breastcancer

January 5, 2026

[94]:
```
import pandas as pd
import numpy as np
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix,␣
 ↪roc_auc_score, roc_curve,accuracy_score, recall_score
```

[106]:
```
df = pd.read_csv(r"C:\Users\karan\Downloads\Breast_Cancer_data.csv")
```

[108]:
```
df
```

[108]:
```
           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
..                      …            …
564                  0.07115          NaN
565                  0.06637          NaN
566                  0.07820          NaN
567                  0.12400          NaN
568                  0.07039          NaN
```

```
[569 rows x 33 columns]
```

[110]:
```
skewness_values = df.skew(numeric_only=True)
print(skewness_values)
```

```
id                         6.473752
radius_mean                0.942380
texture_mean               0.650450
perimeter_mean             0.990650
area_mean                  1.645732
smoothness_mean            0.456324
compactness_mean           1.190123
concavity_mean             1.401180
concave points_mean        1.171180
symmetry_mean              0.725609
fractal_dimension_mean     1.304489
radius_se                  3.088612
texture_se                 1.646444
perimeter_se               3.443615
area_se                    5.447186
smoothness_se              2.314450
compactness_se             1.902221
concavity_se               5.110463
concave points_se          1.444678
symmetry_se                2.195133
fractal_dimension_se       3.923969
radius_worst               1.103115
texture_worst              0.498321
perimeter_worst            1.128164
area_worst                 1.859373
smoothness_worst           0.415426
compactness_worst          1.473555
concavity_worst            1.150237
concave points_worst       0.492616
symmetry_worst             1.433928
fractal_dimension_worst    1.662579
Unnamed: 32                     NaN
dtype: float64
```

[112]:
```
df.shape
```

[112]:
```
(569, 33)
```

[114]:
```
df.isnull().sum()
```

[114]:
```
id                         0
diagnosis                  0
radius_mean                0
```

```
texture_mean                 0
perimeter_mean               0
area_mean                    0
smoothness_mean              0
compactness_mean             0
concavity_mean               0
concave points_mean          0
symmetry_mean                0
fractal_dimension_mean       0
radius_se                    0
texture_se                   0
perimeter_se                 0
area_se                      0
smoothness_se                0
compactness_se               0
concavity_se                 0
concave points_se            0
symmetry_se                  0
fractal_dimension_se         0
radius_worst                 0
texture_worst                0
perimeter_worst              0
area_worst                   0
smoothness_worst             0
compactness_worst            0
concavity_worst              0
concave points_worst         0
symmetry_worst               0
fractal_dimension_worst      0
Unnamed: 32                569
dtype: int64
```

[116]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

[69]:
```
df.describe().T
```

[69]:
```
                         count        mean         std         min  \
radius_mean              569.0   14.062916    3.340025    6.981000
texture_mean             569.0   19.254736    4.187510    9.710000
perimeter_mean           569.0   91.543787   23.047218   43.790000
area_mean                569.0  639.765202  305.343508  143.500000
smoothness_mean          569.0    0.096266    0.013685    0.057975
compactness_mean         569.0    0.103222    0.049386    0.019380
concavity_mean           569.0    0.086937    0.073900    0.000000
concave points_mean      569.0    0.048552    0.037633    0.000000
symmetry_mean            569.0    0.180734    0.026067    0.111200
fractal_dimension_mean   569.0    0.062604    0.006418    0.049960
radius_se                569.0    0.384698    0.203612    0.111500
texture_se               569.0    1.198057    0.485500    0.360200
perimeter_se             569.0    2.699075    1.402982    0.757000
area_se                  569.0   34.959487   24.294515    6.802000
smoothness_se            569.0    0.006876    0.002410    0.001713
compactness_se           569.0    0.024561    0.014947    0.002252
concavity_se             569.0    0.030038    0.020577    0.000000
```

```
concave points_se        569.0    0.011601    0.005486    0.000000
symmetry_se              569.0    0.020047    0.006572    0.007882
fractal_dimension_se     569.0    0.003591    0.001780    0.000895
radius_worst             569.0   16.183882    4.587249    7.930000
texture_worst            569.0   25.648453    6.054406   12.020000
perimeter_worst          569.0  106.705369   31.957777   50.410000
area_worst               569.0  849.907821  475.645240  185.200000
smoothness_worst         569.0    0.132209    0.022320    0.072500
compactness_worst        569.0    0.249883    0.142851    0.027290
concavity_worst          569.0    0.268754    0.197461    0.000000
concave points_worst     569.0    0.114606    0.065732    0.000000
symmetry_worst           569.0    0.287616    0.053868    0.156500
fractal_dimension_worst  569.0    0.083342    0.015993    0.055040

                                25%         50%          75%          max
radius_mean               11.700000   13.370000    15.780000    21.900000
texture_mean              16.170000   18.840000    21.800000    30.245000
perimeter_mean            75.170000   86.240000   104.100000   147.495000
area_mean                420.300000  551.100000   782.700000  1326.300000
smoothness_mean            0.086370    0.095870     0.105300     0.133695
compactness_mean           0.064920    0.092630     0.130400     0.228620
concavity_mean             0.029560    0.061540     0.130700     0.282410
concave points_mean        0.020310    0.033500     0.074000     0.154535
symmetry_mean              0.161900    0.179200     0.195700     0.246400
fractal_dimension_mean     0.057700    0.061540     0.066120     0.078750
radius_se                  0.232400    0.324200     0.478900     0.848650
texture_se                 0.833900    1.108000     1.474000     2.434150
perimeter_se               1.606000    2.287000     3.357000     5.983500
area_se                   17.850000   24.530000    45.190000    86.200000
smoothness_se              0.005169    0.006380     0.008146     0.012612
compactness_se             0.013080    0.020450     0.032450     0.061505
concavity_se               0.015090    0.025890     0.042050     0.082490
concave points_se          0.007638    0.010930     0.014710     0.025318
symmetry_se                0.015160    0.018730     0.023480     0.035960
fractal_dimension_se       0.002248    0.003187     0.004558     0.008023
radius_worst              13.010000   14.970000    18.790000    27.460000
texture_worst             21.080000   25.410000    29.720000    42.680000
perimeter_worst           84.110000   97.660000   125.400000   187.335000
area_worst               515.300000  686.500000  1084.000000  1937.050000
smoothness_worst           0.116600    0.131300     0.146000     0.190100
compactness_worst          0.147200    0.211900     0.339100     0.626950
concavity_worst            0.114500    0.226700     0.382900     0.785500
concave points_worst       0.064930    0.099930     0.161400     0.291000
symmetry_worst             0.250400    0.282200     0.317900     0.419150
fractal_dimension_worst    0.071460    0.080040     0.092080     0.123010
```

[118]:
```
df= df.drop(["Unnamed: 32","id"], axis = 1)
```

```
[120]: df
```

```
[120]:    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_worst  texture_worst  perimeter_worst  \
     0           0.2419  …        25.380          17.33           184.60
     1           0.1812  …        24.990          23.41           158.80
     2           0.2069  …        23.570          25.53           152.50
     3           0.2597  …        14.910          26.50            98.87
     4           0.1809  …        22.540          16.67           152.20
     ..             …  …           …              …               …
     564         0.1726  …        25.450          26.40           166.10
     565         0.1752  …        23.690          38.25           155.00
     566         0.1590  …        18.980          34.12           126.70
     567         0.2397  …        25.740          39.42           184.60
     568         0.1587  …         9.456          30.37            59.16

          area_worst  smoothness_worst  compactness_worst  concavity_worst  \
     0        2019.0           0.16220            0.66560           0.7119
     1        1956.0           0.12380            0.18660           0.2416
     2        1709.0           0.14440            0.42450           0.4504
     3         567.7           0.20980            0.86630           0.6869
     4        1575.0           0.13740            0.20500           0.4000
```

```
..        …            …             …           …
564     2027.0      0.14100        0.21130         0.4107
565     1731.0      0.11660        0.19220         0.3215
566     1124.0      0.11390        0.30940         0.3403
567     1821.0      0.16500        0.86810         0.9387
568      268.6      0.08996        0.06444         0.0000

     concave points_worst  symmetry_worst  fractal_dimension_worst
0                  0.2654          0.4601                  0.11890
1                  0.1860          0.2750                  0.08902
2                  0.2430          0.3613                  0.08758
3                  0.2575          0.6638                  0.17300
4                  0.1625          0.2364                  0.07678
..                    …              …                       …
564                0.2216          0.2060                  0.07115
565                0.1628          0.2572                  0.06637
566                0.1418          0.2218                  0.07820
567                0.2650          0.4087                  0.12400
568                0.0000          0.2871                  0.07039

[569 rows x 31 columns]
```

[122]:
```
df["diagnosis"] = df["diagnosis"].astype("category")
```

[124]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   diagnosis                569 non-null    category
 1   radius_mean              569 non-null    float64
 2   texture_mean             569 non-null    float64
 3   perimeter_mean           569 non-null    float64
 4   area_mean                569 non-null    float64
 5   smoothness_mean          569 non-null    float64
 6   compactness_mean         569 non-null    float64
 7   concavity_mean           569 non-null    float64
 8   concave points_mean      569 non-null    float64
 9   symmetry_mean            569 non-null    float64
 10  fractal_dimension_mean   569 non-null    float64
 11  radius_se                569 non-null    float64
 12  texture_se               569 non-null    float64
 13  perimeter_se             569 non-null    float64
 14  area_se                  569 non-null    float64
 15  smoothness_se            569 non-null    float64
 16  compactness_se           569 non-null    float64
 17  concavity_se             569 non-null    float64
 18  concave points_se        569 non-null    float64
 19  symmetry_se              569 non-null    float64
 20  fractal_dimension_se     569 non-null    float64
 21  radius_worst             569 non-null    float64
 22  texture_worst            569 non-null    float64
 23  perimeter_worst          569 non-null    float64
 24  area_worst               569 non-null    float64
 25  smoothness_worst         569 non-null    float64
 26  compactness_worst        569 non-null    float64
 27  concavity_worst          569 non-null    float64
 28  concave points_worst     569 non-null    float64
 29  symmetry_worst           569 non-null    float64
 30  fractal_dimension_worst  569 non-null    float64
dtypes: category(1), float64(30)
memory usage: 134.2 KB
```

[126]:
```
num_cols = df.select_dtypes(include=["float","int"]).columns

for col in num_cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    upper_bound = Q3 + 1.5*IQR
    lower_bound = Q1 - 1.5*IQR
    df[col] = np.where(df[col] < lower_bound , lower_bound , df[col])
    df[col] = np.where(df[col] > upper_bound , upper_bound , df[col])
```

[128]:
```
df["diagnosis"].value_counts()
```

[128]:
```
diagnosis
B    357
M    212
Name: count, dtype: int64
```

[130]:
```
y = df['diagnosis'].map({'B':0, 'M':1})  # benign = 0, malignant = 1
X = df.drop(columns=['diagnosis'])
```

[132]:
```
# Identify numeric and categorical columns
num_cols = X.select_dtypes(include=['int64','float64']).columns.tolist()
cat_cols = X.select_dtypes(include=['object','category']).columns.tolist()
```

[134]:
```
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
```

```
[136]: num_transformer = Pipeline(steps=[('scaler',StandardScaler())])

       cat_transformer = Pipeline(steps=[('OneHot', OneHotEncoder(handle_unknown =␣
        ↪"ignore" , sparse_output = False))])

       preprocessor = ColumnTransformer(transformers=[('num',␣
        ↪num_transformer,num_cols),
                                                       ('cat',␣
        ↪cat_transformer,cat_cols)],
                                        remainder='drop'
                                       )
```

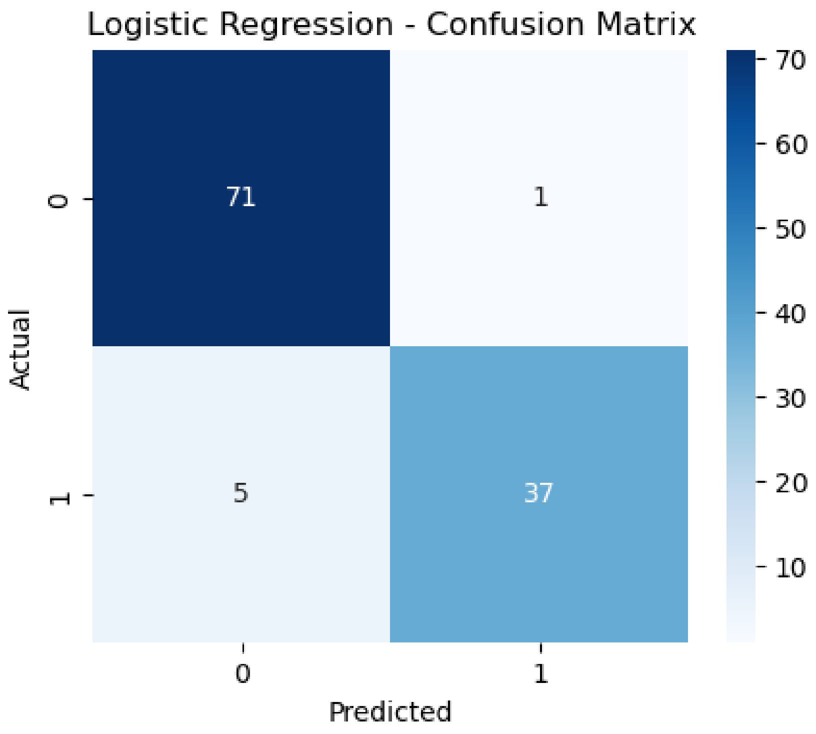
```
[138]: X_train, X_test,y_train,y_test= train_test_split(X,y,test_size=0.
        ↪2,random_state=42,stratify=y)
       print("Training set size:", X_train.shape)
       print("Test set size:", X_test.shape)
```

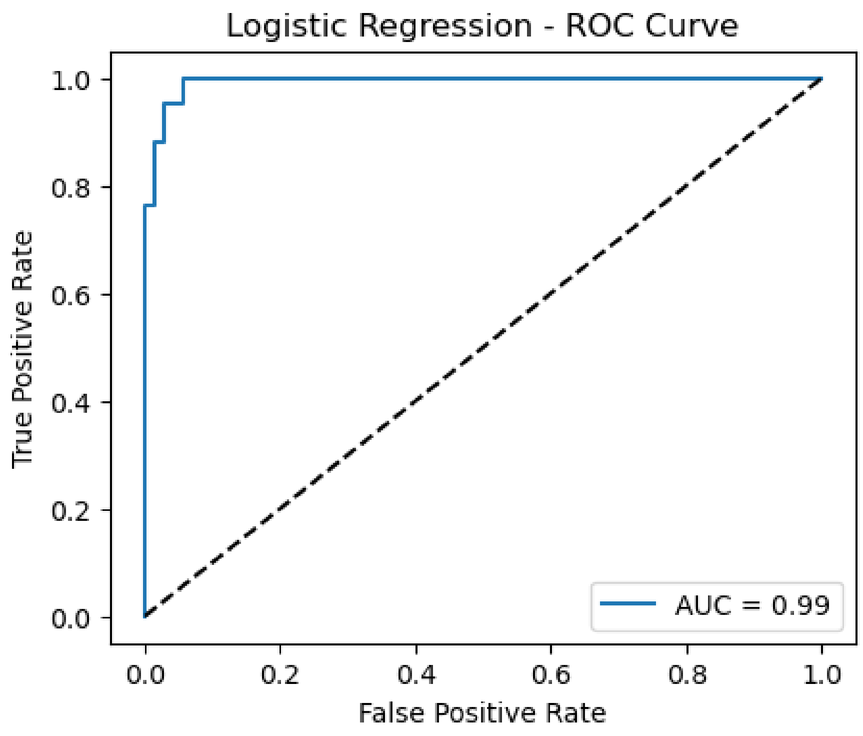
```
Training set size: (455, 30)
Test set size: (114, 30)
```

```
[140]: models = {
           "Logistic Regression": LogisticRegression(max_iter=1000),
           "Decision Tree": DecisionTreeClassifier(random_state=42),
           "Random Forest": RandomForestClassifier(random_state=42),
           "Gradient Boosting": GradientBoostingClassifier(random_state=42),
           "XGBoost": XGBClassifier(eval_metric='logloss', random_state=42),
           "SVM": SVC(probability=True, random_state=42)  # probability=True needed␣
        ↪for ROC/AUC
       }

       for name, model in models.items():
           # Fit model
           model.fit(X_train, y_train)

           # Predictions
           y_pred = model.predict(X_test)
           y_prob = model.predict_proba(X_test)[:,1]  # needed for ROC/AUC

           print(f"\n=== {name} ===")
           # Classification report
           print(classification_report(y_test, y_pred))

           # Confusion matrix
           cm = confusion_matrix(y_test, y_pred)
           plt.figure(figsize=(5,4))
           sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
           plt.title(f'{name} - Confusion Matrix')
```

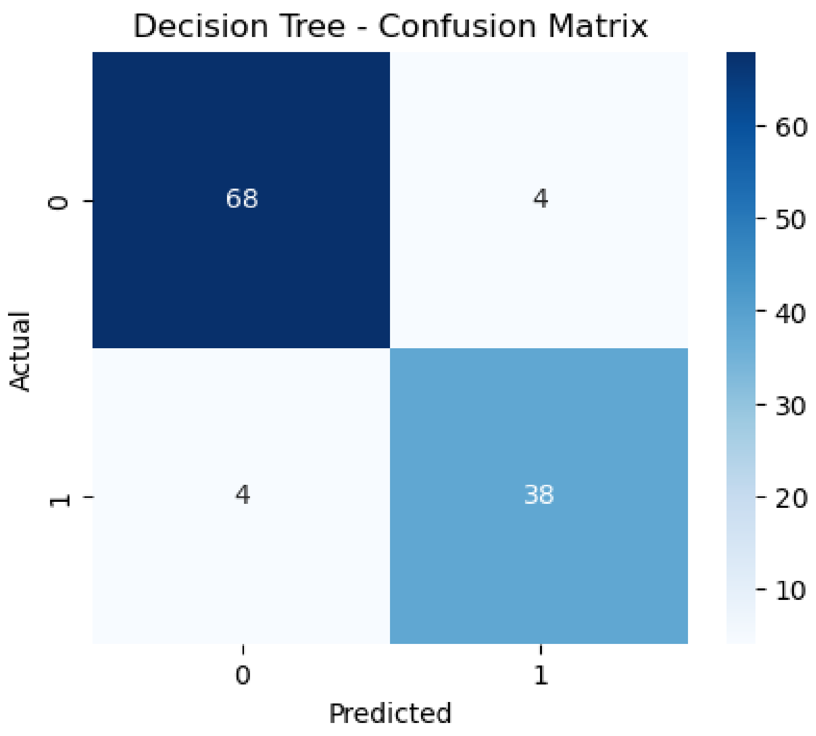
```python
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()

    # AUC-ROC
    auc = roc_auc_score(y_test, y_prob)
    fpr, tpr, thresholds = roc_curve(y_test, y_prob)
    plt.figure(figsize=(5,4))
    plt.plot(fpr, tpr, label=f'AUC = {auc:.2f}')
    plt.plot([0,1],[0,1],'k--')
    plt.title(f'{name} - ROC Curve')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.legend(loc='lower right')
    plt.show()
```

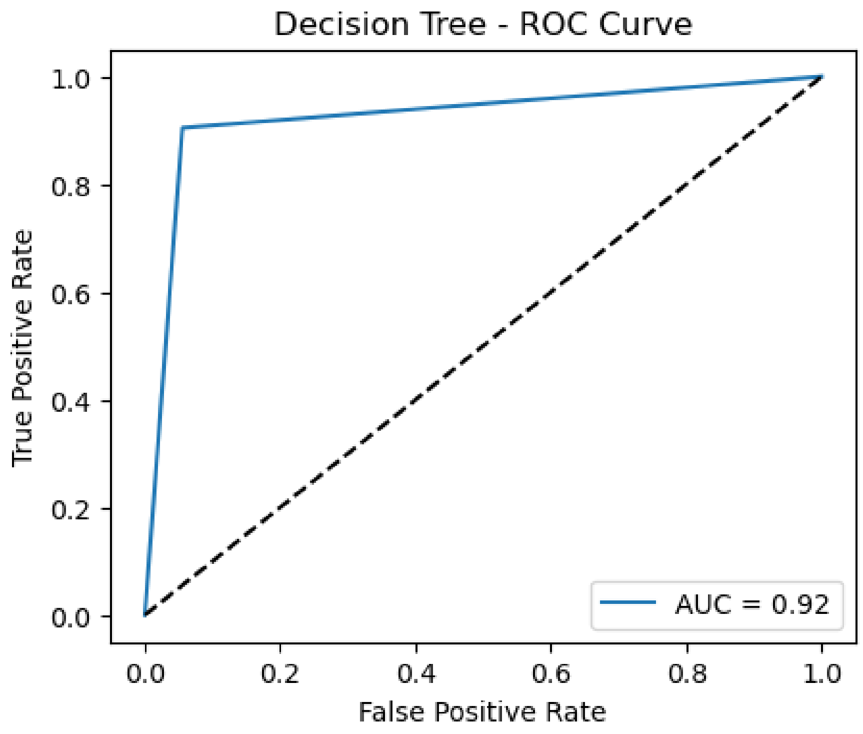
```
=== Logistic Regression ===
              precision    recall  f1-score   support

           0       0.93      0.99      0.96        72
           1       0.97      0.88      0.93        42

    accuracy                           0.95       114
   macro avg       0.95      0.93      0.94       114
weighted avg       0.95      0.95      0.95       114

C:\Users\karan\anaconda3\Lib\site-
packages\sklearn\linear_model\_logistic.py:469: ConvergenceWarning: lbfgs failed
to converge (status=1):
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-
regression
  n_iter_i = _check_optimize_result(
```


Logistic Regression - Confusion Matrix
Actual
Predicted
0
1
71
1
5
37
70
60
50
40
30
20
10

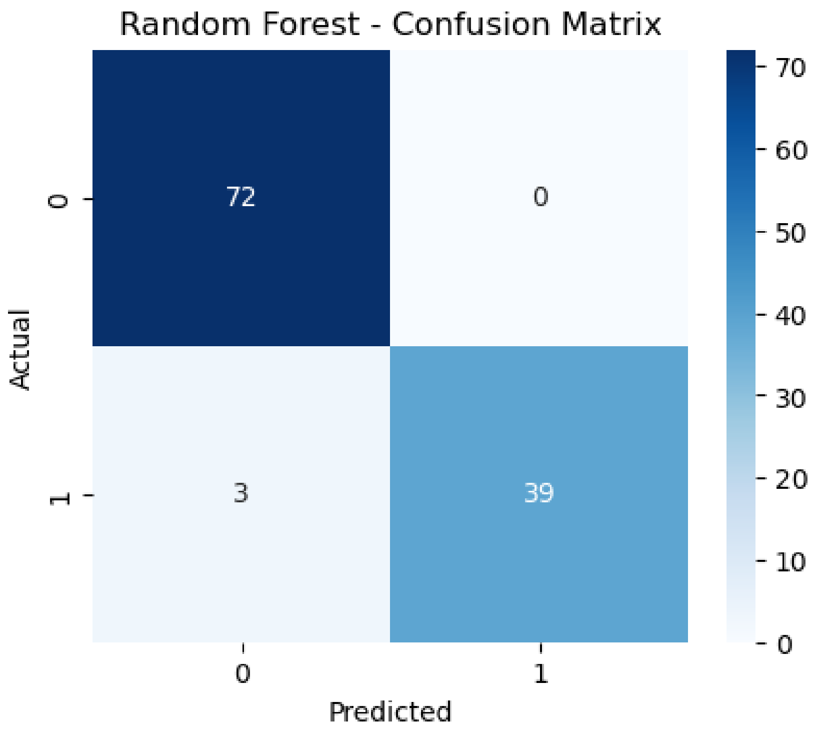

Logistic Regression - ROC Curve
True Positive Rate
False Positive Rate
AUC = 0.99
0.0
0.2
0.4
0.6
0.8
1.0

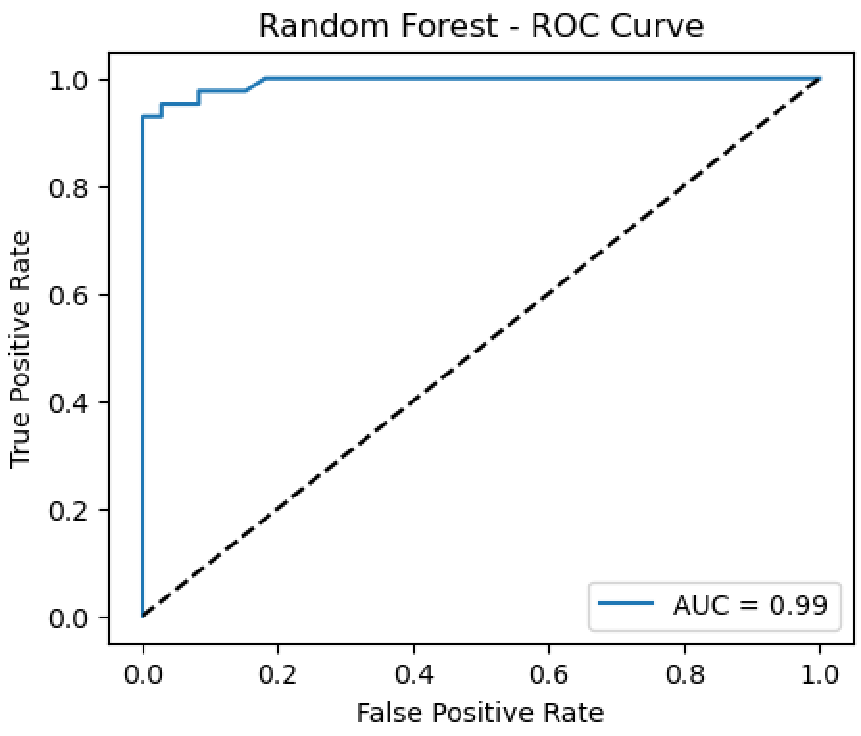
```
=== Decision Tree ===
              precision    recall  f1-score   support

           0       0.94      0.94      0.94        72
           1       0.90      0.90      0.90        42

    accuracy                           0.93       114
   macro avg       0.92      0.92      0.92       114
weighted avg       0.93      0.93      0.93       114
```



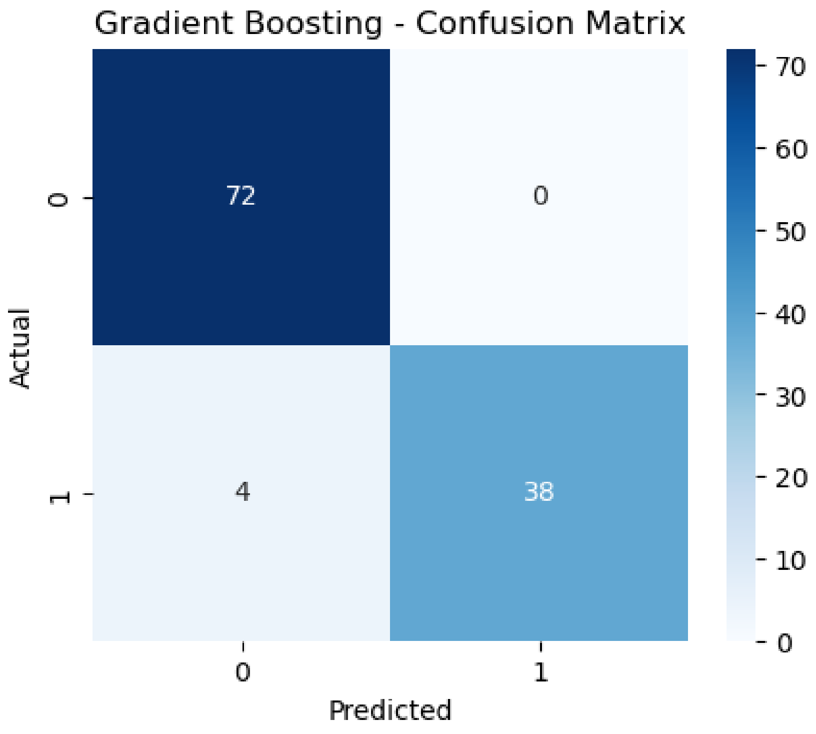


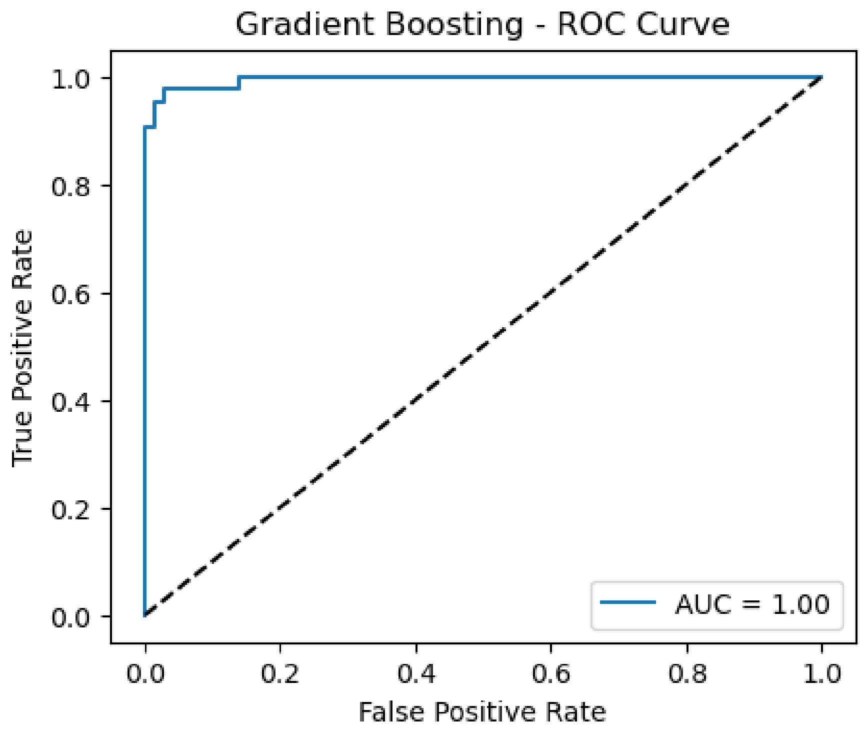
```
=== Random Forest ===
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        72
           1       1.00      0.93      0.96        42

    accuracy                           0.97       114
   macro avg       0.98      0.96      0.97       114
weighted avg       0.97      0.97      0.97       114
```


Random Forest - Confusion Matrix
Actual
Predicted
0
1
72
0
3
39
70
60
50
40
30
20
10
0

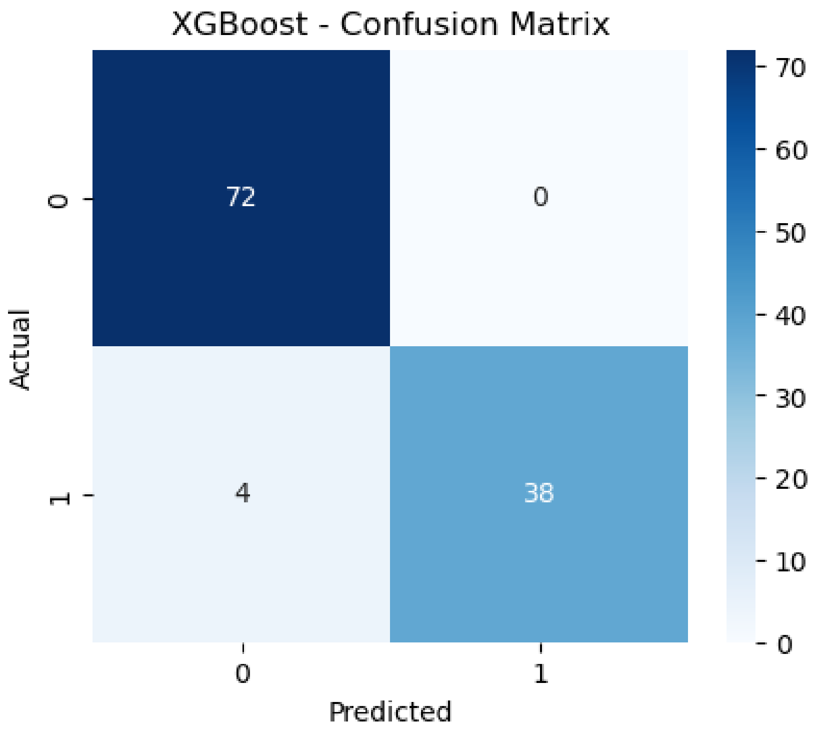

Random Forest - ROC Curve
True Positive Rate
False Positive Rate
AUC = 0.99
0.0
0.2
0.4
0.6
0.8
1.0

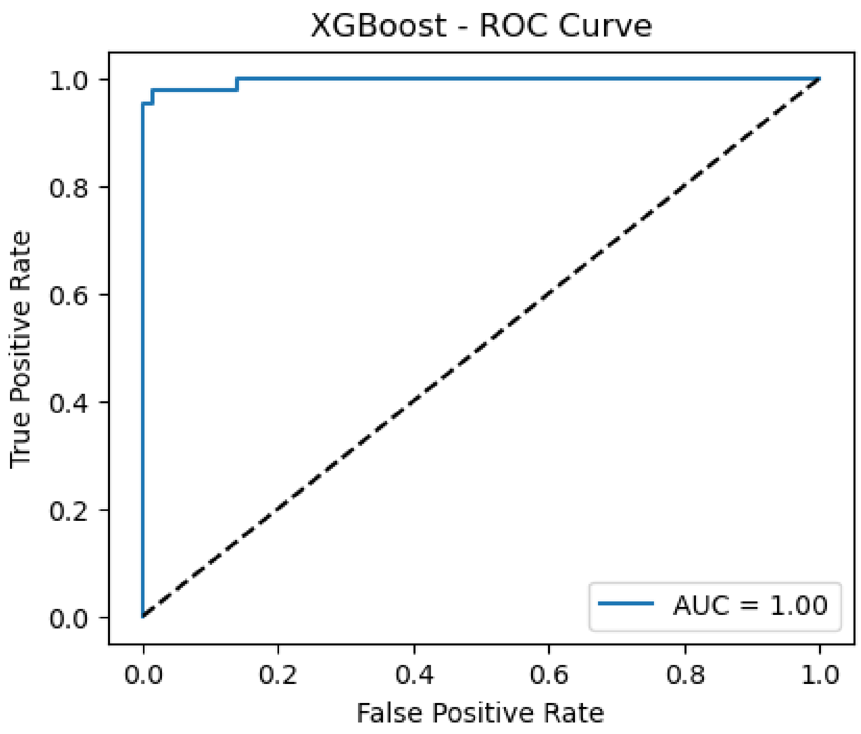
```
=== Gradient Boosting ===
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        72
           1       1.00      0.90      0.95        42

    accuracy                           0.96       114
   macro avg       0.97      0.95      0.96       114
weighted avg       0.97      0.96      0.96       114
```



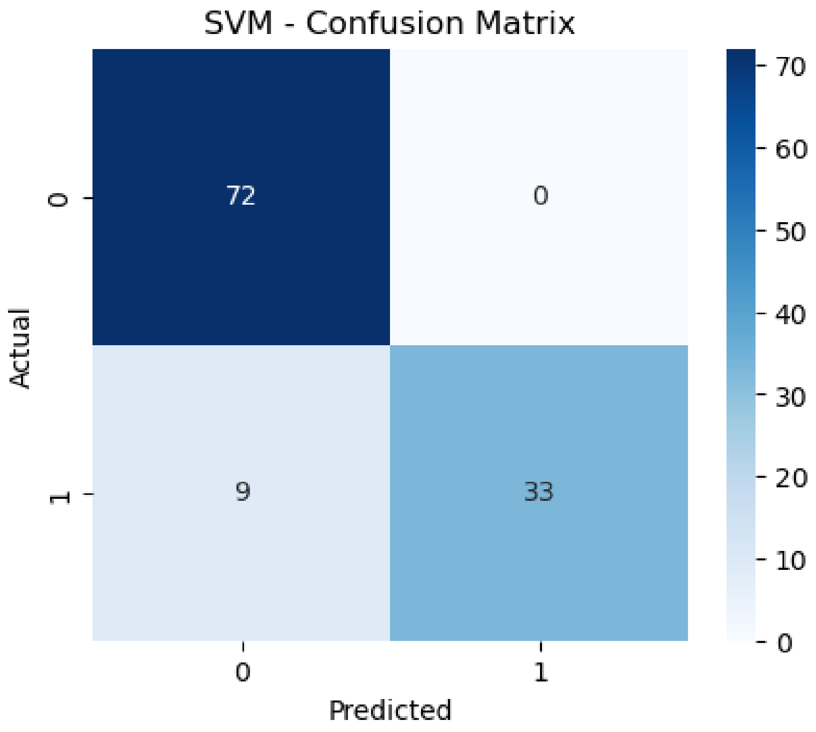


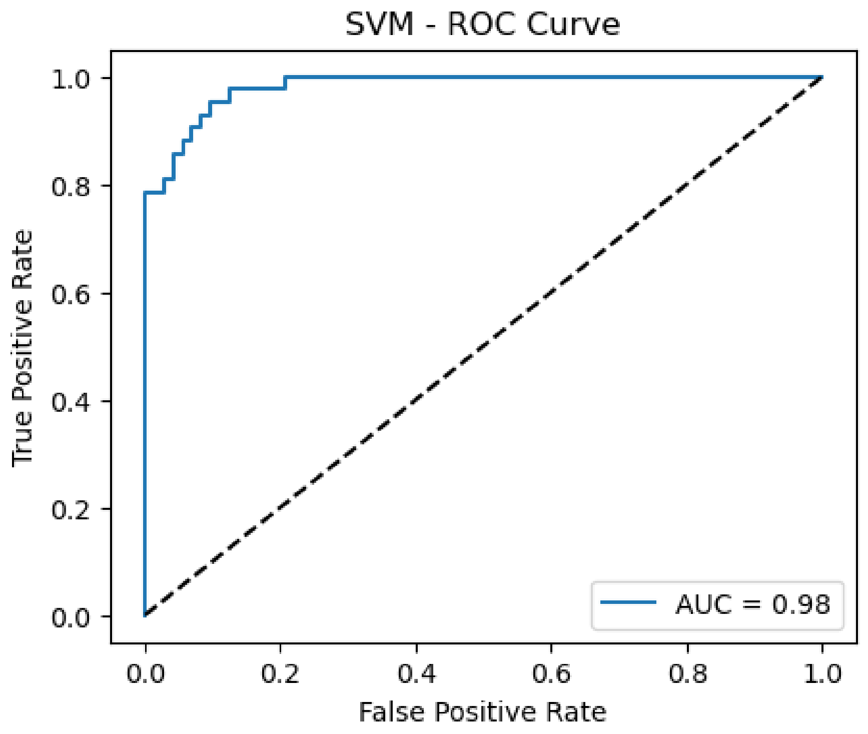
```
=== XGBoost ===
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        72
           1       1.00      0.90      0.95        42

    accuracy                           0.96       114
   macro avg       0.97      0.95      0.96       114
weighted avg       0.97      0.96      0.96       114
```


XGBoost - Confusion Matrix
Actual
Predicted
72
0
4
38
0
1
0
1
70
60
50
40
30
20
10
0


XGBoost - ROC Curve
True Positive Rate
False Positive Rate
AUC = 1.00
0.0
0.2
0.4
0.6
0.8
1.0

```
=== SVM ===
              precision    recall  f1-score   support

           0       0.89      1.00      0.94        72
           1       1.00      0.79      0.88        42

    accuracy                           0.92       114
   macro avg       0.94      0.89      0.91       114
weighted avg       0.93      0.92      0.92       114
```




SVM - ROC Curve
True Positive Rate
False Positive Rate
AUC = 0.98
0.0
0.2
0.4
0.6
0.8
1.0

```
[102]:
```

```
[ ]:
```