```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```python
data=pd.read_csv("/content/cancer_data.csv")
print(data)
```

```
           id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0      842302         M        17.99         10.38          122.80     1001.0
1      842517         M        20.57         17.77          132.90     1326.0
2    84300903         M        19.69         21.25          130.00     1203.0
3    84348301         M        11.42         20.38           77.58      386.1
4    84358402         M        20.29         14.34          135.10     1297.0
..        ...       ...          ...           ...             ...        ...
564    926424         M        21.56         22.39          142.00     1479.0
565    926682         M        20.13         28.25          131.20     1261.0
566    926954         M        16.60         28.08          108.30      858.1
567    927241         M        20.60         29.33          140.10     1265.0
568     92751         B         7.76         24.54           47.92      181.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0            0.11840           0.27760         0.30010              0.14710
1            0.08474           0.07864         0.08690              0.07017
2            0.10960           0.15990         0.19740              0.12790
3            0.14250           0.28390         0.24140              0.10520
4            0.10030           0.13280         0.19800              0.10430
..               ...               ...             ...                  ...
564          0.11100           0.11590         0.24390              0.13890
565          0.09780           0.10340         0.14400              0.09791
566          0.08455           0.10230         0.09251              0.05302
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     ...  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0    ...          17.33           184.60      2019.0           0.16220
1    ...          23.41           158.80      1956.0           0.12380
2    ...          25.53           152.50      1709.0           0.14440
3    ...          26.50            98.87       567.7           0.20980
4    ...          16.67           152.20      1575.0           0.13740
..   ...            ...              ...         ...               ...
564  ...          26.40           166.10      2027.0           0.14100
565  ...          38.25           155.00      1731.0           0.11660
566  ...          34.12           126.70      1124.0           0.11390
567  ...          39.42           184.60      1821.0           0.16500
568  ...          30.37            59.16       268.6           0.08996

     compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0              0.66560           0.7119                0.2654          0.4601
1              0.18660           0.2416                0.1860          0.2750
2              0.42450           0.4504                0.2430          0.3613
3              0.86630           0.6869                0.2575          0.6638
4              0.20500           0.4000                0.1625          0.2364
..                 ...              ...                   ...             ...
```

```
564          0.21130        0.4107               0.2216          0.2060
565          0.19220        0.3215               0.1628          0.2572
566          0.30940        0.3403               0.1418          0.2218
567          0.86810        0.9387               0.2650          0.4087
568          0.06444        0.0000               0.0000          0.2871

     fractal_dimension_worst  Unnamed: 32
0                    0.11890          NaN
1                    0.08902          NaN
2                    0.08758          NaN
3                    0.17300          NaN
4                    0.07678          NaN
```

```
data_final = data.drop(["id", "Unnamed: 32","texture_se","area_se","compactness_se","concav
print(data_final)
```

```
    diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0           M        17.99         10.38          122.80     1001.0
1           M        20.57         17.77          132.90     1326.0
2           M        19.69         21.25          130.00     1203.0
3           M        11.42         20.38           77.58      386.1
4           M        20.29         14.34          135.10     1297.0
..        ...          ...           ...             ...        ...
564         M        21.56         22.39          142.00     1479.0
565         M        20.13         28.25          131.20     1261.0
566         M        16.60         28.08          108.30      858.1
567         M        20.60         29.33          140.10     1265.0
568         B         7.76         24.54           47.92      181.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0            0.11840           0.27760         0.30010              0.14710
1            0.08474           0.07864         0.08690              0.07017
2            0.10960           0.15990         0.19740              0.12790
3            0.14250           0.28390         0.24140              0.10520
4            0.10030           0.13280         0.19800              0.10430
..               ...               ...             ...                  ...
564          0.11100           0.11590         0.24390              0.13890
565          0.09780           0.10340         0.14400              0.09791
566          0.08455           0.10230         0.09251              0.05302
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     symmetry_mean  ...  radius_se  perimeter_se  smoothness_se  concavity_se  \
0           0.2419  ...     1.0950         8.589       0.006399       0.05373
1           0.1812  ...     0.5435         3.398       0.005225       0.01860
2           0.2069  ...     0.7456         4.585       0.006150       0.03832
3           0.2597  ...     0.4956         3.445       0.009110       0.05661
4           0.1809  ...     0.7572         5.438       0.011490       0.05688
..             ...  ...        ...           ...            ...           ...
564         0.1726  ...     1.1760         7.673       0.010300       0.05198
565         0.1752  ...     0.7655         5.203       0.005769       0.03950
566         0.1590  ...     0.4564         3.425       0.005903       0.04730
567         0.2397  ...     0.7260         5.772       0.006522       0.07117
568         0.1587  ...     0.3857         2.548       0.007189       0.00000

     symmetry_se  radius_worst  perimeter_worst  smoothness_worst  \
```

0	0.03003	25.380	184.60	0.16220
1	0.01389	24.990	158.80	0.12380
2	0.02250	23.570	152.50	0.14440
3	0.05963	14.910	98.87	0.20980
4	0.01756	22.540	152.20	0.13740
..	...	...	...	...
564	0.01114	25.450	166.10	0.14100
565	0.01898	23.690	155.00	0.11660
566	0.01318	18.980	126.70	0.11390
567	0.02324	25.740	184.60	0.16500
568	0.02676	9.456	59.16	0.08996

	concavity_worst	symmetry_worst
0	0.7119	0.4601
1	0.2416	0.2750
2	0.4504	0.3613
3	0.6869	0.6638
4	0.4000	0.2364

```
sns.heatmap(data_final.isnull())
```

<Axes: >

```
data_final[["diagnosis","texture_mean","area_mean","compactness_mean","concave points_mean"
print(data_final)
```

```
    diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0           M        17.99         10.38          122.80     1001.0   
1           M        20.57         17.77          132.90     1326.0   
2           M        19.69         21.25          130.00     1203.0   
3           M        11.42         20.38           77.58      386.1   
4           M        20.29         14.34          135.10     1297.0   
..        ...          ...           ...             ...        ...   
564         M        21.56         22.39          142.00     1479.0   
565         M        20.13         28.25          131.20     1261.0   
566         M        16.60         28.08          108.30      858.1   
567         M        20.60         29.33          140.10     1265.0   
568         B         7.76         24.54           47.92      181.0   

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0            0.11840           0.27760         0.30010              0.14710   
1            0.08474           0.07864         0.08690              0.07017   
2            0.10960           0.15990         0.19740              0.12790   
3            0.14250           0.28390         0.24140              0.10520   
4            0.10030           0.13280         0.19800              0.10430   
..               ...               ...             ...                  ...   
564          0.11100           0.11590         0.24390              0.13890   
565          0.09780           0.10340         0.14400              0.09791   
566          0.08455           0.10230         0.09251              0.05302   
567          0.11780           0.27700         0.35140              0.15200   
568          0.05263           0.04362         0.00000                  NaN   

     symmetry_mean  ...  radius_se  perimeter_se  smoothness_se  concavity_se  \
0           0.2419  ...     1.0950         8.589       0.006399       0.05373   
1           0.1812  ...     0.5435         3.398       0.005225       0.01860   
2           0.2069  ...     0.7456         4.585       0.006150       0.03832   
3           0.2597  ...     0.4956         3.445       0.009110       0.05661   
4           0.1809  ...     0.7572         5.438       0.011490       0.05688   
..             ...  ...        ...           ...            ...           ...   
564         0.1726  ...     1.1760         7.673       0.010300       0.05198   
565         0.1752  ...     0.7655         5.203       0.005769       0.03950   
566         0.1590  ...     0.4564         3.425       0.005903       0.04730   
567         0.2397  ...     0.7260         5.772       0.006522       0.07117   
568         0.1587  ...     0.3857         2.548       0.007189       0.00000   

     symmetry_se  radius_worst  perimeter_worst  smoothness_worst  \
0        0.03003        25.380           184.60           0.16220   
1        0.01389        24.990           158.80           0.12380   
2        0.02250        23.570           152.50           0.14440   
3        0.05963        14.910            98.87           0.20980   
4        0.01756        22.540           152.20           0.13740   
..           ...           ...              ...               ...   
564      0.01114        25.450           166.10           0.14100   
565      0.01898        23.690           155.00           0.11660   
566      0.01318        18.980           126.70           0.11390   
567      0.02324        25.740           184.60           0.16500   
568      0.02676         9.456            59.16           0.08996   

     concavity_worst  symmetry_worst  
```

0	0.7119	0.4601
1	0.2416	0.2750
2	0.4504	0.3613
3	0.6869	0.6638
4	0.4000	0.2364

```
sns.heatmap(data_final.isnull())
```

```
<Axes: >
```

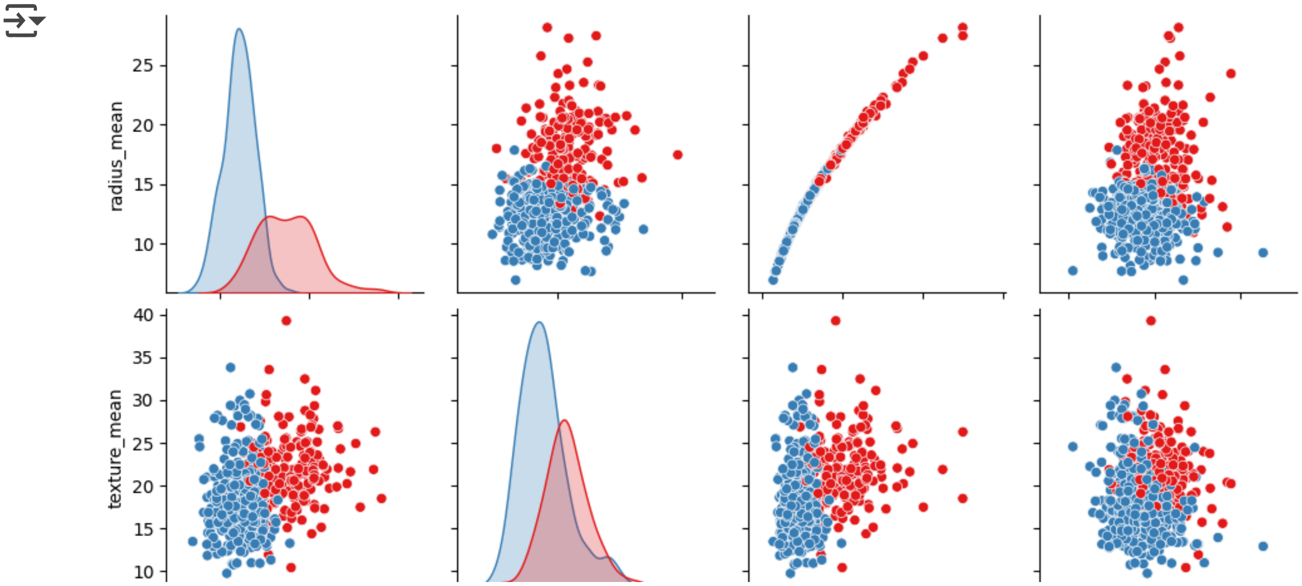

```
data_final.fillna({'texture_mean':data_final['texture_mean'].mean()},inplace=True)
data_final.fillna({'area_mean':data_final['area_mean'].mean()},inplace=True)
data_final.fillna({'compactness_mean':data_final['compactness_mean'].mean()},inplace=True)
data_final.fillna({'concave points_mean':data_final['concave points_mean'].mean()},inplace=
data_final.fillna({'fractal_dimension_mean':data_final['fractal_dimension_mean'].mean()},in
```

```
sns.heatmap(data_final.isnull())
```

<Axes: >


```
selected_features = ['radius_mean', 'texture_mean', 'area_mean', 'smoothness_mean', 'diagno

sns.pairplot(data[selected_features], hue='diagnosis', palette='Set1')
plt.show()
```


radius_mean
texture_mean
25
20
15
10
40
35
30
25
20
15
10

```
x=data_final.drop('diagnosis', axis=1).values
y=data_final['diagnosis'].values
```

```
print(data_final.shape)
```

```
(569, 31)
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=.2,random_state=0)
```

```
from sklearn.linear_model import LogisticRegression
logm=LogisticRegression()
logm.fit(x_train,y_train)
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: C
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressio
  n_iter_i = _check_optimize_result(
```

▾ LogisticRegression i ?

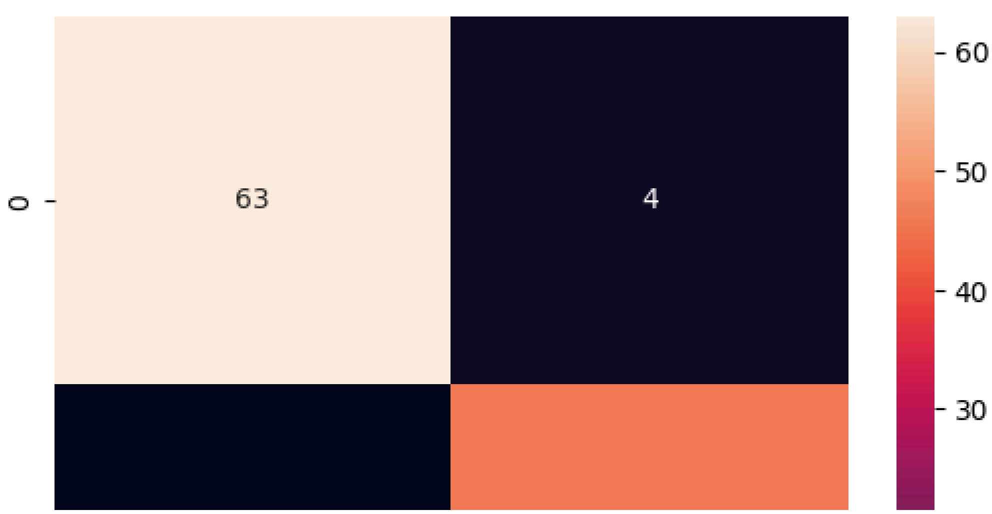
LogisticRegression()

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
ac=accuracy_score(y_test,logm.predict(x_test))
print(ac)
```

```
0.956140350877193
```

```
sns.heatmap(confusion_matrix(y_test,logm.predict(x_test)),annot=True)
```

```
<Axes: >
```



```
logm_cr=classification_report(y_test,logm.predict(x_test))
print(logm_cr)
```

```
              precision    recall  f1-score   support

           B       0.98      0.94      0.96        67
           M       0.92      0.98      0.95        47

    accuracy                           0.96       114
   macro avg       0.95      0.96      0.96       114
weighted avg       0.96      0.96      0.96       114
```


```
logm_cr=classification_report(y_test,logm.predict(x_test))
print(logm_cr)
```

```
              precision    recall  f1-score   support

           B       0.98      0.94      0.96        67
           M       0.92      0.98      0.95        47

    accuracy                           0.96       114
   macro avg       0.95      0.96      0.96       114
weighted avg       0.96      0.96      0.96       114
```


```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=7)
knn.fit(x_train,y_train)
```

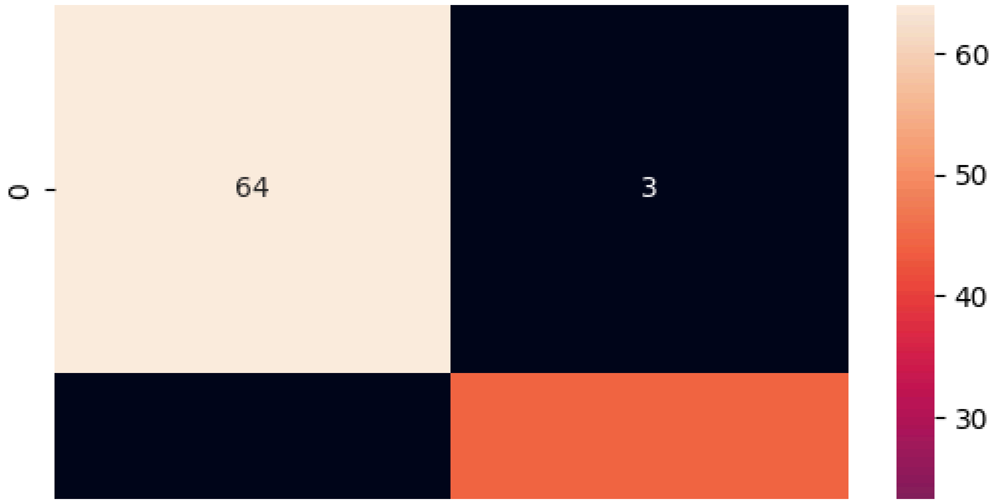
```
▾ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=7)
```

```
acknn=accuracy_score(y_test,knn.predict(x_test))
print(acknn)
```

```
0.9473684210526315
```

```
sns.heatmap(confusion_matrix(y_test,knn.predict(x_test)),annot=True)
```

```
<Axes: >
```




```
crknn=classification_report(y_test,knn.predict(x_test))
print(crknn)
```

```
              precision    recall  f1-score   support

           B       0.96      0.96      0.96        67
           M       0.94      0.94      0.94        47

    accuracy                           0.95       114
   macro avg       0.95      0.95      0.95       114
weighted avg       0.95      0.95      0.95       114
```

```
from sklearn.naive_bayes import GaussianNB
nb=GaussianNB()
nb.fit(x_train,y_train)
```

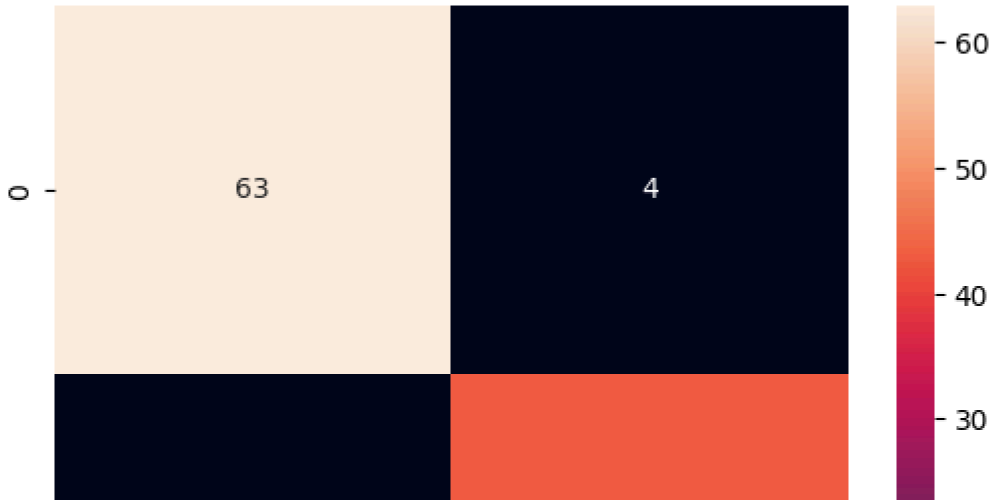
```
▾ GaussianNB
GaussianNB()
```

```
acnb=accuracy_score(y_test,nb.predict(x_test))
print(acnb)
```

```
0.9298245614035088
```

```
sns.heatmap(confusion_matrix(y_test,nb.predict(x_test)),annot=True)
```

```
<Axes: >
```




```
crnb=classification_report(y_test,nb.predict(x_test))
print(crnb)
```

```
              precision    recall  f1-score   support

           B       0.94      0.94      0.94        67
           M       0.91      0.91      0.91        47

    accuracy                           0.93       114
   macro avg       0.93      0.93      0.93       114
weighted avg       0.93      0.93      0.93       114
```

SVM

```
from sklearn.svm import SVC
svmmodel=SVC(kernel="linear")
svmmodel.fit(x_train,y_train)
```

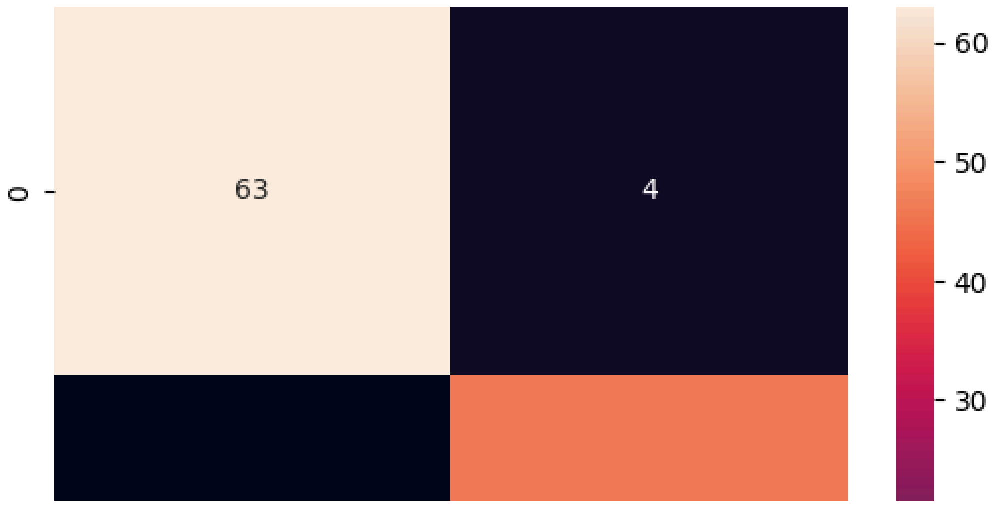
```
SVC
SVC(kernel='linear')
```

```
acsvm=accuracy_score(y_test,svmmodel.predict(x_test))
print(acsvm)
```

```
0.956140350877193
```

```
sns.heatmap(confusion_matrix(y_test,svmmodel.predict(x_test)),annot=True)
```

```
<Axes: >
```




```
crsvm=classification_report(y_test,svmmodel.predict(x_test))
print(crsvm)
```

```
              precision    recall  f1-score   support

           B       0.98      0.94      0.96        67
           M       0.92      0.98      0.95        47

    accuracy                           0.96       114
   macro avg       0.95      0.96      0.96       114
weighted avg       0.96      0.96      0.96       114
```

DT

```
from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
```

```
dt.fit(x_train,y_train)
```

DecisionTreeClassifier

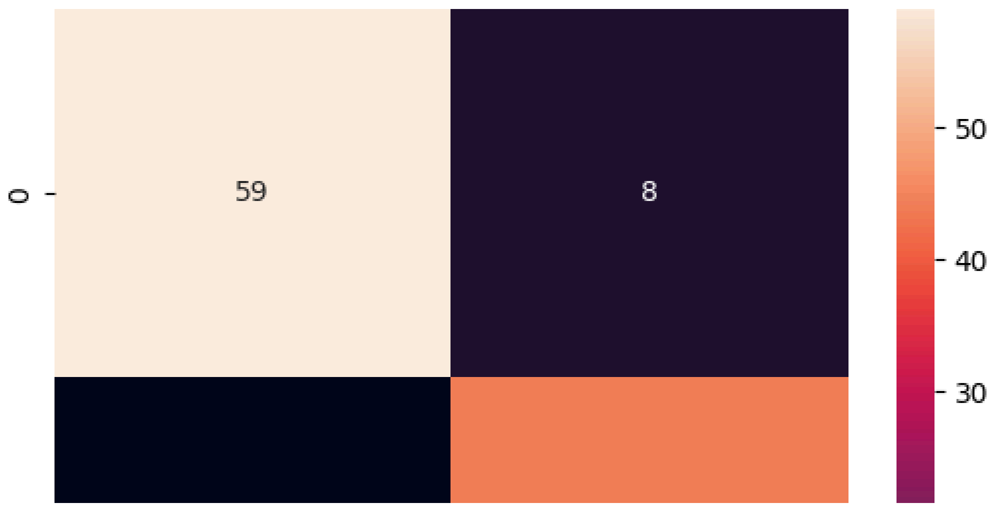
DecisionTreeClassifier()

```
acdt=accuracy_score(y_test,dt.predict(x_test))
print(acdt)
```

0.9035087719298246

```
sns.heatmap(confusion_matrix(y_test,dt.predict(x_test)),annot=True)
```

<Axes: >




```
crdt=classification_report(y_test,dt.predict(x_test))
print(crdt)
```

```
              precision    recall  f1-score   support

           B       0.95      0.88      0.91        67
           M       0.85      0.94      0.89        47

    accuracy                           0.90       114
   macro avg       0.90      0.91      0.90       114
weighted avg       0.91      0.90      0.90       114
```

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(n_estimators=100)
rf.fit(x_train,y_train)
```

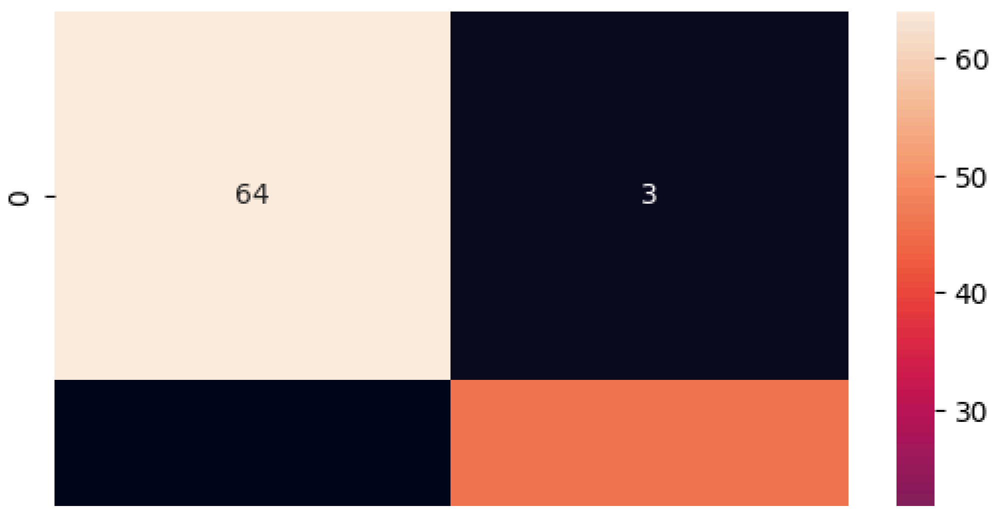
```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
acrf=accuracy_score(y_test,rf.predict(x_test))
print(acrf)
```

```
0.9649122807017544
```

```
sns.heatmap(confusion_matrix(y_test,rf.predict(x_test)),annot=True)
```

```
<Axes: >
```




```
crrf=classification_report(y_test,rf.predict(x_test))
print(crrf)
```

```
              precision    recall  f1-score   support

           B       0.98      0.96      0.97        67
           M       0.94      0.98      0.96        47
```

```
    accuracy                           0.96       114
   macro avg       0.96      0.97      0.96       114
weighted avg       0.97      0.96      0.97       114
```