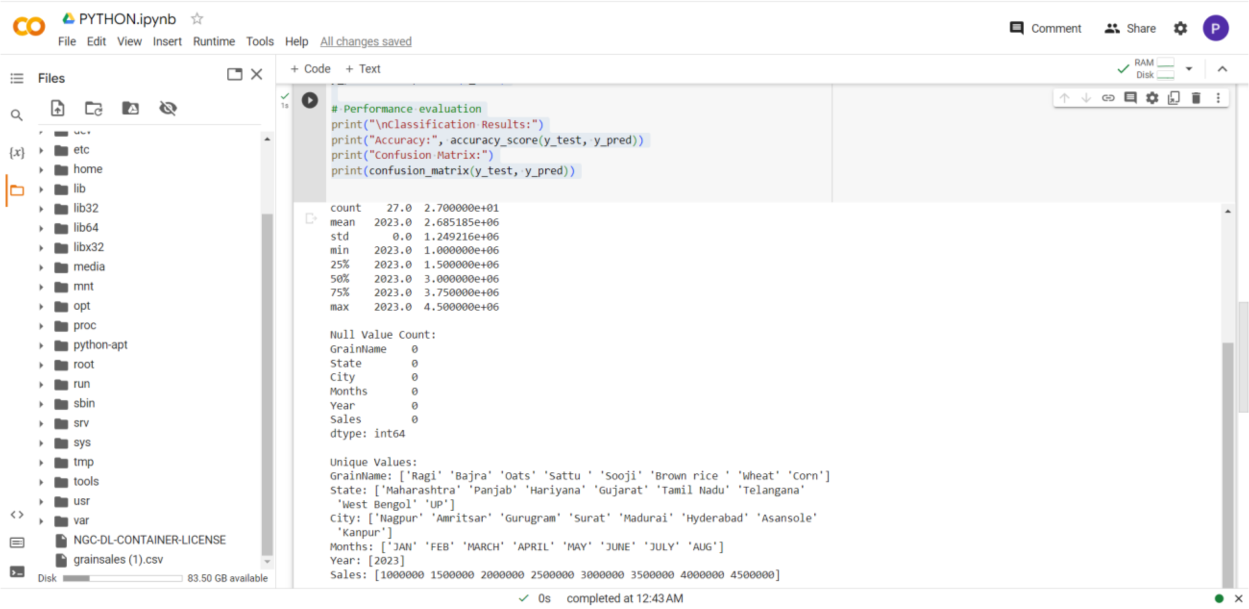
```
                 GROUP PROJECT (MINOR )
                PRATHAMESH SANJAY GALUGADE (755)
                HEMANT MADHAV MANKAR (744)
                PIYUSH RAJENDRA PATIL(752)




import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix

# Load the dataset
data = pd.read_csv("/grainsales (1).csv")

# Data computation operations
print("Dataset Information:")
print(data.info())

print("\nSummary Statistics:")
print(data.describe())

print("\nNull Value Count:")
print(data.isnull().sum())

print("\nUnique Values:")
for column in data.columns:
    unique_values = data[column].unique()
    print(column + ":", unique_values)

# Data manipulation operations
# Remove any duplicate rows
data = data.drop_duplicates()

# Encode categorical variables
label_encoder = LabelEncoder()
data['GrainName'] = label_encoder.fit_transform(data['GrainName'])
data['State'] = label_encoder.fit_transform(data['State'])
data['City'] = label_encoder.fit_transform(data['City'])
data['Months'] = label_encoder.fit_transform(data['Months'])
```

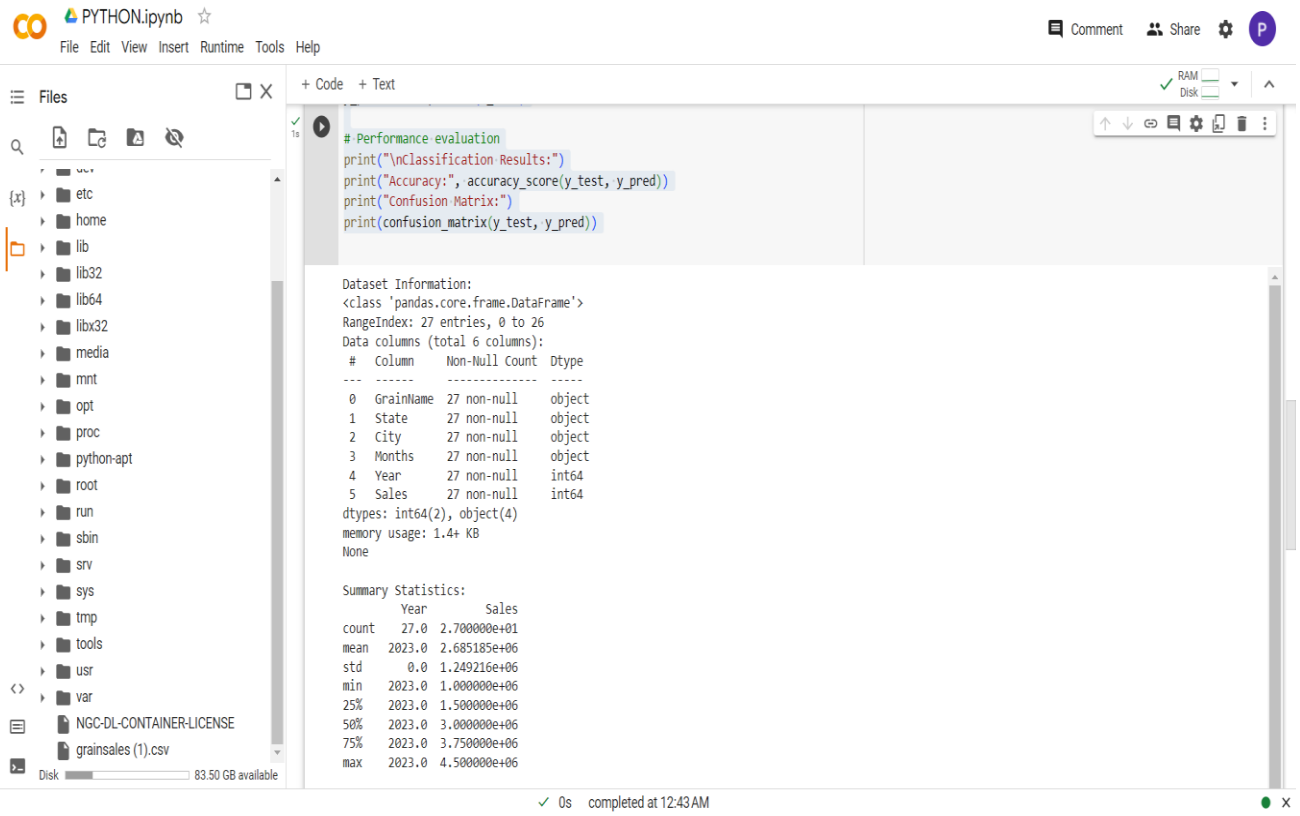
```python
# Data visualization operations
plt.figure(figsize=(10, 6))
sns.boxplot(x='GrainName', y='Sales', data=data)
plt.xlabel('Grain Name')
plt.ylabel('Sales')
plt.title('Sales by Grain Name')
plt.show()

plt.figure(figsize=(10, 6))
sns.countplot(x='State', data=data)
plt.xlabel('State')
plt.ylabel('Count')
plt.title('Distribution of States')
plt.show()

# Split the dataset into features (X) and labels (y)
X = data.drop('Sales', axis=1)
y = data['Sales']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Classification using KNN
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)

# Performance evaluation
print("\nClassification Results:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```



```
# Performance evaluation
print("\nClassification Results:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
count    27.0  2.700000e+01
mean   2023.0  2.685185e+06
std       0.0  1.249216e+06
min    2023.0  1.000000e+06
25%    2023.0  1.500000e+06
50%    2023.0  3.000000e+06
75%    2023.0  3.750000e+06
max    2023.0  4.500000e+06

Null Value Count:
GrainName    0
State        0
City         0
Months       0
Year         0
Sales        0
dtype: int64

Unique Values:
GrainName: ['Ragi' 'Bajra' 'Oats' 'Sattu ' 'Sooji' 'Brown rice ' 'Wheat' 'Corn']
State: ['Maharashtra' 'Panjab' 'Hariyana' 'Gujarat' 'Tamil Nadu' 'Telangana'
 'West Bengol' 'UP']
City: ['Nagpur' 'Amritsar' 'Gurugram' 'Surat' 'Madurai' 'Hyderabad' 'Asansole'
 'Kanpur']
Months: ['JAN' 'FEB' 'MARCH' 'APRIL' 'MAY' 'JUNE' 'JULY' 'AUG']
Year: [2023]
Sales: [1000000 1500000 2000000 2500000 3000000 3500000 4000000 4500000]
```

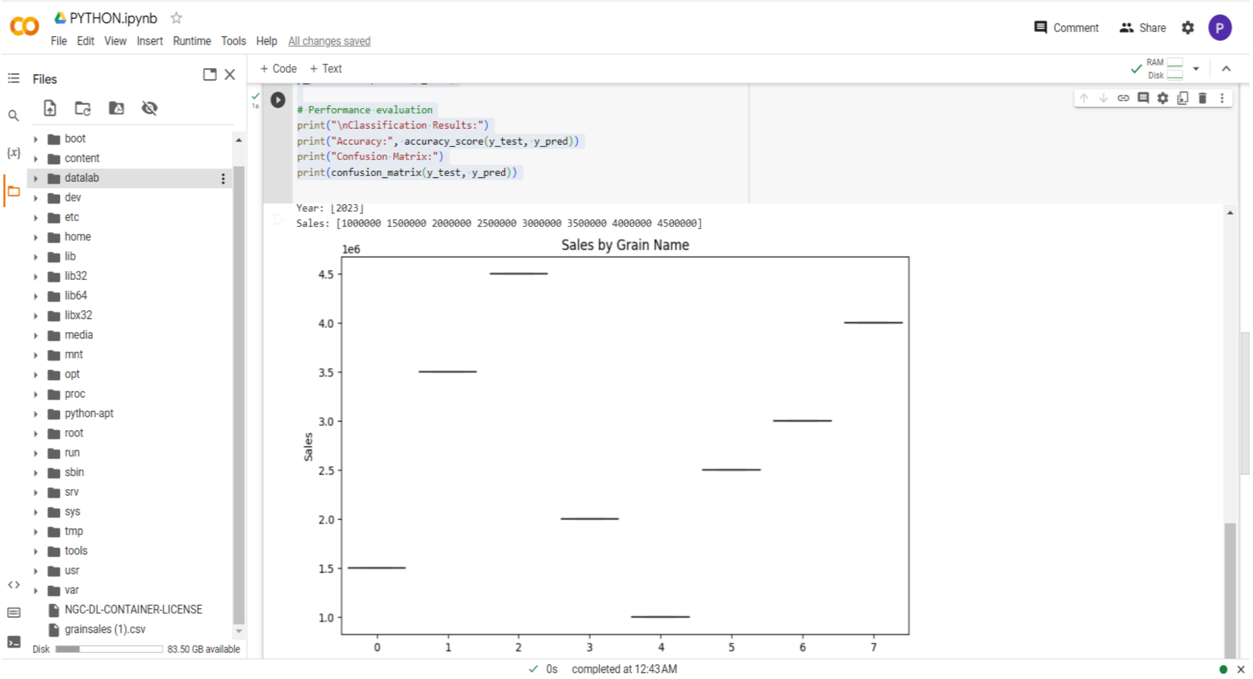
✓ 0s completed at 12:43 AM



```
# Performance evaluation
print("\nClassification Results:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

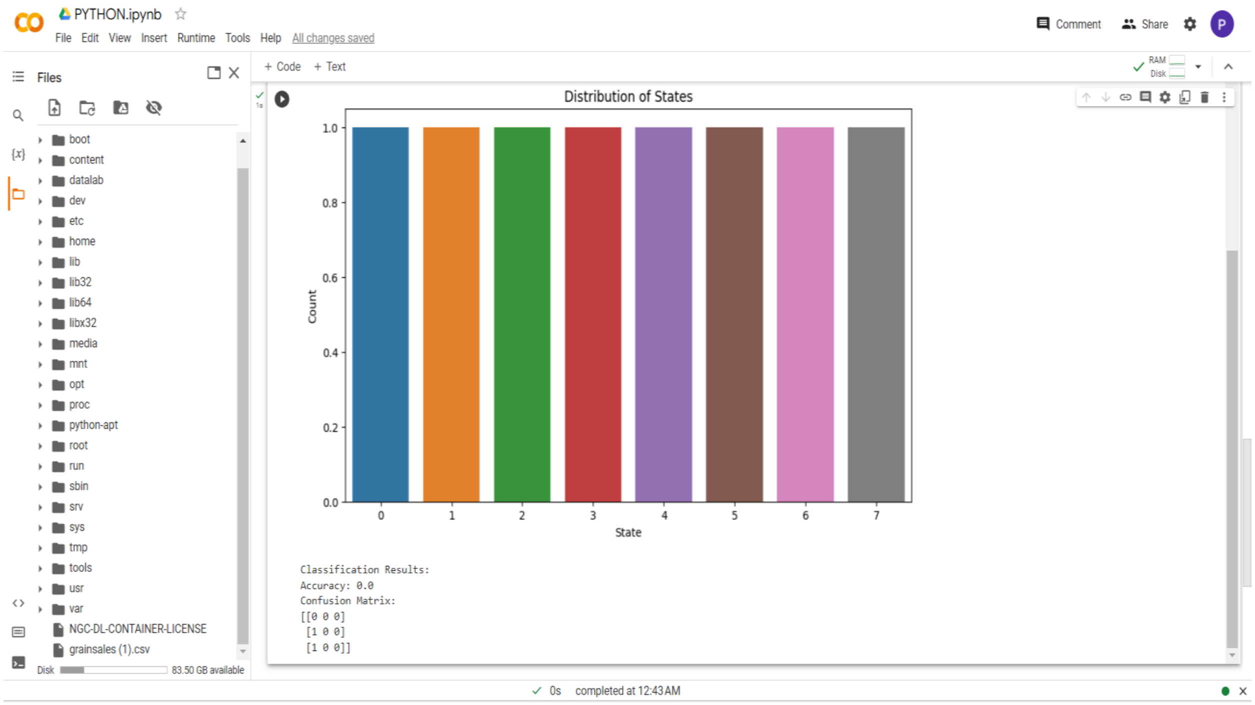
```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27 entries, 0 to 26
Data columns (total 6 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   GrainName  27 non-null     object
 1   State      27 non-null     object
 2   City       27 non-null     object
 3   Months     27 non-null     object
 4   Year       27 non-null     int64
 5   Sales      27 non-null     int64
dtypes: int64(2), object(4)
memory usage: 1.4+ KB
None

Summary Statistics:
         Year         Sales
count    27.0  2.700000e+01
mean   2023.0  2.685185e+06
std       0.0  1.249216e+06
min    2023.0  1.000000e+06
25%    2023.0  1.500000e+06
50%    2023.0  3.000000e+06
75%    2023.0  3.750000e+06
max    2023.0  4.500000e+06
```

✓ 0s completed at 12:43 AM



```
# Performance evaluation
print("\nClassification Results:")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
Year: [2023]
Sales: [1000000 1500000 2000000 2500000 3000000 3500000 4000000 4500000]
```



```
Classification Results:
Accuracy: 0.0
Confusion Matrix:
[[0 0 0]
 [1 0 0]
 [1 0 0]]
```