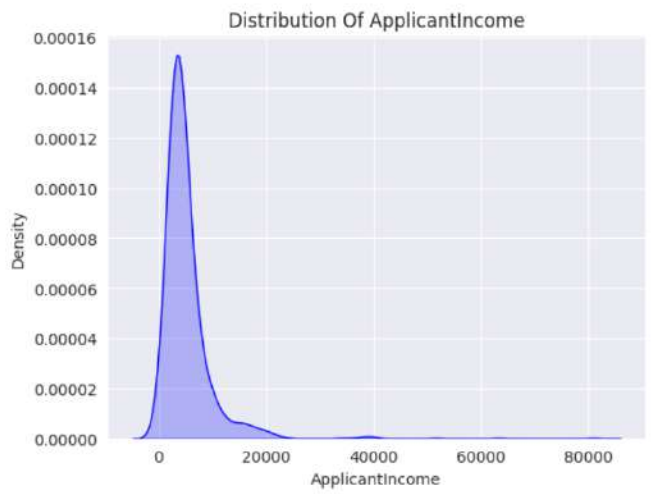
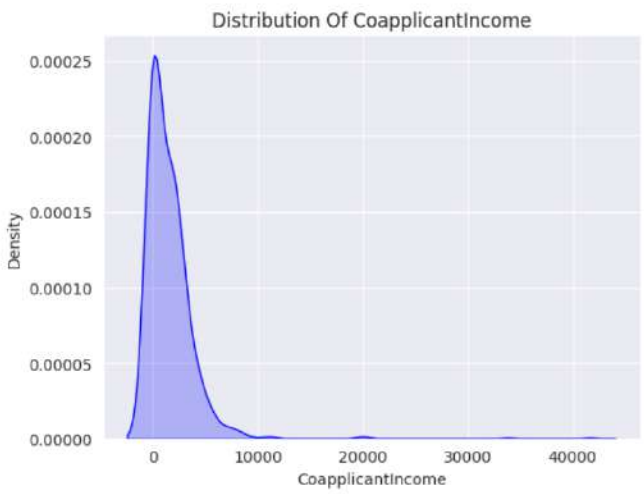
```
plt.tight_layout(pad=4.0)

plt.show()
```

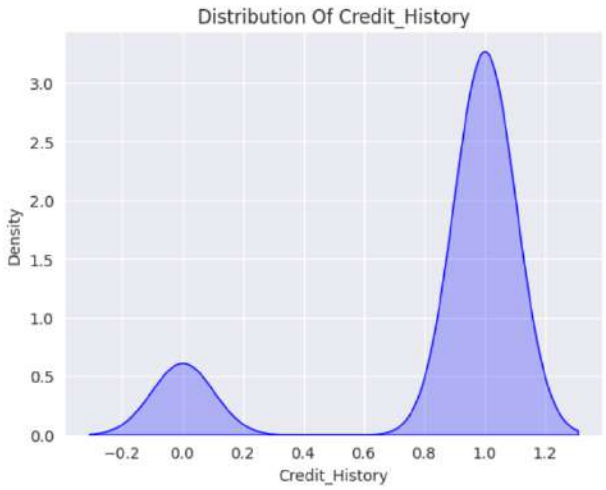




Distribution Of LoanAmount

Distribution Of Loan_Amount_Term

LoanAmount Loan_Amount_Term



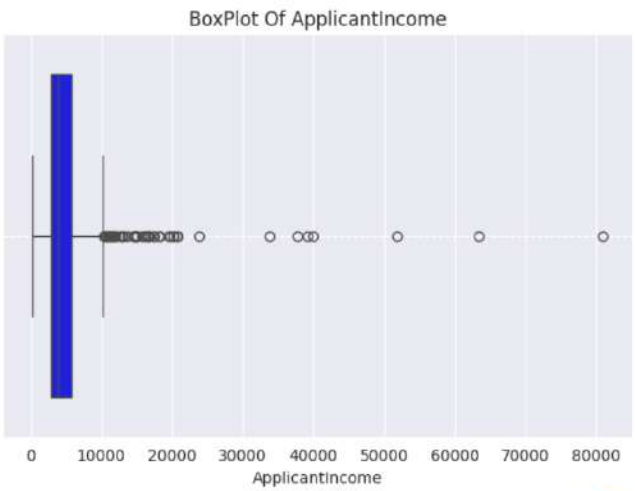
[24]
✓ 0s

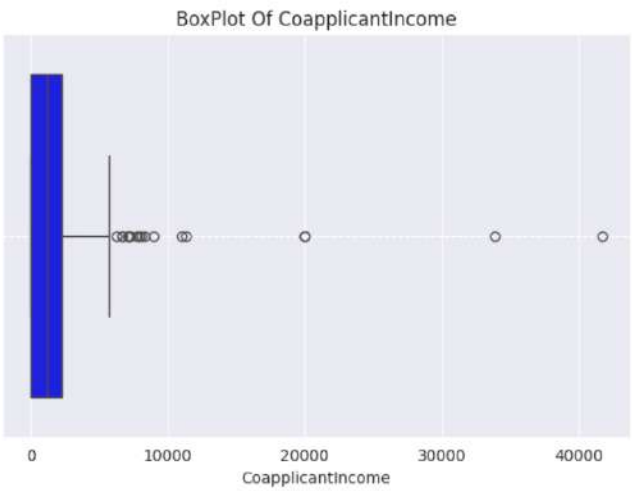
```
plt.figure(figsize=(13,15))
for c, cols in enumerate(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term',
```

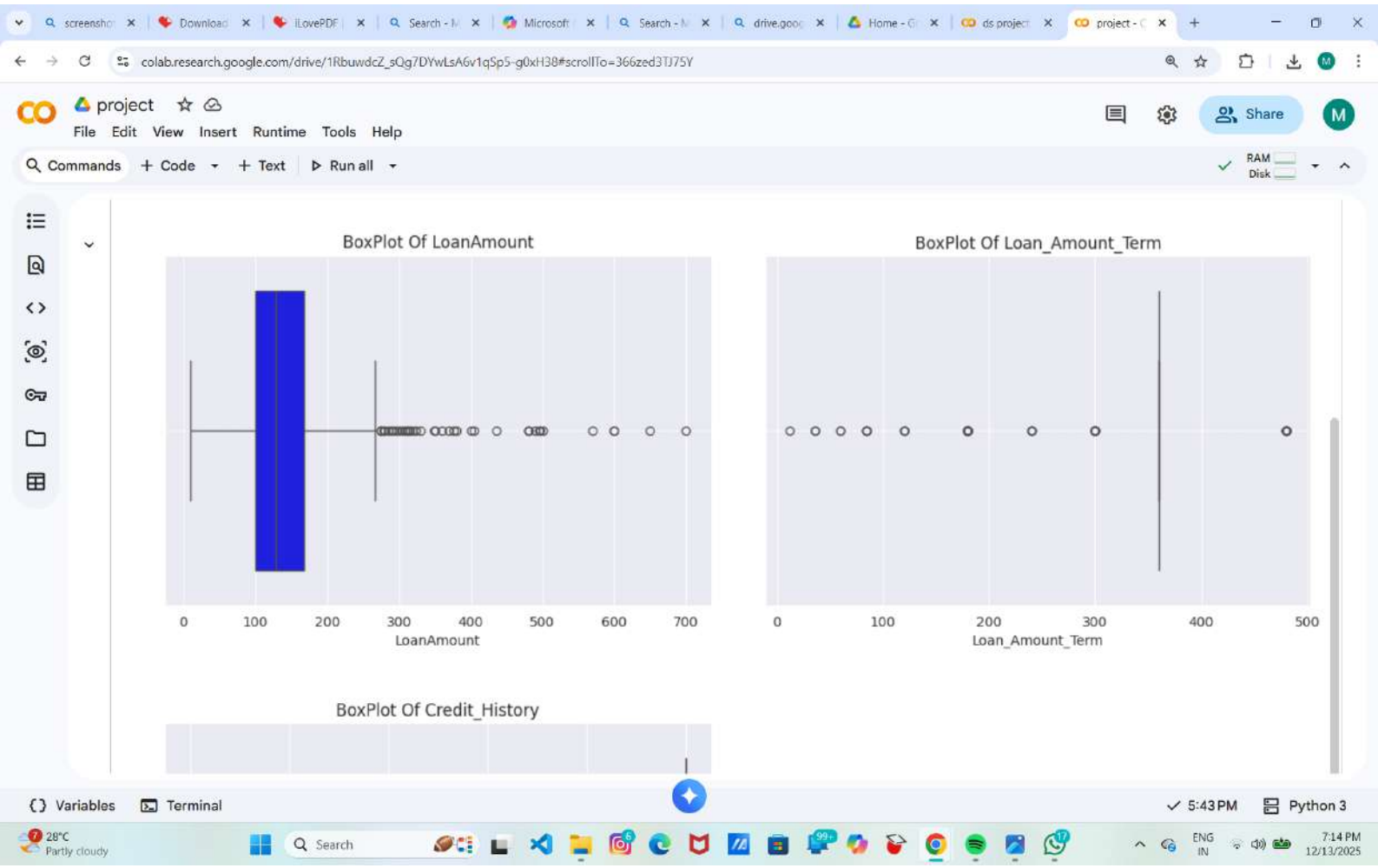
Variables Terminal 5:43 PM Python 3

```
    plt.subplot(3, 2, c+1)
    sns.boxplot(x=cols, data=df, color='blue')
    plt.title(f"BoxPlot Of {cols}")
    plt.grid(axis='y', linestyle='--')
    plt.tight_layout(pad=4.0)

plt.show()
```






BoxPlot Of LoanAmount
0
100
200
300
400
500
600
700
LoanAmount
BoxPlot Of Loan_Amount_Term
0
100
200
300
400
500
Loan_Amount_Term
BoxPlot Of Credit_History

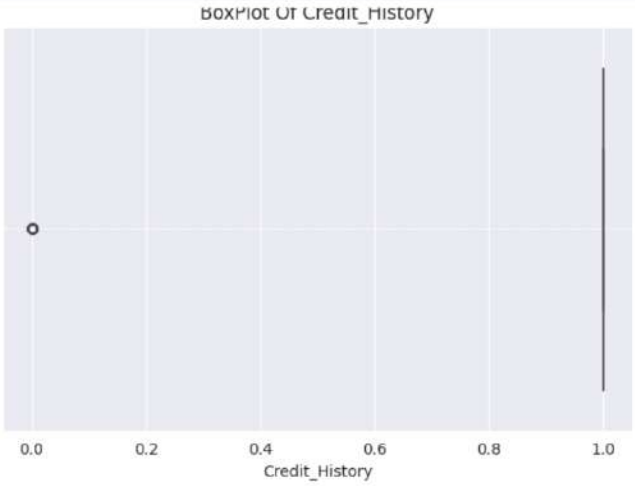
project

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all

RAM Disk



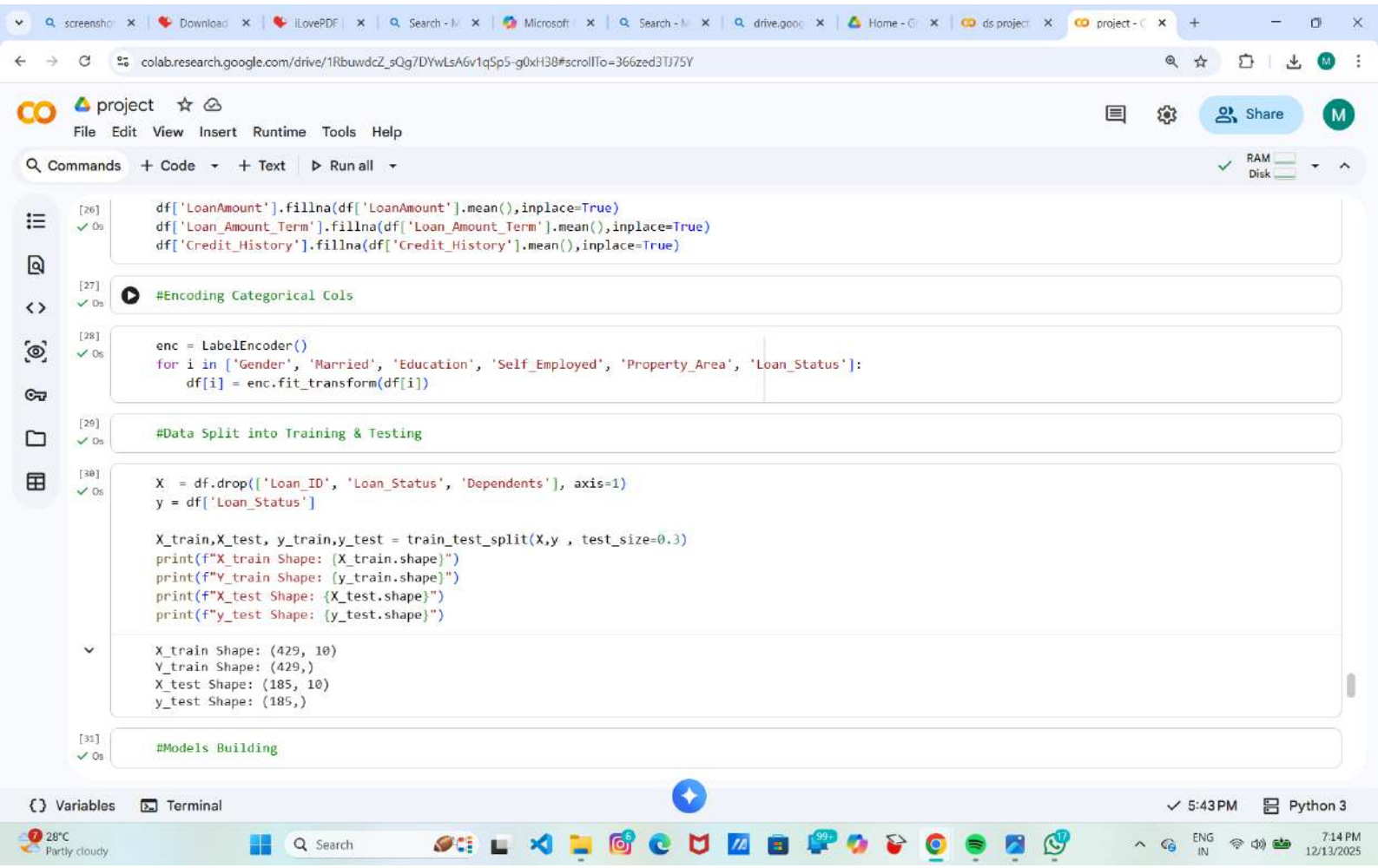
[25] 0s

```
#Fill Null Values
```

[26] 0s

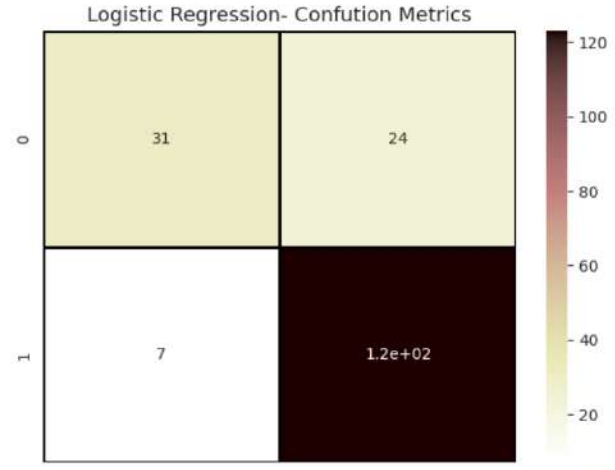
```
df['Gender'].fillna(df['Gender'].mode()[0], inplace=True)
df['Married'].fillna(df['Married'].mode()[0],inplace=True)
df['Dependents'].fillna(df['Dependents'].mode()[0],inplace=True)
df['Self_Employed'].fillna(df['Self_Employed'].mode()[0],inplace=True)
```

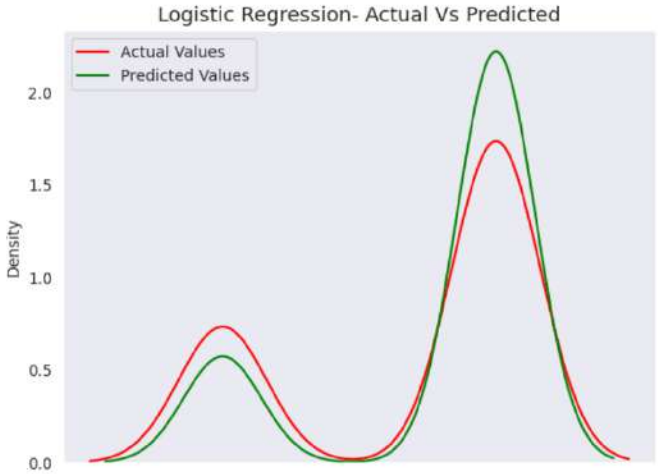
Variables Terminal 5:43 PM Python 3


project
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM
Disk
[26] df['LoanAmount'].fillna(df['LoanAmount'].mean(),inplace=True)
df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mean(),inplace=True)
df['Credit_History'].fillna(df['Credit_History'].mean(),inplace=True)
[27] #Encoding Categorical Cols
[28] enc = LabelEncoder()
for i in ['Gender', 'Married', 'Education', 'Self_Employed', 'Property_Area', 'Loan_Status']:
df[i] = enc.fit_transform(df[i])
[29] #Data Split into Training & Testing
[30] X = df.drop(['Loan_ID', 'Loan_Status', 'Dependents'], axis=1)
y = df['Loan_Status']
X_train,X_test, y_train,y_test = train_test_split(X,y , test_size=0.3)
print(f"X_train Shape: {X_train.shape}")
print(f"Y_train Shape: {y_train.shape}")
print(f"X_test Shape: {X_test.shape}")
print(f"y_test Shape: {y_test.shape}")
X_train Shape: (429, 10)
Y_train Shape: (429,)
X_test Shape: (185, 10)
y_test Shape: (185,)
[31] #Models Building
Variables
Terminal
5:43PM
Python 3
28°C
Partly cloudy
Search
ENG
IN
7:14 PM
12/13/2025

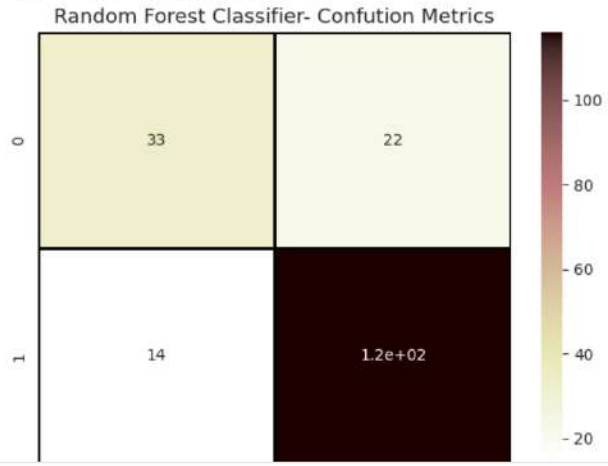
```
    plt.title(f'{names}- Actual Vs Predicted', fontsize=13)
    plt.grid()
    plt.legend()
    plt.show()
```

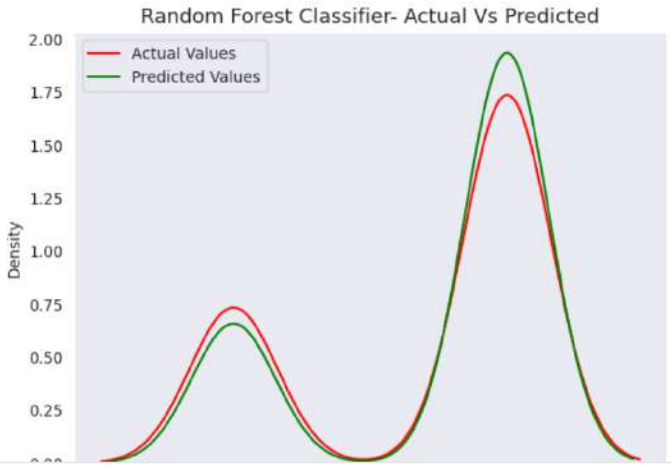
```
Name: Logistic Regression
Accuracy: 0.83
Precision Score: 0.8367346938775511
```



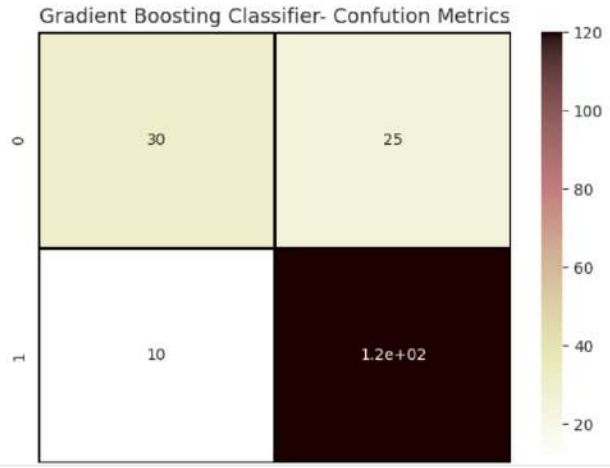


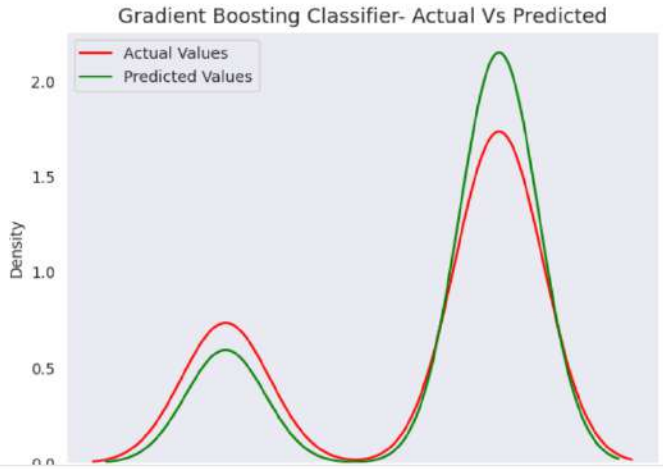
```
Name: Random Forest Classifier
Accuracy: 0.81
Precision Score: 0.8405797101449275
```





```
Name: Gradient Boosting Classifier
Accuracy: 0.81
Precision Score: 0.8275862068965517
```





```
Name: Naive Bayes
Accuracy: 0.82
Precision Score: 0.8439716312056738
```