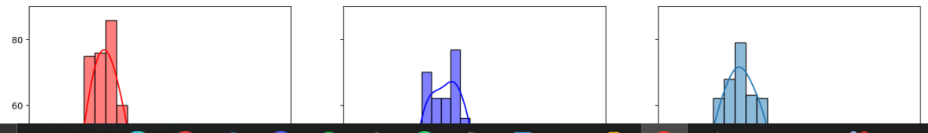
In [10]:
```
data = data[["mean_radius", "mean_texture", "mean_smoothness", "diagnosis"]]
data.head(10)
```

Out[10]:

	mean_radius	mean_texture	mean_smoothness	diagnosis
0	17.99	10.38	0.11840	0
1	20.57	17.77	0.08474	0
2	19.69	21.25	0.10960	0
3	11.42	20.38	0.14250	0
4	20.29	14.34	0.10030	0
5	12.45	15.70	0.12780	0
6	18.25	19.98	0.09463	0
7	13.71	20.83	0.11890	0
8	13.00	21.82	0.12730	0
9	12.46	24.04	0.11860	0

In [11]:
```
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)
sns.histplot(data, ax=axes[0], x="mean_radius", kde=True, color='r')
sns.histplot(data, ax=axes[1], x="mean_smoothness", kde=True, color='b')
sns.histplot(data, ax=axes[2], x="mean_texture", kde=True)
```

Out[11]: <AxesSubplot:xlabel='mean_texture', ylabel='Count'>



localhost:8888/notebooks/DSE%20Practicals/EXP%20-%206/Shreerang%20Mhatre%20Exp6.ipynb

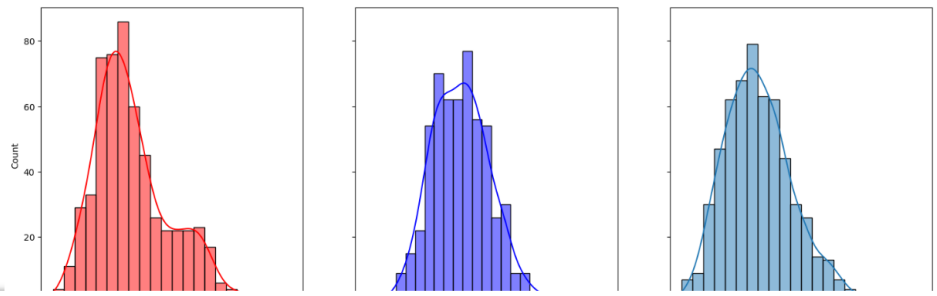
jupyter Shreerang Mhatre Exp6 Last Checkpoint: Last Tuesday at 10:18 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

7	13.71	20.83	0.11890	0
8	13.00	21.82	0.12730	0
9	12.46	24.04	0.11860	0

In [11]:
```
fig, axes = plt.subplots(1, 3, figsize=(18, 6), sharey=True)
sns.histplot(data, ax=axes[0], x="mean_radius", kde=True, color='r')
sns.histplot(data, ax=axes[1], x="mean_smoothness", kde=True, color='b')
sns.histplot(data, ax=axes[2], x="mean_texture", kde=True)
```

Out[11]:
```
<AxesSubplot:xlabel='mean_texture', ylabel='Count'>
```



In [12]:
```
X = data[["mean_radius", "mean_texture", "mean_smoothness"]]
y = data["diagnosis"]
```

In [13]:
```
from sklearn.model_selection import train_test_split
#X_train, y_train, y_test = train_test_split(X, y, text_size=0.2, random_state=42)
```

Type here to search | 37°C | ENG | 05:28 PM 25-04-2024

localhost:8888/notebooks/DSE%20Practicals/EXP%20-%206/Shreerang%20Mhatre%20Exp6.ipynb

jupyter Shreerang Mhatre Exp6 Last Checkpoint: Last Tuesday at 10:18 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

In [13]:
```
from sklearn.model_selection import train_test_split
#X_train, y_train, y_test = train_test_split(X, y, text_size=0.2, random_state=42)

train, test = train_test_split(data, test_size=0.1, random_state=42)
X_test= test.iloc[:,:-1].values
y_test= test.iloc [:,-1].values


X_train= train.iloc[:,:-1].values
y_train= train.iloc [:,-1].values
```

In [14]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)
```

Out[14]:
```
GaussianNB()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [15]:
```
y_pred = model.predict(X_test)
```

In [16]:
```
y_pred
```

Out[16]:
```
array([1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1], dtype=int64)
```

In [17]:
```
from sklearn.metrics import confusion_matrix, f1_score
CM = confusion_matrix(y_test,y_pred)
f1 = f1_score(y_test,y_pred)
```

In [18]:
```
f1
```

Out[18]:
```
0.975
```

Type here to search | Result | ENG | 05:28 PM 25-04-2024

Exp 6 - DSE

Name - Shreerang Mhatre
Rollno - 52

* Post Lab Questions -

Q1) How does the Naive Bayes classifier handle the assumption of feature independence?

→ The Naive Bayes classifier handles the assumption of feature independence by assuming that the features used to make predictions are conditionally independent given the class label. This means that the presence or absence of a particular feature is unrelated to the class label. Despite this simplifying assumption, Naive Bayes can perform well in practice, especially with large datasets and when the independence assumption is approximately met.

Q2) Explain the significance of precision, recall & the F1 score in evaluating the performance of a classification model.

→ ① Precision -
- Precision is the ratio of true positive

predictions to the total number of positive predictions made by the model. It measures the accuracy of the positive predictions made by the model.

② Recall (sensitivity or True Positive Rate):

- Recall is the ratio of true positive predictions to the total number of actual positive instances in the dataset. It measures the model's ability to correctly identify all positive instances in the dataset, regardless of how many false negatives it generates.

③ F1 Score:

- The F1 Score is the harmonic mean of precision & recall. It provides a single metric that balances both precision & recall. The F1 score is useful when you want to consider both false positives & false negatives equally important

→

www.mitwpu.edu.in

Q3) How does the interpretation of a confusion matrix help in understanding the strengths & weakness of a classification model?

→ A confusion matrix is a powerful tool for evaluating the performance of a classification model by summarizing the counts of true positive (TP), false positive (FP), true negative (TN), & false negative (FN) predictions. Interpreting a confusion matrix helps in understanding the strengths & weaknesses of a classification model.

1. Accuracy Assesment
2. Precision & Roll call
3. Specificity & sensitivity
4. F1 Score
5. Identifying Model weaknesses
6. Adjusting Model Parameters