

In [21]: `import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns`

In [22]: `df = pd.read_csv("Amazon Sale Report.csv")
df`

Out[22]:

| | index | Order ID | Date | Status | Fulfillment | Sales Channel | ship-service-level | Style | SKU | Category | ... | currency | Amount | ship-city | ship-state | ship-postal-code | ship-country | promotion-ids | B2B | fulfilled-by | Unnamed: 22 |
|-------|-------|-----------------|-----------------|--------------------|-------------|---------------|--------------------|---------|-----------------|---------------|-----|----------|--------|-------------|--------------|------------------|--------------|---|-------|--------------|-------------|
| | 0 | 0 | 8078784-5731545 | Cancelled | Merchant | Amazon.in | Standard | SET389 | SET389-KR-NP-S | Set | ... | INR | 647.62 | MUMBAI | MAHARASHTRA | 400081.0 | IN | NaN | False | Easy Ship | NaN |
| | 1 | 1 | 9198151-1101146 | Delivered to Buyer | Merchant | Amazon.in | Standard | JNE3781 | JNE3781-KR-XXXL | kurta | ... | INR | 406.00 | BENGALURU | KARNATAKA | 560085.0 | IN | Amazon PLCC Free-Financing Universal Merchant ... | False | Easy Ship | NaN |
| | 2 | 2 | 0687676-7273146 | Shipped | Amazon | Amazon.in | Expedited | JNE3371 | JNE3371-KR-XL | kurta | ... | INR | 329.00 | NAVI MUMBAI | MAHARASHTRA | 410210.0 | IN | IN Core Free Shipping 2015/04/08 23-48-5-108 | True | NaN | NaN |
| | 3 | 3 | 9615377-8133951 | Cancelled | Merchant | Amazon.in | Standard | J0341 | J0341-DR-L | Western Dress | ... | INR | 753.33 | PUDUCHERRY | PUDUCHERRY | 605008.0 | IN | NaN | False | Easy Ship | NaN |
| | 4 | 4 | 1069790-7240320 | Shipped | Amazon | Amazon.in | Expedited | JNE3671 | JNE3671-TU-XXXL | Top | ... | INR | 574.00 | CHENNAI | TAMIL NADU | 600073.0 | IN | NaN | False | NaN | NaN |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 20024 | 20024 | 6959123-6990721 | 402-18-22 | Shipped | Amazon | Amazon.in | Expedited | J0215 | J0215-BL-XXL | Blouse | ... | INR | 545.00 | PUNE | MAHARASHTRA | 411057.0 | IN | NaN | False | NaN | NaN |
| 20025 | 20025 | 2519123-7608313 | 402-18-22 | Shipped | Amazon | Amazon.in | Expedited | JNE3670 | JNE3670-TU-L | Top | ... | INR | 399.00 | RAIPUR | CHHATTISGARH | 492013.0 | IN | NaN | False | NaN | NaN |
| 20026 | 20026 | 2461398-7492309 | 407-18-22 | Shipped | Amazon | Amazon.in | Expedited | SET345 | SET345-KR-NP-XL | Set | ... | INR | 626.00 | CHENNAI | TAMIL NADU | 600117.0 | IN | IN Core Free Shipping 2015/04/08 23-48-5-108 | False | NaN | NaN |
| 20027 | 20027 | 4204297-9932357 | 407-18-22 | Shipped | Amazon | Amazon.in | Expedited | SET268 | SET268-KR-NP-XL | Set | ... | INR | 788.00 | CHANDRAPUR | MAHARASHTRA | 442401.0 | IN | IN Core Free Shipping 2015/04/08 23-48-5-108 | False | NaN | NaN |
| 20028 | 20028 | 9633636-2861101 | 404-18-22 | Cancelled | Merchant | Amazon.in | Standard | JNE3721 | JNE3721-KR-KR | kurta | ... | INR | 313.33 | THALASSERY | KERALA | 670103.0 | IN | NaN | False | Easy Ship | NaN |

20029 rows × 24 columns

In [23]: `# checking null value
df.isnull().sum()`

Out[23]:

| | |
|--------------------|-------|
| index | 0 |
| Order ID | 0 |
| Date | 0 |
| Status | 0 |
| Fulfillment | 0 |
| Sales Channel | 0 |
| ship-service-level | 0 |
| Style | 0 |
| SKU | 0 |
| Category | 0 |
| Size | 0 |
| ASIN | 0 |
| Courier Status | 1249 |
| Qty | 0 |
| currency | 1212 |
| Amount | 1212 |
| ship-city | 8 |
| ship-state | 8 |
| ship-postal-code | 8 |
| ship-country | 8 |
| promotion-ids | 7078 |
| B2B | 0 |
| fulfilled-by | 12677 |
| Unnamed: 22 | 20029 |
| dtype: | int64 |

In [24]: `df.dtypes`

Out[24]:

| | |
|--------------------|---------|
| index | int64 |
| Order ID | object |
| Date | object |
| Status | object |
| Fulfillment | object |
| Sales Channel | object |
| ship-service-level | object |
| Style | object |
| SKU | object |
| Category | object |
| Size | object |
| ASIN | object |
| Courier Status | object |
| Qty | int64 |
| currency | object |
| Amount | float64 |
| ship-city | object |
| ship-state | object |
| ship-postal-code | float64 |
| ship-country | object |
| promotion-ids | object |
| B2B | bool |
| fulfilled-by | object |
| Unnamed: 22 | float64 |
| dtype: | object |

In [25]: `df['Status'].value_counts()`

Out[25]:

| | |
|-------------------------------|-------|
| Status | |
| Shipped | 11090 |
| Shipped - Delivered to Buyer | 5678 |
| Cancelled | 2834 |
| Shipped - Returned to Seller | 421 |
| Shipped - Rejected by Buyer | 2 |
| Shipped - Lost in Transit | 2 |
| Shipped - Out for Delivery | 1 |
| Shipped - Returning to Seller | 1 |
| Name: count, dtype: int64 | |

In [40]: `# Drop unwanted columns
df = df.drop(columns=['Order ID','Date'], errors='ignore')
df`

Out[40]:

| | index | Status | Fulfillment | Sales Channel | ship-service-level | Style | SKU | Category | Size | ASIN | ... | currency | Amount | ship-city | ship-state | ship-postal-code | ship-country | promotion-ids | B2B | fulfilled-by | Unnamed: 22 |
|-------|-------|--------|-------------|---------------|--------------------|-------|----------|----------|------|------|-----|----------|--------|-----------|------------|------------------|--------------|---------------|-------|--------------|-------------|
| | 0 | 0 | 0 | 1 | 0 | 1 | 993 3173 | 4 | 7 | 2617 | ... | 0 | 647.62 | 1245 | 23 | 400081.0 | 1 | 0 | False | 1 | 0.0 |
| | 2 | 2 | 1 | 0 | 0 | 0 | 380 1122 | 7 | 8 | 278 | ... | 0 | 329.00 | 1372 | 23 | 410210.0 | 1 | 3 | True | 0 | 0.0 |
| | 3 | 3 | 0 | 1 | 0 | 1 | 272 756 | 6 | 5 | 2184 | ... | 0 | 753.33 | 1582 | 33 | 605008.0 | 1 | 0 | False | 1 | 0.0 |
| | 4 | 4 | 1 | 0 | 0 | 0 | 554 1696 | 5 | 0 | 2017 | ... | 0 | 574.00 | 381 | 41 | 600073.0 | 1 | 0 | False | 0 | 0.0 |
| | 5 | 5 | 1 | 0 | 0 | 0 | 895 2758 | 4 | 8 | 1295 | ... | 0 | 824.00 | 633 | 44 | 201102.0 | 1 | 3 | False | 0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 20024 | 20024 | 1 | 0 | 0 | 0 | 0 | 204 525 | 0 | 10 | 2643 | ... | 0 | 545.00 | 1589 | 23 | 411057.0 | 1 | 0 | False | 0 | 0.0 |
| 20025 | 20025 | 1 | 0 | 0 | 0 | 0 | 553 1690 | 5 | 5 | 1699 | ... | 0 | 399.00 | 1666 | 8 | 492013.0 | 1 | 0 | False | 0 | 0.0 |
| 20026 | 20026 | 1 | 0 | 0 | 0 | 0 | 962 3029 | 4 | 8 | 2639 | ... | 0 | 626.00 | 381 | 41 | 600117.0 | 1 | 3 | False | 0 | 0.0 |
| 20027 | 20027 | 1 | 0 | 0 | 0 | 0 | 899 2774 | 4 | 8 | 1271 | ... | 0 | 788.00 | 369 | 23 | 442401.0 | 1 | 3 | False | 0 | 0.0 |
| 20028 | 20028 | 0 | 1 | 0 | 0 | 1 | 591 1834 | 7 | 8 | 2079 | ... | 0 | 313.33 | 1925 | 20 | 670103.0 | 1 | 0 | False | 1 | 0.0 |

12714 rows × 22 columns

In [41]: `# Remove missing rows for important columns
df = df.dropna(subset=['Status','Amount'])
df`

Out[41]:

| | index | Status | Fulfillment | Sales Channel | ship-service-level | Style | SKU | Category | Size | ASIN | ... | currency | Amount | ship-city | ship-state | ship-postal-code | ship-country | promotion-ids | B2B | fulfilled-by | Unnamed: 22 |
|--|-------|--------|-------------|---------------|--------------------|-------|----------|----------|------|------|-----|----------|--------|-----------|------------|------------------|--------------|---------------|-------|--------------|-------------|
| | 0 | 0 | 0 | 1 | 0 | 1 | 993 3173 | 4 | 7 | 2617 | ... | 0 | 647.62 | 1245 | 23 | 400081.0 | 1 | 0 | False | 1 | 0.0 |
| | 2 | 2 | 1 | 0 | 0 | 0 | 380 1122 | 7 | 8 | 278 | ... | 0 | 329.00 | 1372 | 23 | 410210.0 | 1 | 3 | True | 0 | 0.0 |
| | 3 | 3 | 0 | 1 | 0 | 1 | 272 756 | 6 | 5 | 2184 | ... | 0 | 753.33 | 1582 | 33 | 605008.0 | 1 | 0 | False | 1 | 0.0 |
| | 4 | 4 | 1 | 0 | 0 | 0 | 554 1696 | 5 | 0 | 2017 | ... | 0 | 574.00 | 381 | 41 | 600073.0 | 1 | 0 | False | 0 | 0.0 |
| | 5 | 5 | 1 | 0 | 0 | 0 | 895 2758 | 4 | 8 | 1295 | ... | 0 | 824.00 | 633 | 44 | 201102.0 | 1 | 3 | False | 0 | 0.0 |

5 rows × 22 columns

In [28]: `# Keep only Shipped and Cancelled
df = df[df['Status'].isin(['Shipped','Cancelled'])]`

In [42]: `# convert status to 1 and 0
df['Status']=df['Status'].apply(lambda x: 1 if x=="Shipped" else 0)
df.head()`

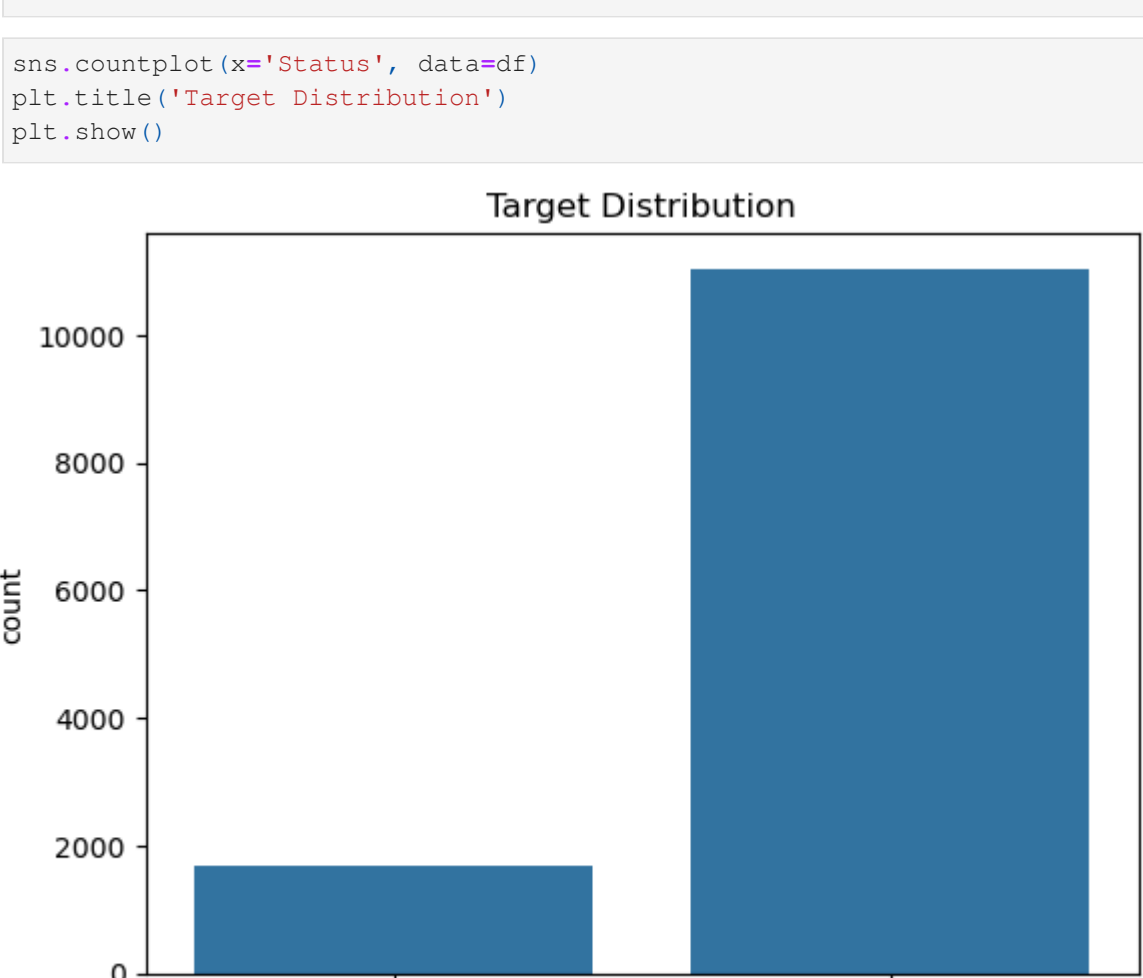
Out[42]:

| | index | Status | Fulfillment | Sales Channel | ship-service-level | Style | SKU | Category | Size | ASIN | ... | currency | Amount | ship-city | ship-state | ship-postal-code | ship-country | promotion-ids | B2B | fulfilled-by | Unnamed: 22 |
|--|-------|--------|-------------|---------------|--------------------|-------|----------|----------|------|------|-----|----------|--------|-----------|------------|------------------|--------------|---------------|-------|--------------|-------------|
| | 0 | 0 | 0 | 1 | 0 | 1 | 993 3173 | 4 | 7 | 2617 | ... | 0 | 647.62 | 1245 | 23 | 400081.0 | 1 | 0 | False | 1 | 0.0 |
| | 2 | 2 | 0 | 0 | 0 | 0 | 380 1122 | 7 | 8 | 278 | ... | 0 | 329.00 | 1372 | 23 | 410210.0 | 1 | 3 | True | 0 | 0.0 |
| | 3 | 3 | 0 | 1 | 0 | 1 | 272 756 | 6 | 5 | 2184 | ... | 0 | 753.33 | 1582 | 33 | 605008.0 | 1 | 0 | False | 1 | 0.0 |
| | 4 | 4 | 0 | 0 | 0 | 0 | 554 1696 | 5 | 0 | 2017 | ... | 0 | 574.00 | 381 | 41 | 600073.0 | 1 | 0 | False | 0 | 0.0 |
| | 5 | 5 | 0 | 0 | 0 | 0 | 895 2758 | 4 | 8 | 1295 | ... | 0 | 824.00 | 633 | 44 | 201102.0 | 1 | 3 | False | 0 | 0.0 |

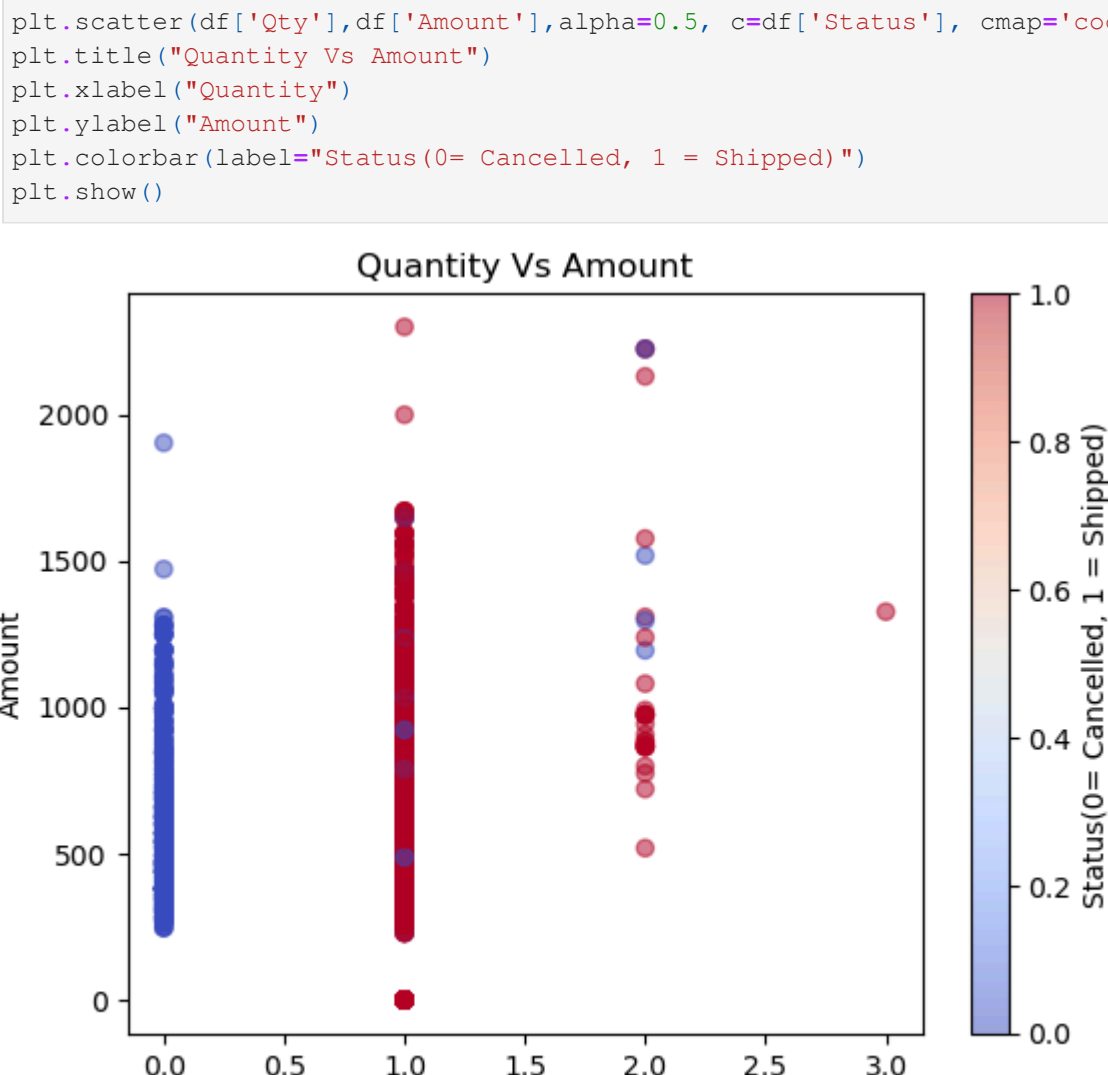
5 rows × 22 columns

In [30]: `# fill any other missing values
df = df.fillna(0)`

In [31]: `sns.countplot(x='Status', data=df)
plt.title('Target Distribution')
plt.show()`



In [33]: `plt.scatter(df['Qty'],df['Amount'],alpha=0.5, c=df['Status'], cmap='coolwarm')
plt.title("Quantity Vs Amount")
plt.xlabel("Quantity")
plt.ylabel("Amount")
plt.colorbar(label="Status(0= Cancelled, 1 = Shipped)")
plt.show()`



In [35]: `# Encode categorical Variable
from sklearn.preprocessing import LabelEncoder
cat_cols = df.select_dtypes(include='object').columns
le = LabelEncoder()
for col in cat_cols:
 df[col] = df[col].astype(str)
 df[col] = le.fit_transform(df[col])`

In [37]: `# separate the input
X = df.drop('Status', axis=1)
Y = df['Status']`

In [39]: `# Train test the model
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2 , random_state = 0)
X_train.shape , Y_train.shape`

Out[39]: ((10171, 21), (10171,))

In [52]: `# Model tranning
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(random_state = 30)
model.fit(X_train,Y_train)`

Out[52]:

```
RandomForestClassifier
```

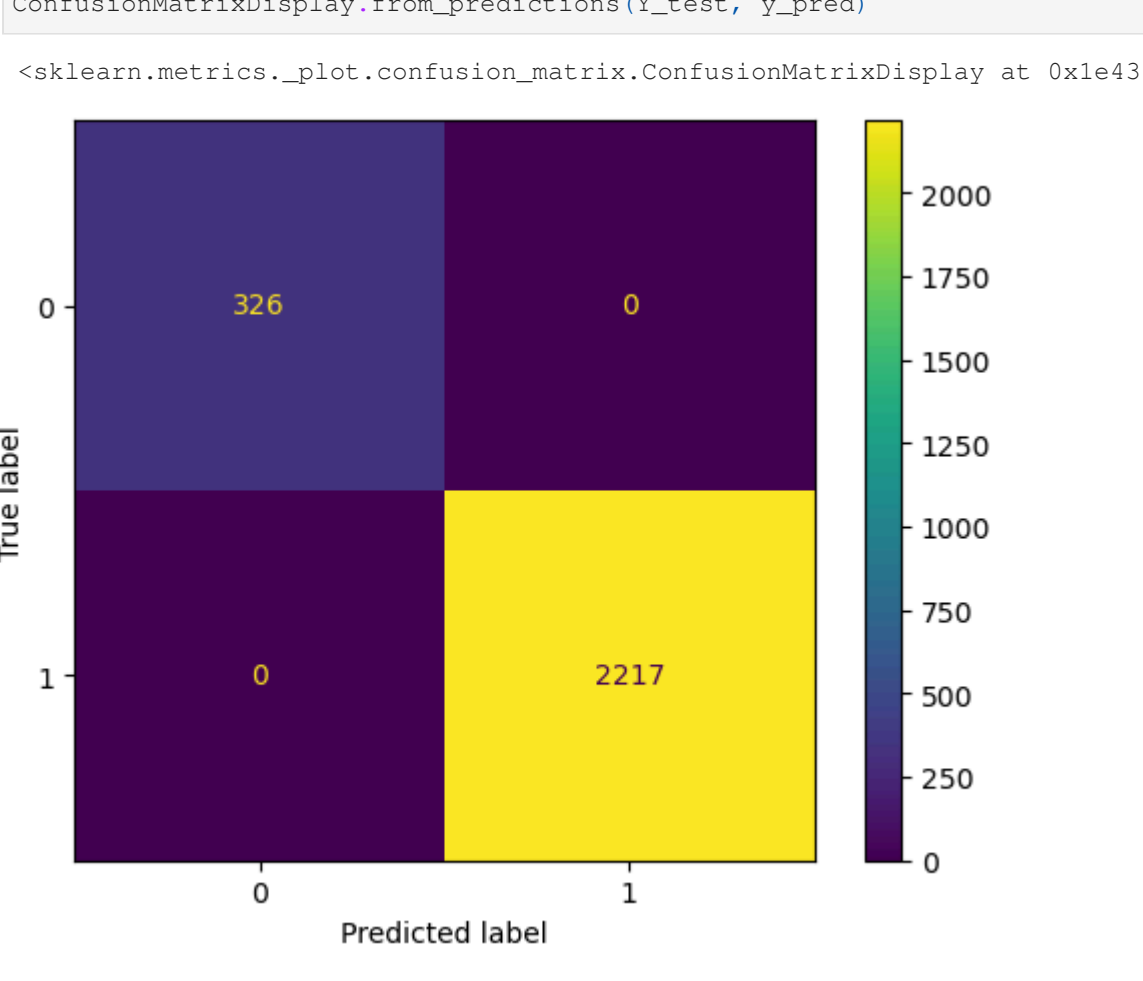
In [53]: `# prediction
y_pred = model.predict(X_test)
y_pred`

Out[53]: array([1, 1, 1, ..., 1, 1, 1], dtype=int64)

In [54]: `from sklearn.metrics import ConfusionMatrixDisplay`

In [55]: `ConfusionMatrixDisplay.from_predictions(Y_test, y_pred)`

Out[55]: <sklearn.metrics_plot.confusion_matrix.ConfusionMatrixDisplay at 0x1e43f21a720>



In [56]: `from sklearn.metrics import accuracy_score,precision_score,recall_score`

In [57]: `# Calculate metrics
accuracy = accuracy_score(Y_test, y_pred)
accuracy`

Out[57]: 1.0

In [58]: `from sklearn.metrics import classification_report
print(classification_report(Y_test,y_pred))`

```
              precision    recall  f1-score   support

     0               1.00        1.00        1.00         326
     1               1.00        1.00        1.00        2217

 accuracy               1.00
 macro avg              1.00
 weighted avg           1.00
```

In []: