

```

1 import pandas as pd
2 df = pd.read_csv('/content/defect.csv')
3 df.head()

```

	ProductionVolume	ProductionCost	SupplierQuality	DeliveryDelay	DefectRate	QualityScore	MaintenanceHours	DowntimePercentage
0	202	13175.403783	86.648534	1	3.121492	63.463494	9	0.052343
1	535	19770.046093	86.310664	4	0.819531	83.697818	20	4.908328
2	960	19060.820997	82.132472	0	4.514504	90.350550	1	2.464923
3	370	5647.606037	87.335966	5	0.638524	67.628690	8	4.692476
4	206	7472.222236	81.989893	3	3.867784	82.728334	9	2.746726

Next steps:

[Generate code with df](#)

[View recommended plots](#)

```

1 from sklearn.model_selection import train_test_split
2 from sklearn.naive_bayes import GaussianNB
3 from sklearn.metrics import accuracy_score, confusion_matrix, roc_curve, auc, classification_report
4 import matplotlib.pyplot as plt

```

```

1 # Split the data into train and test sets
2 X = df.drop('DefectStatus', axis=1)
3 y = df['DefectStatus']
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

```

```

1 # Train the Naive Bayes model
2 model = GaussianNB()
3 model.fit(X_train, y_train)

```



▼ GaussianNB
GaussianNB()

```

1 # Make predictions on the test set
2 y_pred = model.predict(X_test)

```

```

1 # Calculate the model performance
2 accuracy = accuracy_score(y_test, y_pred)
3 print("Accuracy:", accuracy)

```



Accuracy: 0.9022633744855967

```

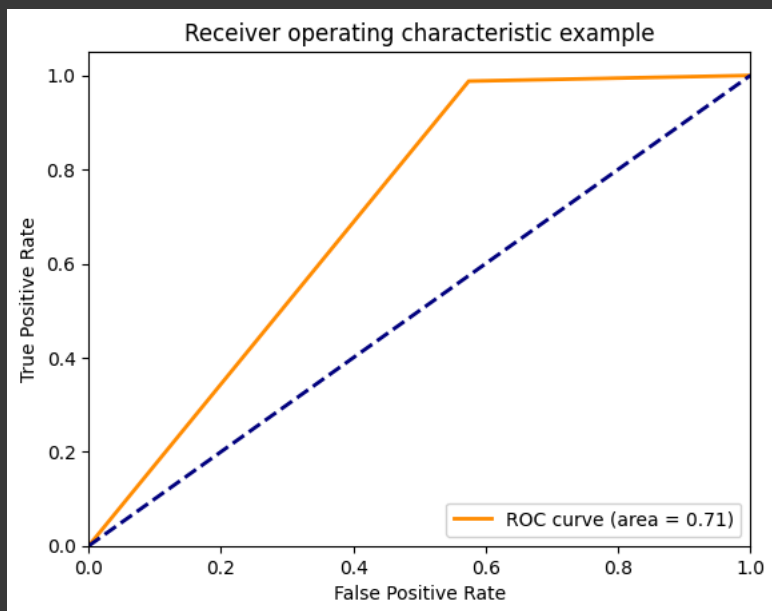
1 # Calculate the AUC and ROC curve
2 fpr, tpr, thresholds = roc_curve(y_test, y_pred)
3 roc_auc = auc(fpr, tpr)

```

```

1 # Plot the ROC curve
2 plt.figure()
3 lw = 2
4 plt.plot(fpr, tpr, color='darkorange',
5          lw=lw, label='ROC curve (area = %0.2f)' % roc_auc)
6 plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
7 plt.xlim([0.0, 1.0])
8 plt.ylim([0.0, 1.05])
9 plt.xlabel('False Positive Rate')
10 plt.ylabel('True Positive Rate')
11 plt.title('Receiver operating characteristic example')
12 plt.legend(loc="lower right")
13 plt.show()

```



```
1 # Print the classification report
2 print(classification_report(y_test, y_pred))
```



	precision	recall	f1-score	support
0	0.86	0.43	0.57	148
1	0.91	0.99	0.94	824
accuracy			0.90	972
macro avg	0.88	0.71	0.76	972
weighted avg	0.90	0.90	0.89	972