```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, precision_score, recall_score, roc_auc_score, roc_curve
```

```
df=pd.read_csv('/bin/data.csv')
```

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

```
df['diagnosis'] = df['diagnosis'].map({'M': 1, 'B': 0})
```

```
X = df.drop('diagnosis', axis=1)
y = df['diagnosis']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

+ Code + Text

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train_scaled, y_train)
```

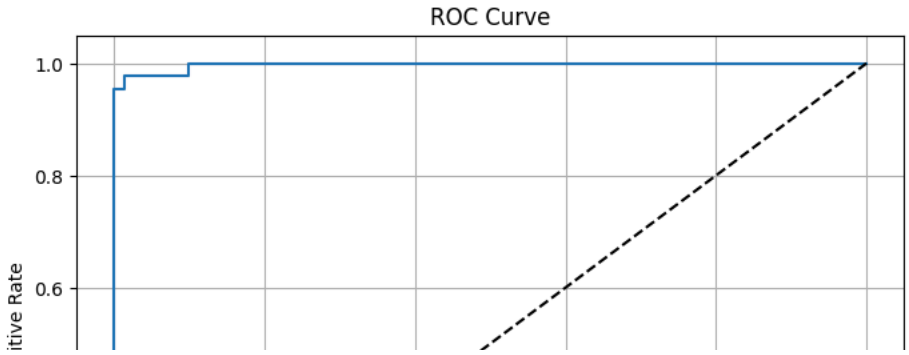
```
LogisticRegression(max_iter=1000)
```

```
y_probs = model.predict_proba(X_test_scaled)[:, 1]
y_pred = (y_probs >= 0.5).astype(int)
```

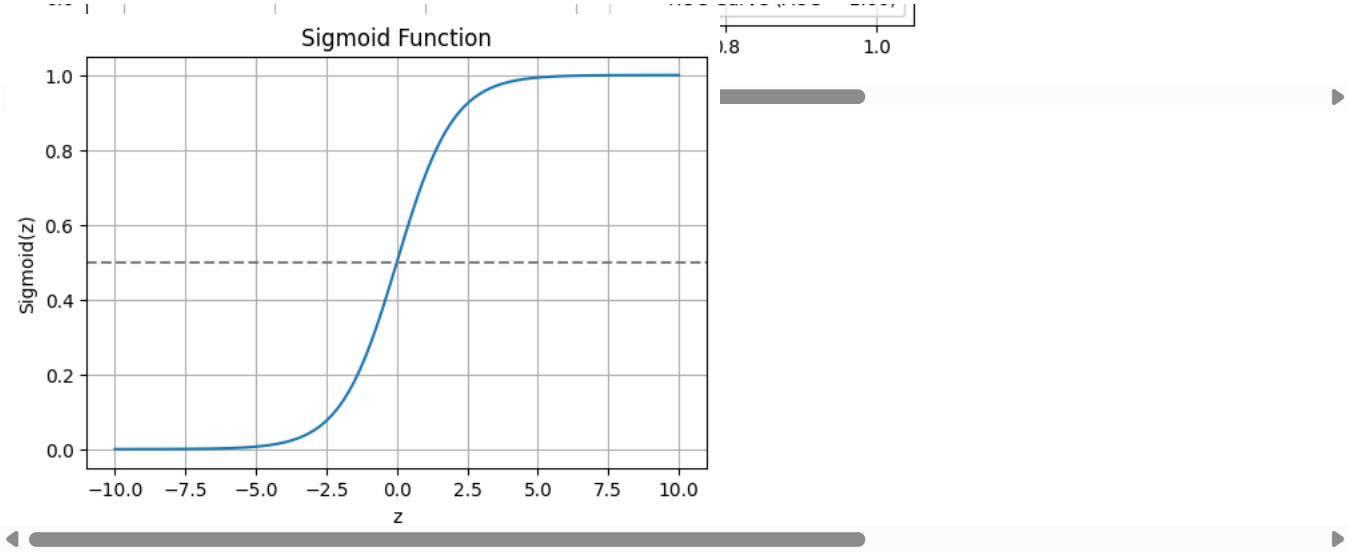
```
conf_mat = confusion_matrix(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_probs)
print("Confusion Matrix:\n", conf_mat)
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"ROC-AUC Score: {roc_auc:.2f}")
```

```
Confusion Matrix:
 [[70  1]
 [ 2 41]]
Precision: 0.98
Recall: 0.95
ROC-AUC Score: 1.00
```

```
fpr, tpr, thresholds = roc_curve(y_test, y_probs)
plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, label=f"ROC Curve (AUC = {roc_auc:.2f})")
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.grid(True)
plt.show()
```




```
z = np.linspace(-10, 10, 100)
sigmoid = 1 / (1 + np.exp(-z))
plt.figure(figsize=(6,4))
plt.plot(z, sigmoid)
plt.axhline(0.5, ls='--', c='gray')
plt.title("Sigmoid Function")
plt.xlabel("z")
plt.ylabel("Sigmoid(z)")
plt.grid(True)
plt.show()
```




```
df.to_csv('Task-4.csv', index=False)
```