The script performs several key analyses:

Data Loading and Cleaning:

- Loads Amazon sales data from a CSV file
- Removes irrelevant columns and duplicates
- Handles missing values
- Converts dates to proper datetime format

Sales Analysis:

- Analyzes sales by product category
- Examines sales distribution across states
- Tracks monthly sales trends
- The top selling category is "Set" with ₹39.2M in sales
- Maharashtra leads in state-wise sales with ₹13.3M

Visualizations:

- Creates line plots for monthly sales trends
- Shows bar charts for top 10 categories
- Displays clustering results of sales patterns

Sales Forecasting:

- Uses ARIMA (AutoRegressive Integrated Moving Average) model
- Predicts sales for the next 5 months
- Forecast shows sales expected to stay around ₹24-25M per month

Geographic Analysis:

- Top 5 states by sales:
  - Maharashtra: ₹13.3M
  - Karnataka: ₹10.4M
  - Telangana: ₹6.9M
  - Uttar Pradesh: ₹6.8M
  - Tamil Nadu: ₹6.5M

Product Categories:

- Leading categories:
  - Set: ₹39.2M
  - Kurta: ₹21.3M
  - Western Dress: ₹11.2M
  - Top: ₹5.3M
  - Ethnic Dress: ₹791K

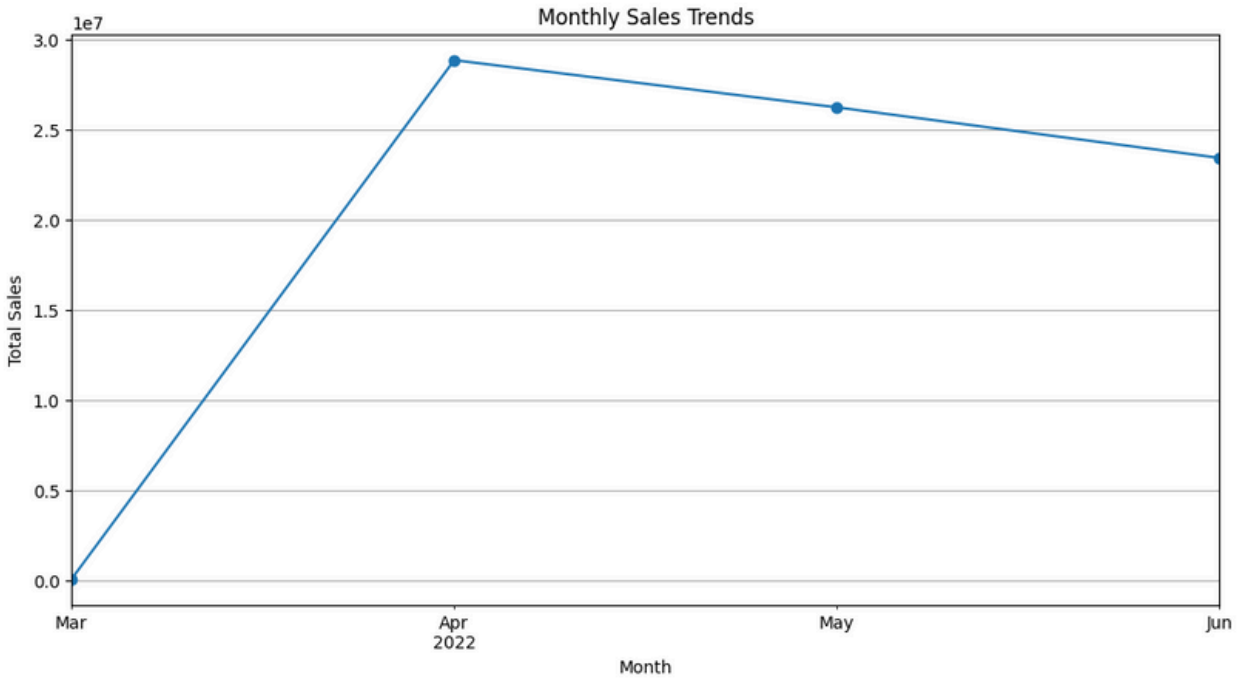
```python
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from statsmodels.tsa.arima.model import ARIMA

# Load the dataset
file_path = 'Desktop/Amazon Sale Report.csv'
# Error handling for file loading
try:
    sales_data = pd.read_csv(file_path,low_memory=False)
    print (sales_data.head())
except FileNotFoundError:
    print("File not found. Please check the file path.")
    raise
```

```
   index                Order ID        Date                                 Status
\
0      0 405-8078784-5731545  4/30/2022                              Cancelled
1      1 171-9198151-1101146  4/30/2022  Shipped - Delivered to Buyer
2      2 404-0687676-7273146  4/30/2022                              Shipped
3      3 403-9615377-8133951  4/30/2022                              Cancelled
4      4 407-1069790-7240320  4/30/2022                              Shipped


  Fulfilment Sales Channel ship-service-level          Style
SKU  \
0    Merchant       Amazon.in             Standard    SET389  SET389-KR-
NP-S
1    Merchant       Amazon.in             Standard  JNE3781  JNE3781-KR-
XXXL
                    Amazon.in            Expedited  JNE3371
2      Amazon       Amazon.in             Standard     J0341  JNE3371-KR-
XL
3    Merchant       Amazon.in            Expedited  JNE3671        J0341-
DR-L
4      Amazon                                                 JNE3671-TU-
XXXL

            Category ... currency  Amount      ship-city      ship-state  \
0                Set ...      INR  647.62         MUMBAI    MAHARASHTRA
1              kurta ...      INR  406.00      BENGALURU      KARNATAKA
2              kurta ...      INR  329.00  NAVI MUMBAI    MAHARASHTRA
```

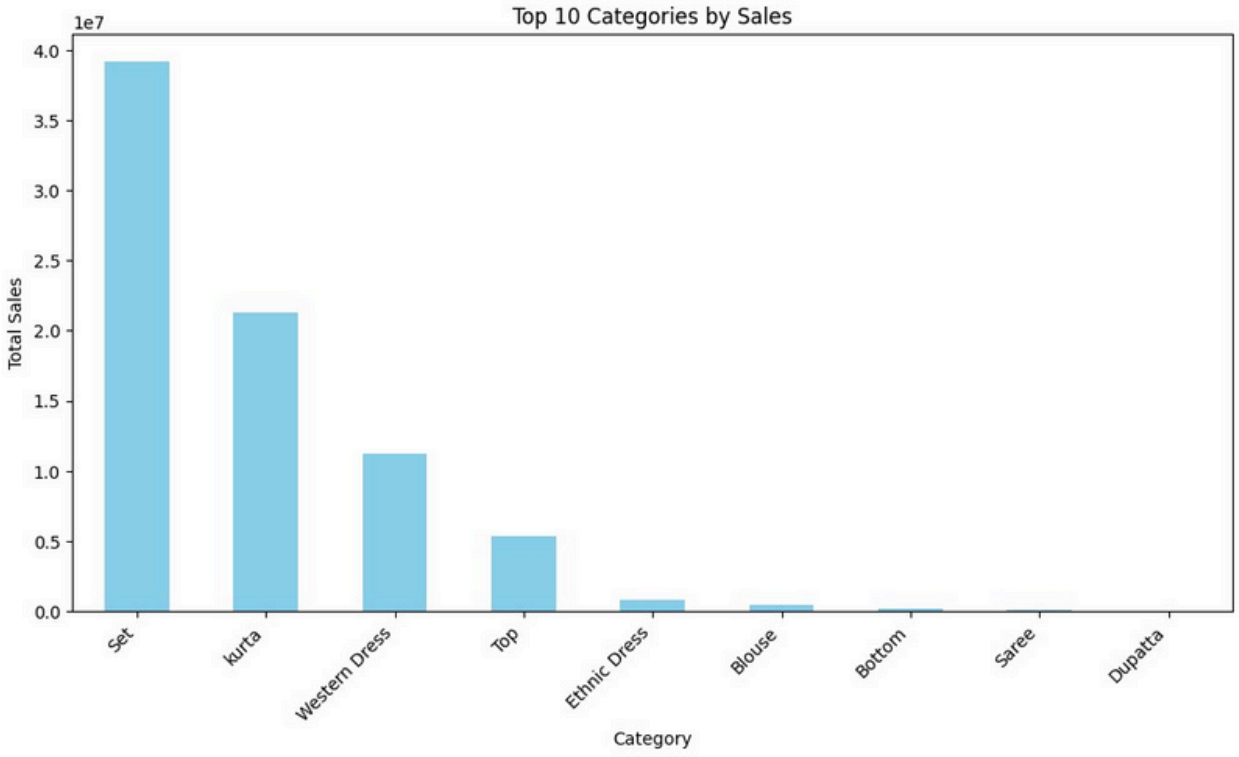
```
3        Western Dress ...       INR  753.33     PUDUCHERRY PUDUCHER
4                     Top ...    INR  574.00        CHENNAI    RY    TAMIL
                                                                    NADU

      ship-postal-code ship-country \
0             400081.0                IN
1             560085.0                IN
2             410210.0                IN
3             605008.0                IN
4             600073.0                IN

                                                promotion-ids    B2B
fulfilled-by \
0                                                         NaN  False      Easy
Ship
1    Amazon PLCC Free-Financing Universal Merchant ...        False      Easy
Ship
2                IN Core Free Shipping 2015/04/08 23-48-5-108     True
NaN
3                                                         NaN  False      Easy
Ship
4                                                         NaN
NaN

   Unnamed: 22
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN

[5 rows x 24 columns]
```

```
# Drop irrelevant columns
sales_data_cleaned = sales_data.drop(columns=['index', 'Unnamed: 22'],
errors='ignore')

# Handle missing values
sales_data_cleaned['Amount'] = sales_data_cleaned['Amount'].fillna(0)
categorical_columns = ['Courier Status', 'currency', 'fulfilled-by',
'promotion-ids']
for col in categorical_columns:
    sales_data_cleaned[col] =
sales_data_cleaned[col].fillna('Unknown')

# Convert 'Date' column to datetime format with error handling
try:
    sales_data_cleaned['Date'] =
pd.to_datetime(sales_data_cleaned['Date'], errors='coerce')
except Exception as e:
```

```
        print(f"Error in datetime conversion: {e}")
        raise

# Remove duplicates and handle negative/zero values in 'Amount'
sales_data_cleaned = sales_data_cleaned.drop_duplicates()
sales_data_cleaned = sales_data_cleaned[sales_data_cleaned['Amount'] >
0]
# Exploratory Data Analysis (EDA)
# Total sales by category
category_sales = sales_data_cleaned.groupby('Category')
['Amount'].sum().sort_values(ascending=False)
# Total sales by state
state_sales = sales_data_cleaned.groupby('ship-state')
['Amount'].sum().sort_values(ascending=False)
# Monthly sales trends
sales_data_cleaned['Month'] =
sales_data_cleaned['Date'].dt.to_period('M')
monthly_sales = sales_data_cleaned.groupby('Month')['Amount'].sum()

# Plot: Sales trends over time
plt.figure(figsize=(12, 6))
monthly_sales.plot(kind='line', marker='o')
plt.title('Monthly Sales Trends')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.grid()
plt.show()
```

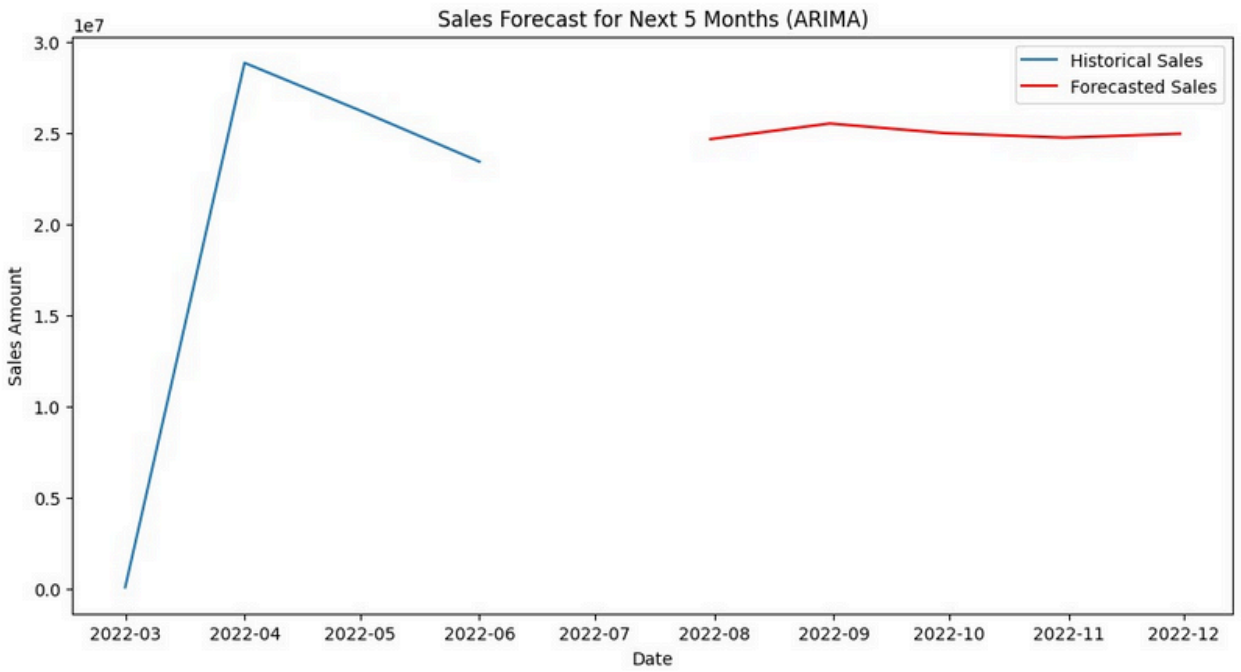


```
# Plot: Sales by category (Top 10)
plt.figure(figsize=(12, 6))
category_sales.head(10).plot(kind='bar', color='skyblue')
plt.title('Top 10 Categories by Sales')
plt.xlabel('Category')
plt.ylabel('Total Sales')
plt.xticks(rotation=45, ha='right')
plt.show()
```

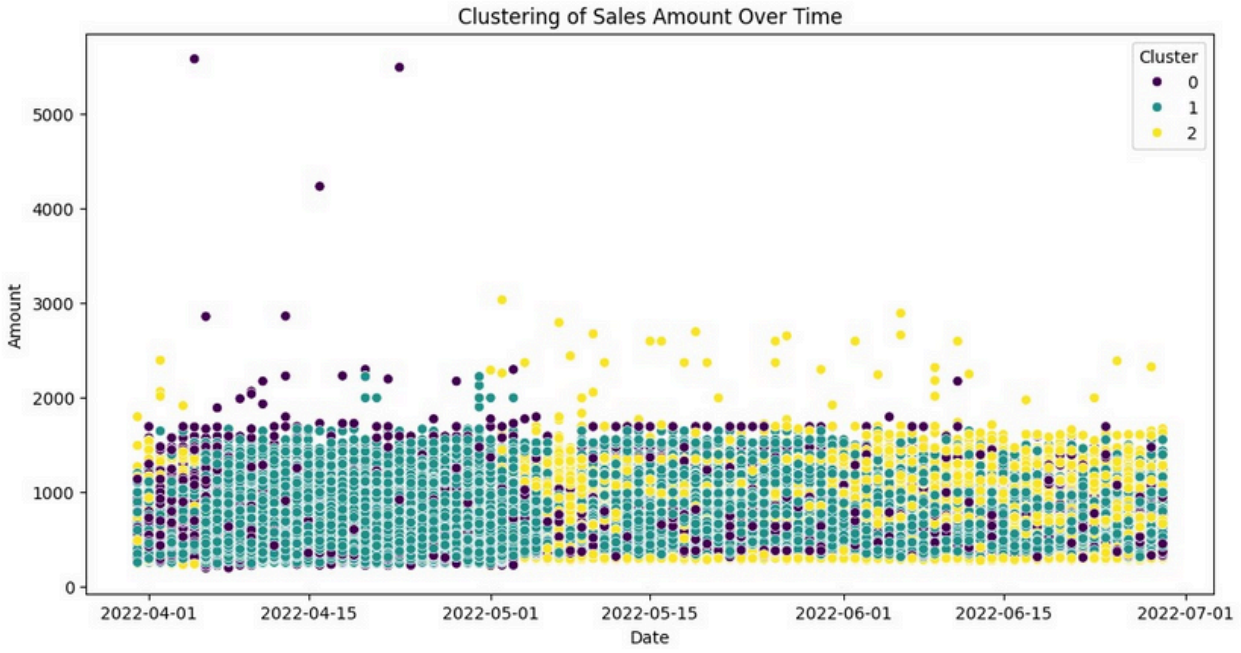



```
# Key Metrics
sales_data_cleaned['Cost Allocation'] = (sales_data_cleaned['Amount']
/ sales_data_cleaned['Amount'].sum()) * 100

# Prepare data for ARIMA
df = monthly_sales.reset_index()
df.columns = ['ds', 'y']
df['ds'] = df['ds'].dt.to_timestamp()
# Fit ARIMA model
model = ARIMA(df['y'], order=(2, 1, 0))  # You can adjust (p, d, q) as
per the data
model_fit = model.fit()
# Make predictions
forecast_steps = 5 # Number of steps to forecast
forecast = model_fit.forecast(steps=forecast_steps)
```

```
# Plot forecast
plt.figure(figsize=(12, 6))
plt.plot(df['ds'], df['y'], label='Historical Sales')
forecast_index = pd.date_range(start=df['ds'].iloc[-1],
periods=forecast_steps + 1, freq='ME')[1:]
plt.plot(forecast_index, forecast, color='red', label='Forecasted
Sales')
plt.title('Sales Forecast for Next 5 Months (ARIMA)')
plt.xlabel('Date')
plt.ylabel('Sales Amount')
plt.legend()
plt.show()
```




```
# Clustering for Optimization # Cluster based on multiple features
(e.g., Amount and Category) kmeans = KMeans(n_clusters=3,
random_state=42) sales_data_cleaned['Cluster'] =
kmeans.fit_predict(sales_data_cleaned[['Amount',
'Category']].apply(lambda x: pd.factorize(x)[0]))

# Plot: Clustering Results
plt.figure(figsize=(12, 6))
sns.scatterplot(data=sales_data_cleaned, x='Date', y='Amount',
hue='Cluster', palette='viridis')
plt.title('Clustering of Sales Amount Over Time')
plt.show()
```




```
# Summary of Insights print("Top Categories by Sales:\n", category_sales.head(10)) print("Top States by Sales:\n", state_sales.head(10)) print("Predicted Future Sales:\n", forecast)

# Save processed data
sales_data_cleaned.to_csv('Cleaned_Amazon_Sales_Report.csv', index=False)
```

```
Top Categories by Sales:
 Category
Set                39202022.03
kurta              21299013.70
Western Dress      11216072.69
Top                 5347792.30
Ethnic Dress         791217.66
Blouse               458408.18
Bottom               150667.98
Saree                123933.76
Dupatta                 915.00
Name: Amount, dtype: float64
Top States by Sales:
 ship-state
MAHARASHTRA        13334595.14
KARNATAKA          10481114.37
TELANGANA           6916615.65
UTTAR PRADESH       6816109.08
TAMIL NADU          6515650.11
DELHI               4235215.97
```

```
KERALA             3830227.58
WEST BENGAL.       3507880.44
ANDHRA PRADESH     3219831.72
HARYANA            2882092.99
Name: Amount, dtype: float64
Predicted Future Sales:
4    2.465799e+07
5    2.551434e+07
6    2.498345e+07
7.   2.473846e+07
8   2.495340e+07
Name: predicted_mean, dtype:
float64
```