Tech Saksham

Capstone Project Report

“E-Commerce Analysis”

“University College Of Engineering - Arni”

NM ID	NAME
aut513321105701	LOKESH G

RAMAR BOSE

Sr. AI Master Trainer

ABSTRACT

The program performs a comprehensive analysis of an e-commerce dataset to provide insights into business performance and customer preferences. It begins with data preprocessing to handle missing values, encode categorical data, and remove duplicate entries. Exploratory data analysis (EDA) includes visualizations such as histograms, boxplots, scatter plots, and heatmaps to understand relationships within the data.

Next, the code conducts feature selection and builds predictive models using machine learning algorithms such as Linear Regression (LR), K-Nearest Neighbors (KNN), Decision Tree, and Random Forest. These models are trained to predict profit based on features like sales, discount, and shipping cost.

The models' performances are evaluated and compared using metrics like mean squared error (MSE) and R-squared (R2). The analysis provides data-driven insights that help identify the most suitable model for profit prediction, aiding the business in making informed decisions for strategic planning.

1. Problem statement
2. Data collection
3. Existing solution
4. Proposed solution with used models
5. Result

INDEX

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

The dataset consists of sales transactions from various categories of products and includes information such as sales revenue, profit, quantity sold, shipping cost, and other factors. The main focus is to explore how different factors such as sales, discount, and shipping cost impact profit. Additionally, the analysis aims to identify trends and patterns in the data, such as most sold products, monthly variations, and category-wise performance.

1.2 Proposed Solution

The goal of this analysis is to understand and predict profitability in an E-commerce dataset.

Data Cleaning and Preprocessing:

- Remove missing values and duplicates from the dataset.
- Convert date columns to appropriate data types and extract relevant information such as month and day of the week.
- Normalize numerical columns and handle outliers for better model performance.

Exploratory Data Analysis:

- Create data visualizations such as histograms, boxplots, scatter plots, and heatmaps to explore relationships between different features.
- Identify top-selling products and categories, as well as category-wise and monthly variations in sales and quantity.

Feature Selection:

- Select relevant features such as sales, discount, and shipping cost for model training to predict profit.

Model Building and Evaluation:

- Split the data into training and testing sets.
- Train different regression models (e.g., linear regression, K-Nearest Neighbors, decision trees, random forest) on the training data.
- Evaluate model performance using metrics such as mean squared error (MSE) and R-squared (R2).
- Compare the performance of different models to identify the best model for predicting profit.

Insights and Recommendations:

- Based on the analysis, derive insights into the factors that have the most significant impact on profit.
- Provide recommendations for the E-commerce business on how to optimize their operations to improve profitability.

This proposed solution aims to provide a comprehensive analysis of the E-commerce dataset and use the insights gained to create predictive models for profitability. These models can help guide strategic decision-making for the business.

1.3 Feature

- **Data Cleaning:** Handles missing values and duplicates, and performs normalization and outlier detection to clean and prepare the data.
- **Data Exploration and Visualization:** Utilizes histograms, boxplots, heatmaps, pie charts, and scatter plots to provide insights into the data's distribution and relationships.
- **Correlation Analysis:** Calculates the correlation matrix and visualizes it to understand relationships between features.
- **Model Building and Evaluation:** Implements multiple regression models (Linear Regression, K-Nearest Neighbors, Decision Tree, and Random Forest) to predict the target variable (Profit).
- **Model Performance Analysis:** Evaluates models based on metrics such as Mean Squared Error and R-squared, and plots model performance for comparison.

1.4 Advantages

- **Comprehensive Analysis:** Provides a complete workflow from data cleaning to model evaluation, allowing for thorough analysis and insights.
- **Multiple Modeling Techniques:** Offers a comparison of multiple regression models, enabling the selection of the most suitable model for the task.
- **Interactive Visualizations:** Uses Plotly and Seaborn for visualizations, allowing for interactive and informative visual exploration of data.
- **Ease of Understanding:** The code includes comments and clear variable names to guide users through each step.

1.5 Scope

- **Wide Application:** This script can be applied to various datasets beyond e-commerce, making it useful for general data analysis and modeling tasks.
- **Adaptability:** The script can be customized for different target variables, features, or models as needed.
- **Scalability:** By adjusting data types and data handling methods, the script can be scaled up for larger datasets.
- **Extensibility:** New features or models can be added to the script as required, providing flexibility for future analyses.

CHAPTER 2

SERVICES AND TOOLS REQUIRED

2.1 LR - Exiting Models

- **Models:** The existing models include simple and multiple linear regression, with potential options for more advanced linear models such as Ridge and Lasso regression.
- **Preprocessing:** Techniques like handling missing values, data normalization, and one-hot encoding categorical variables.
- **Model Training and Testing:** Splitting data into training and testing sets, fitting the model, and evaluating its performance.

2.1.1 Required – System config | Cloud computing

System Configurations:

- A system with at least 8 GB of RAM, though 16 GB or more is recommended for working with larger datasets.
- Multi-core processor for faster computations and parallel processing.
- Storage space depending on the size of the datasets.

Cloud Computing:

- Cloud platforms such as AWS, Google Cloud, and Azure provide scalable infrastructure to run computations on large datasets.
- Services like Jupyter notebooks and Google Colab can be used for coding and executing scripts.
- Managed machine learning services can help with deploying models and automating tasks.

2.1.2 Services Used

- **Data Storage and Management:** Using cloud-based data storage solutions such as AWS S3 or Google Cloud Storage for data access and management.
- **Machine Learning Platforms:** Utilizing machine learning services such as AWS SageMaker or Google Cloud AI for model training and deployment.
- **Data Visualization and Exploration:** Tools like Plotly and Seaborn for data visualization and exploratory data analysis.

2.2 Tools and Software used

Tools:

- Python for scripting and model development.
- Jupyter Notebook or Google Colab for interactive development and experimentation.
- Version control tools like Git for collaboration and tracking changes.
- CI/CD tools for automated testing and deployment pipelines.

Software Requirements:

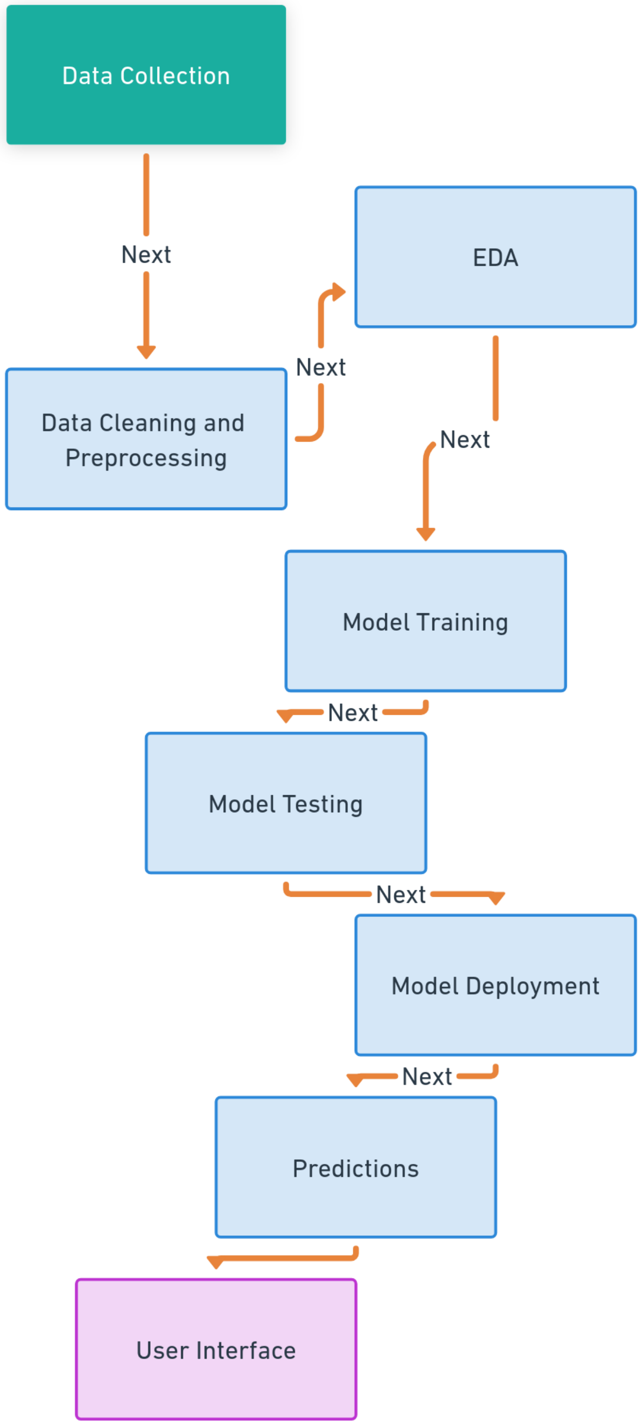
- Python (with libraries such as NumPy, pandas, scikit-learn, TensorFlow, PyTorch).
- Docker Engine for containerization.
- Cloud SDKs for interacting with cloud services.
- Text editors or integrated development environments (IDEs) for coding (e.g., VSCode, PyCharm).
- Web servers (e.g., Flask, FastAPI) for deploying models as APIs.
- Database systems for storing and managing data (e.g., PostgreSQL, MongoDB)

CHAPTER 3

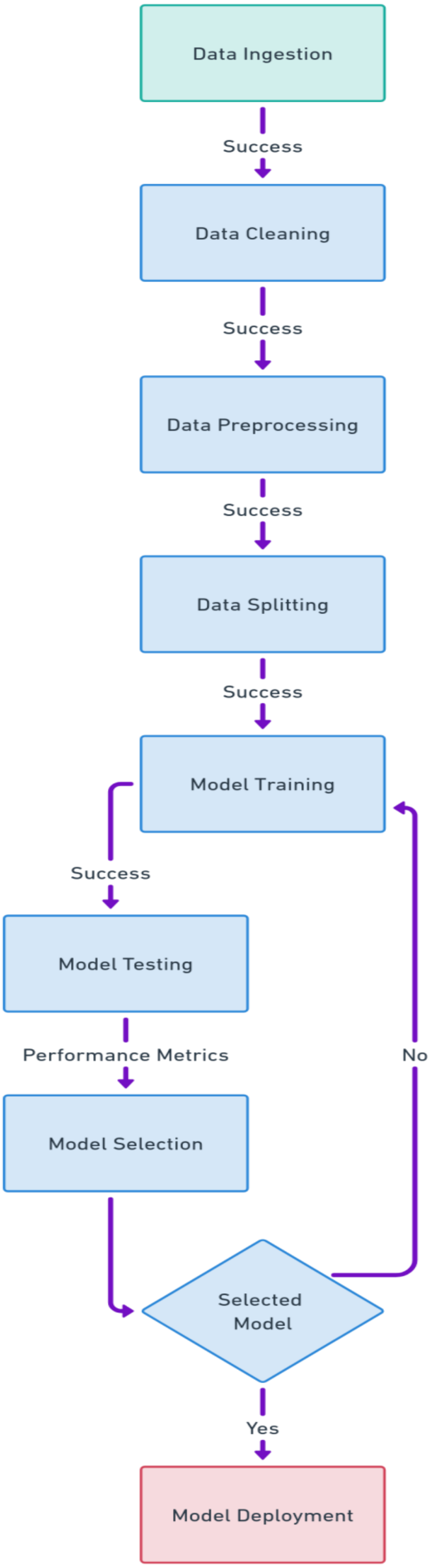
PROJECT ARCHITECTURE

3.1 Architecture

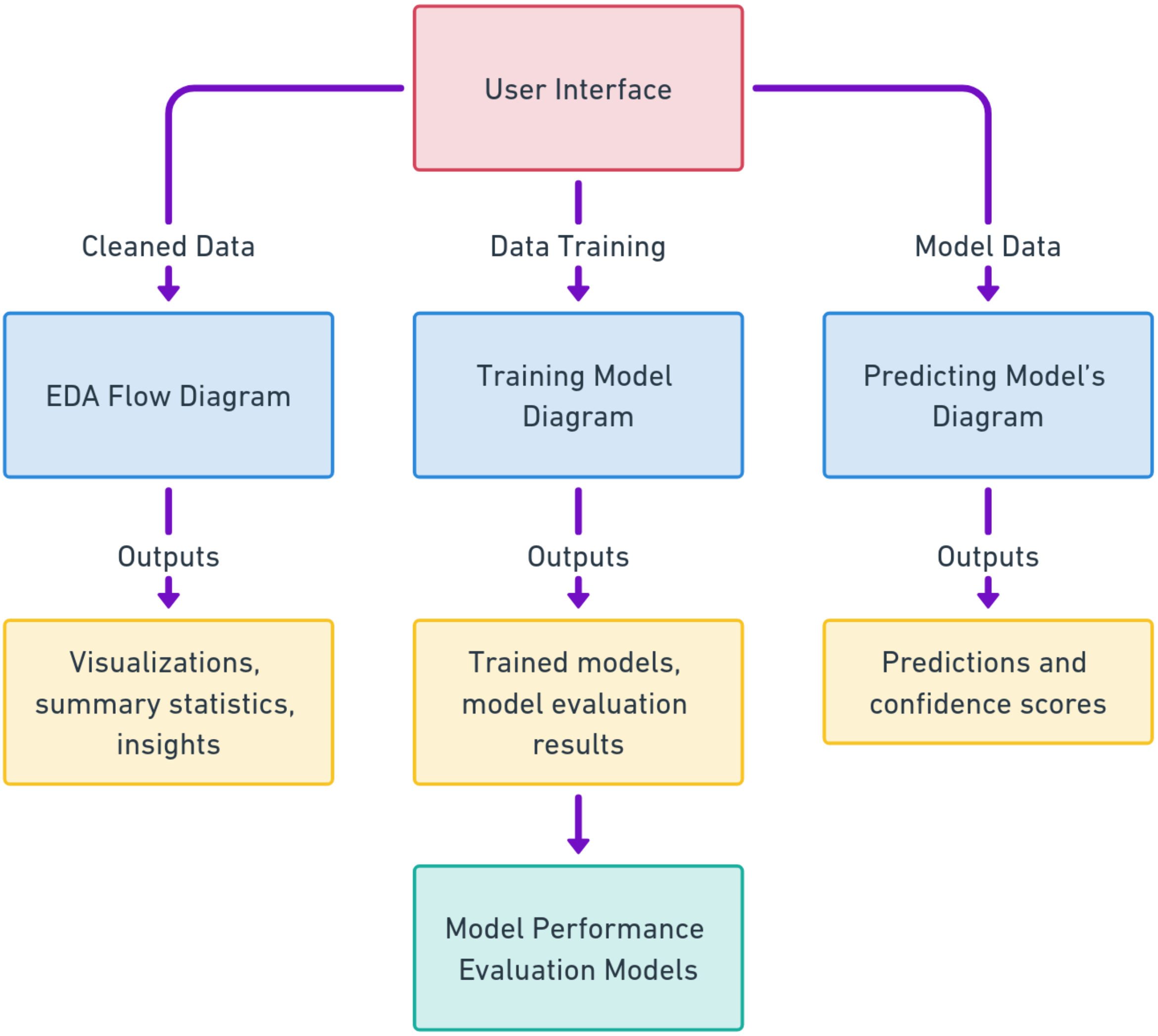
1. System flow diagram



2. Data flow diagram

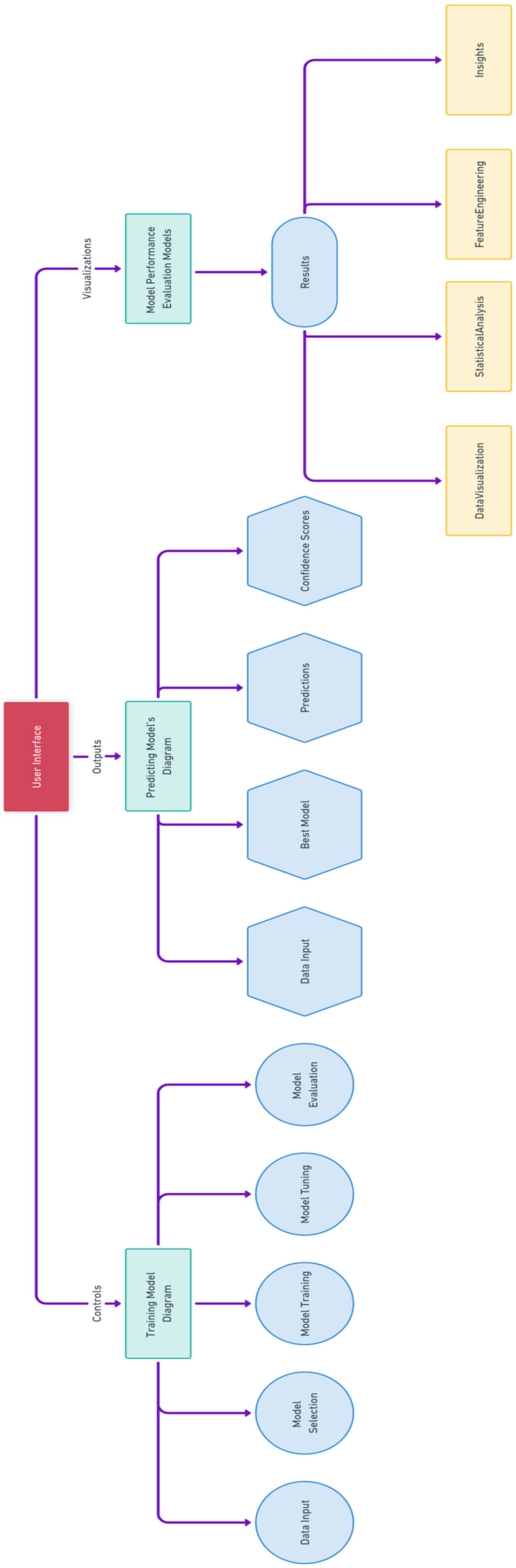


3. Modules


User Interface
Cleaned Data
Data Training
Model Data
EDA Flow Diagram
Training Model Diagram
Predicting Model's Diagram
Outputs
Outputs
Outputs
Visualizations, summary statistics, insights
Trained models, model evaluation results
Predictions and confidence scores
Model Performance Evaluation Models

4. User interface

- Next Module (EDA) flow diagram
- Training model diagram
- Predicting model's diagram
- Model Performance evaluation models


User Interface
Visualizations
Outputs
Controls
Model Performance Evaluation Models
Results
Insights
FeatureEngineering
StatisticalAnalysis
DataVisualization
Predicting Model's Diagram
Confidence Scores
Predictions
Best Model
Data Input
Training Model Diagram
Model Evaluation
Model Tuning
Model Training
Model Selection
Data Input

Here's a high-level architecture for the project:

Data Ingestion:

- **Data Sources:** The system begins with data from various sources such as databases, files, or APIs.
- **Data Collection**: Data is collected and ingested into the system for processing.

Data Cleaning and Preprocessing:

- **Data Cleaning:** Handles missing values, duplicates, and data inconsistencies.
- **Data Preprocessing:** Involves encoding categorical variables, scaling numerical variables, and transforming data for better model performance.

Exploratory Data Analysis (EDA):

- **Visualizations:** Generates visualizations such as histograms, box plots, scatter plots, and heatmaps to understand data distribution and relationships.
- **Statistical Analysis:** Provides summary statistics and feature insights.

Feature Engineering:

- **Feature Selection:** Identifies key features that contribute most to the model's performance.
- **Feature Transformation:** Modifies features as needed (e.g., one-hot encoding for categorical variables).

Model Training:

- **Model Selection:** Chooses models such as linear regression, decision trees, or k-nearest neighbors for training.
- **Model Training:** Trains models using the training dataset.
- **Model Tuning:** Adjusts hyperparameters for optimal model performance.
- **Cross-Validation:** Uses cross-validation techniques to evaluate models and select the best-performing one.

Model Testing and Evaluation:

- **Testing:** Tests the selected model using a separate test dataset.
- **Evaluation Metrics:** Calculates metrics such as accuracy, precision, recall, F1-score, AUC-ROC, and confusion matrix.
- **Model Comparison:** Compares the performance of different models.

Prediction:

- **Prediction:** Uses the best-performing model to make predictions on new, unseen data.
- **Confidence Scores:** Calculates confidence scores for predictions, if applicable.
- **Output:** Presents predictions and confidence scores to the user.

User Interface:

- **Interaction:** Provides a user-friendly interface for data input, model configuration, and output visualization.
- **Feedback:** Offers feedback on data quality and model performance.

Model Deployment:

- **Deployment:** Deploys the best-performing model for real-world use.
- **API:** If applicable, provides an API for external applications to interact with the deployed model.

Feedback Loop:

- **Monitoring:** Monitors the deployed model's performance in real-time.
- **Feedback for Improvement:** Gathers feedback to continually improve the model and data quality.

These high-level components work together to form a comprehensive architecture for a machine learning project, from data ingestion and preprocessing to model training, prediction, and evaluation. The user interface serves as the point of interaction for users to access and utilize the system's capabilities.

CHAPTER 4

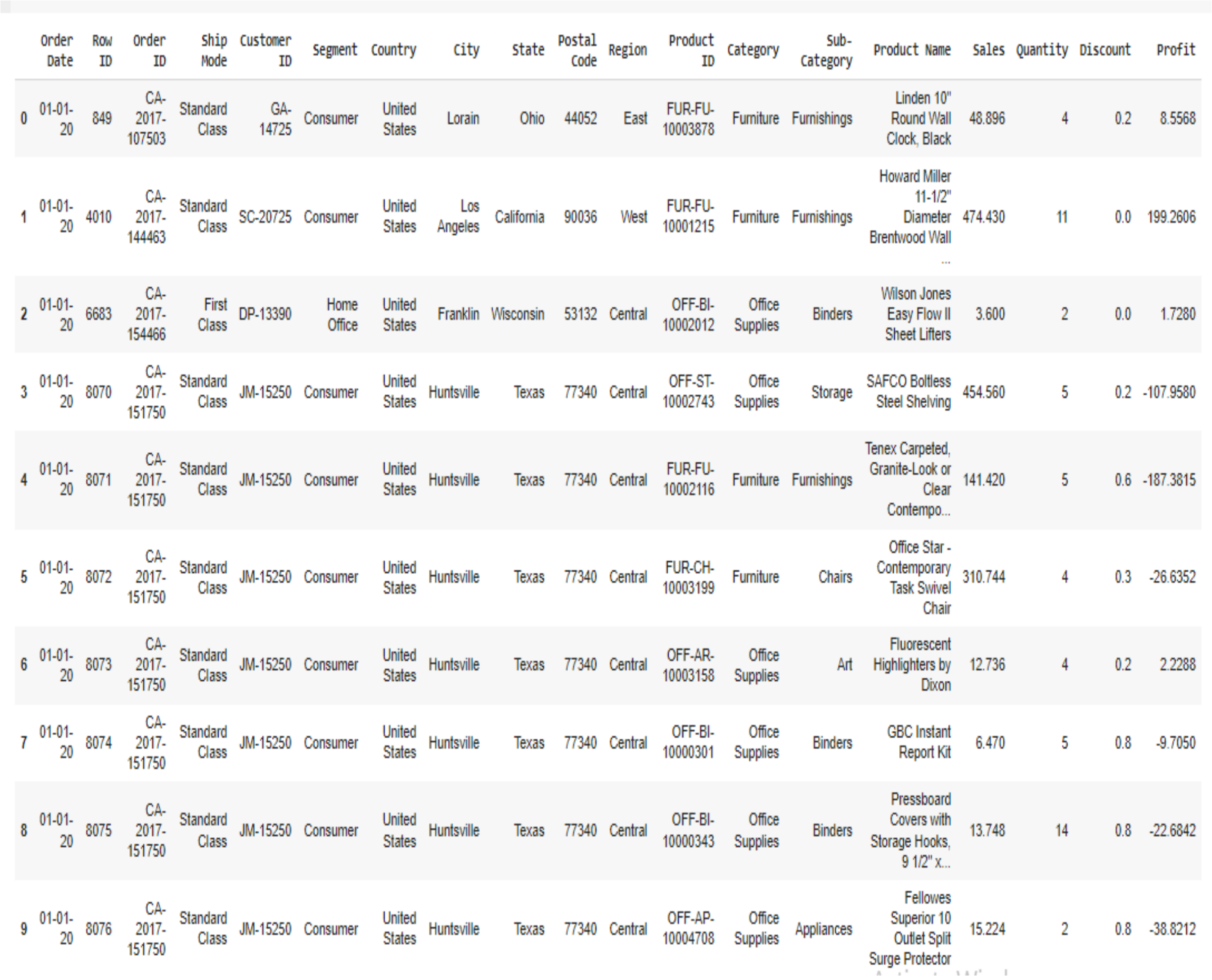
MODELING AND PROJECT OUTCOME

Data load:

Code:

```
df   =   pd.read_csv('/content/US   E-commerce   records   2020.csv',
encoding='latin1')
df.head(10)
```

Ouput:



	Order Date	Row ID	Order ID	Ship Mode	Customer ID	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit
0	01-01-20	849	CA-2017-107503	Standard Class	GA-14725	Consumer	United States	Lorain	Ohio	44052	East	FUR-FU-10003878	Furniture	Furnishings	Linden 10" Round Wall Clock, Black	48.896	4	0.2	8.5568
1	01-01-20	4010	CA-2017-144463	Standard Class	SC-20725	Consumer	United States	Los Angeles	California	90036	West	FUR-FU-10001215	Furniture	Furnishings	Howard Miller 11-1/2" Diameter Brentwood Wall ...	474.430	11	0.0	199.2606
2	01-01-20	6683	CA-2017-154466	First Class	DP-13390	Home Office	United States	Franklin	Wisconsin	53132	Central	OFF-BI-10002012	Office Supplies	Binders	Wilson Jones Easy Flow II Sheet Lifters	3.600	2	0.0	1.7280
3	01-01-20	8070	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	OFF-ST-10002743	Office Supplies	Storage	SAFCO Boltless Steel Shelving	454.560	5	0.2	-107.9580
4	01-01-20	8071	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	FUR-FU-10002116	Furniture	Furnishings	Tenex Carpeted, Granite-Look or Clear Contempo...	141.420	5	0.6	-187.3815
5	01-01-20	8072	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	FUR-CH-10003199	Furniture	Chairs	Office Star - Contemporary Task Swivel Chair	310.744	4	0.3	-26.6352
6	01-01-20	8073	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	OFF-AR-10003158	Office Supplies	Art	Fluorescent Highlighters by Dixon	12.736	4	0.2	2.2288
7	01-01-20	8074	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	OFF-BI-10000301	Office Supplies	Binders	GBC Instant Report Kit	6.470	5	0.8	-9.7050
8	01-01-20	8075	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	OFF-BI-10000343	Office Supplies	Binders	Pressboard Covers with Storage Hooks, 9 1/2" x...	13.748	14	0.8	-22.6842
9	01-01-20	8076	CA-2017-151750	Standard Class	JM-15250	Consumer	United States	Huntsville	Texas	77340	Central	OFF-AP-10004708	Office Supplies	Appliances	Fellowes Superior 10 Outlet Split Surge Protector	15.224	2	0.8	-38.8212

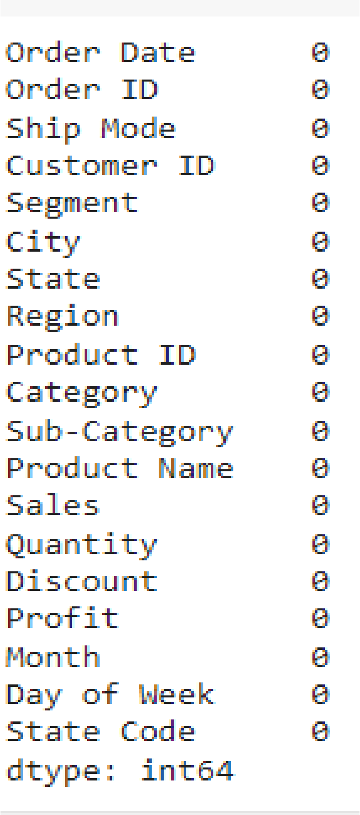
EDA – analysis report:

Handling missing values:

Code:

```
missing_values = df.isnull().sum()
```

Output:



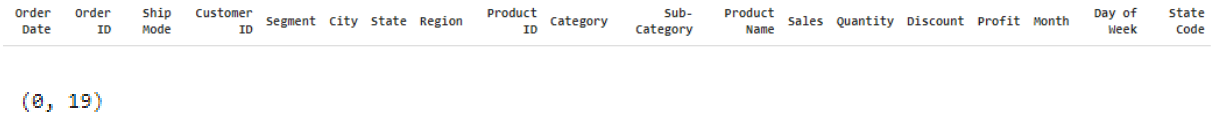
```
Order Date       0
Order ID         0
Ship Mode        0
Customer ID      0
Segment          0
City             0
State            0
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
Month            0
Day of Week      0
State Code       0
dtype: int64
```

Duplicate:

Code:

```
duplicate_rows = df[df.duplicated()]
df = df.drop_duplicates()
duplicate_rows.shape
```

Output:



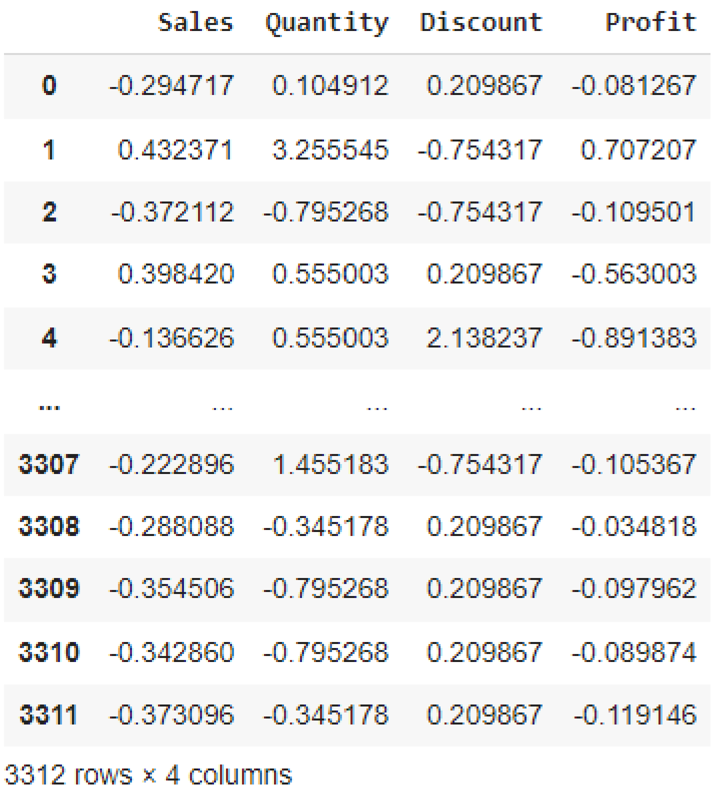
```
Order  Order  Ship  Customer                          Product               Sub-      Product                                                  Day of  State
Date      ID  Mode        ID  Segment  City  State  Region       ID  Category  Category     Name  Sales  Quantity  Discount  Profit  Month    Week   Code

(0, 19)
```

Normalization:

Code:

```
numerical_cols = df.select_dtypes(include=np.number).columns
df[numerical_cols] = (df[numerical_cols] -
df[numerical_cols].mean()) / df[numerical_cols].std()
```

Output:



	Sales	Quantity	Discount	Profit
0	-0.294717	0.104912	0.209867	-0.081267
1	0.432371	3.255545	-0.754317	0.707207
2	-0.372112	-0.795268	-0.754317	-0.109501
3	0.398420	0.555003	0.209867	-0.563003
4	-0.136626	0.555003	2.138237	-0.891383
...	...	...	...	...
3307	-0.222896	1.455183	-0.754317	-0.105367
3308	-0.288088	-0.345178	0.209867	-0.034818
3309	-0.354506	-0.795268	0.209867	-0.097962
3310	-0.342860	-0.795268	0.209867	-0.089874
3311	-0.373096	-0.345178	0.209867	-0.119146

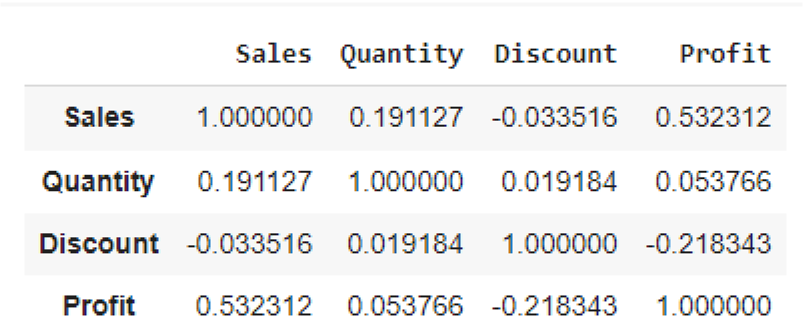
3312 rows × 4 columns

Correlation:

Code:

```
numerical_columns = df.select_dtypes(include=[np.number]).columns
correlation_matrix = df[numerical_columns].corr()
```

Output:



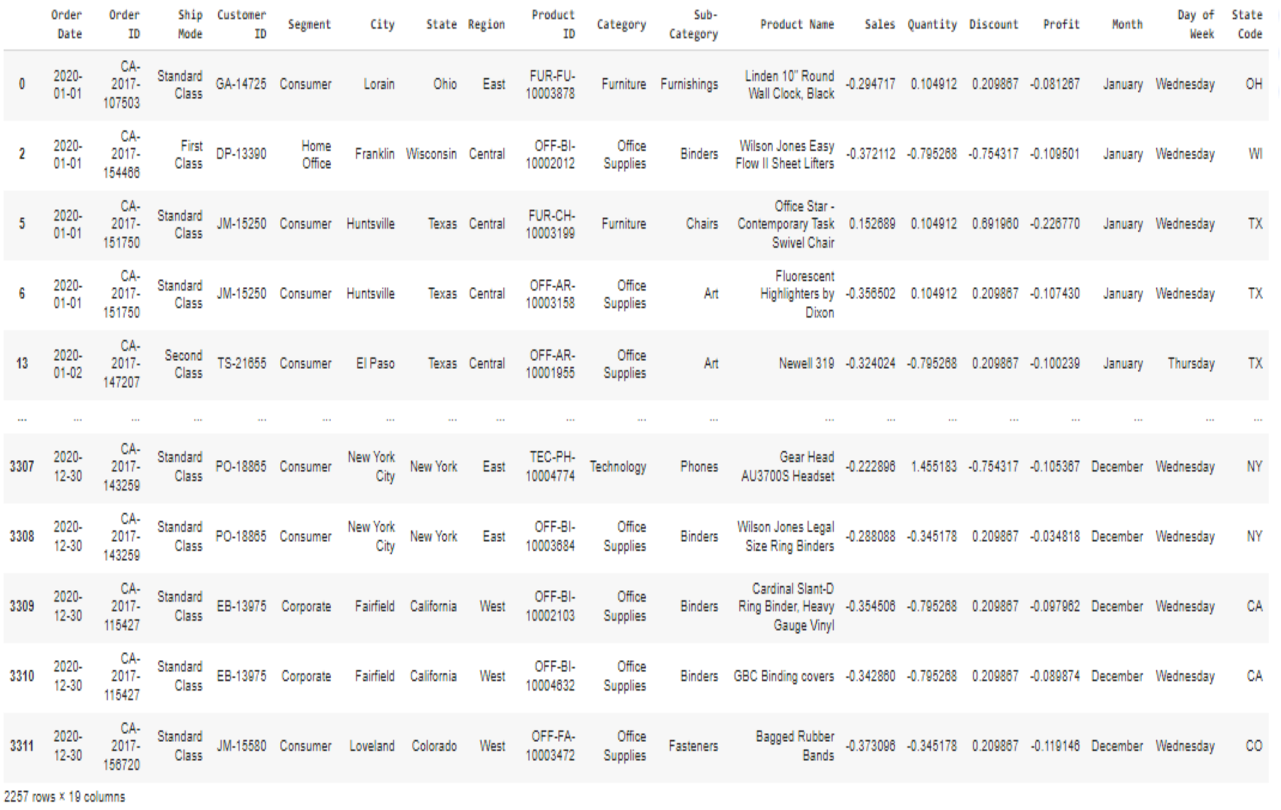
	Sales	Quantity	Discount	Profit
Sales	1.000000	0.191127	-0.033516	0.532312
Quantity	0.191127	1.000000	0.019184	0.053766
Discount	-0.033516	0.019184	1.000000	-0.218343
Profit	0.532312	0.053766	-0.218343	1.000000

Outlier:

Code:

```
def detect_outliers(column):
    Q1 = np.percentile(column, 25)
    Q3 = np.percentile(column, 75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return column[(column < lower_bound) | (column > upper_bound)]
for col in numerical_cols:
    outliers = detect_outliers(df[col])
    df = df[~df[col].isin(outliers)]
```

Output:



	Order Date	Order ID	Ship Mode	Customer ID	Segment	City	State	Region	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit	Month	Day of Week	State Code
0	2020-01-01	CA-2017-107503	Standard Class	GA-14725	Consumer	Lorain	Ohio	East	FUR-FU-10003878	Furniture	Furnishings	Linden 10" Round Wall Clock, Black	-0.294717	0.104912	0.209867	-0.081267	January	Wednesday	OH
2	2020-01-01	CA-2017-154466	First Class	DP-13390	Home Office	Franklin	Wisconsin	Central	OFF-BI-10002012	Office Supplies	Binders	Wilson Jones Easy Flow II Sheet Lifters	-0.372112	-0.795268	-0.754317	-0.109501	January	Wednesday	WI
5	2020-01-01	CA-2017-151750	Standard Class	JM-15250	Consumer	Huntsville	Texas	Central	FUR-CH-10003199	Furniture	Chairs	Office Star - Contemporary Task Swivel Chair	0.152689	0.104912	0.691960	-0.226770	January	Wednesday	TX
6	2020-01-01	CA-2017-151750	Standard Class	JM-15250	Consumer	Huntsville	Texas	Central	OFF-AR-10003158	Office Supplies	Art	Fluorescent Highlighters by Dixon	-0.356502	0.104912	0.209867	-0.107430	January	Wednesday	TX
13	2020-01-02	CA-2017-147207	Second Class	TS-21655	Consumer	El Paso	Texas	Central	OFF-AR-10001955	Office Supplies	Art	Newell 319	-0.324024	-0.795268	0.209867	-0.100239	January	Thursday	TX
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3307	2020-12-30	CA-2017-143259	Standard Class	PO-18865	Consumer	New York City	New York	East	TEC-PH-10004774	Technology	Phones	Gear Head AU3700S Headset	-0.222896	1.455183	-0.754317	-0.105367	December	Wednesday	NY
3308	2020-12-30	CA-2017-143259	Standard Class	PO-18865	Consumer	New York City	New York	East	OFF-BI-10003684	Office Supplies	Binders	Wilson Jones Legal Size Ring Binders	-0.288088	-0.345178	0.209867	-0.034818	December	Wednesday	NY
3309	2020-12-30	CA-2017-115427	Standard Class	EB-13975	Corporate	Fairfield	California	West	OFF-BI-10002103	Office Supplies	Binders	Cardinal Slant-D Ring Binder, Heavy Gauge Vinyl	-0.354506	-0.795268	0.209867	-0.097962	December	Wednesday	CA
3310	2020-12-30	CA-2017-115427	Standard Class	EB-13975	Corporate	Fairfield	California	West	OFF-BI-10004632	Office Supplies	Binders	GBC Binding covers	-0.342860	-0.795268	0.209867	-0.089874	December	Wednesday	CA
3311	2020-12-30	CA-2017-156720	Standard Class	JM-15580	Consumer	Loveland	Colorado	West	OFF-FA-10003472	Office Supplies	Fasteners	Bagged Rubber Bands	-0.373096	-0.345178	0.209867	-0.119148	December	Wednesday	CO

2257 rows × 19 columns

Data Visualizations:

Code:

```
# Histogram for numerical columns
df[numerical_cols].hist(figsize=(10, 10), bins=30)
plt.suptitle("Histograms of Numerical Columns")
plt.show()

# Violin plot of Discount and Month
fig = px.violin(df, y="Discount", x="Month", box=True, points="all",
          hover_data=df.columns)
fig.update_layout(title={
        'text': "Discount Density by Months",
        'y':0.92,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'})
fig.show()
```

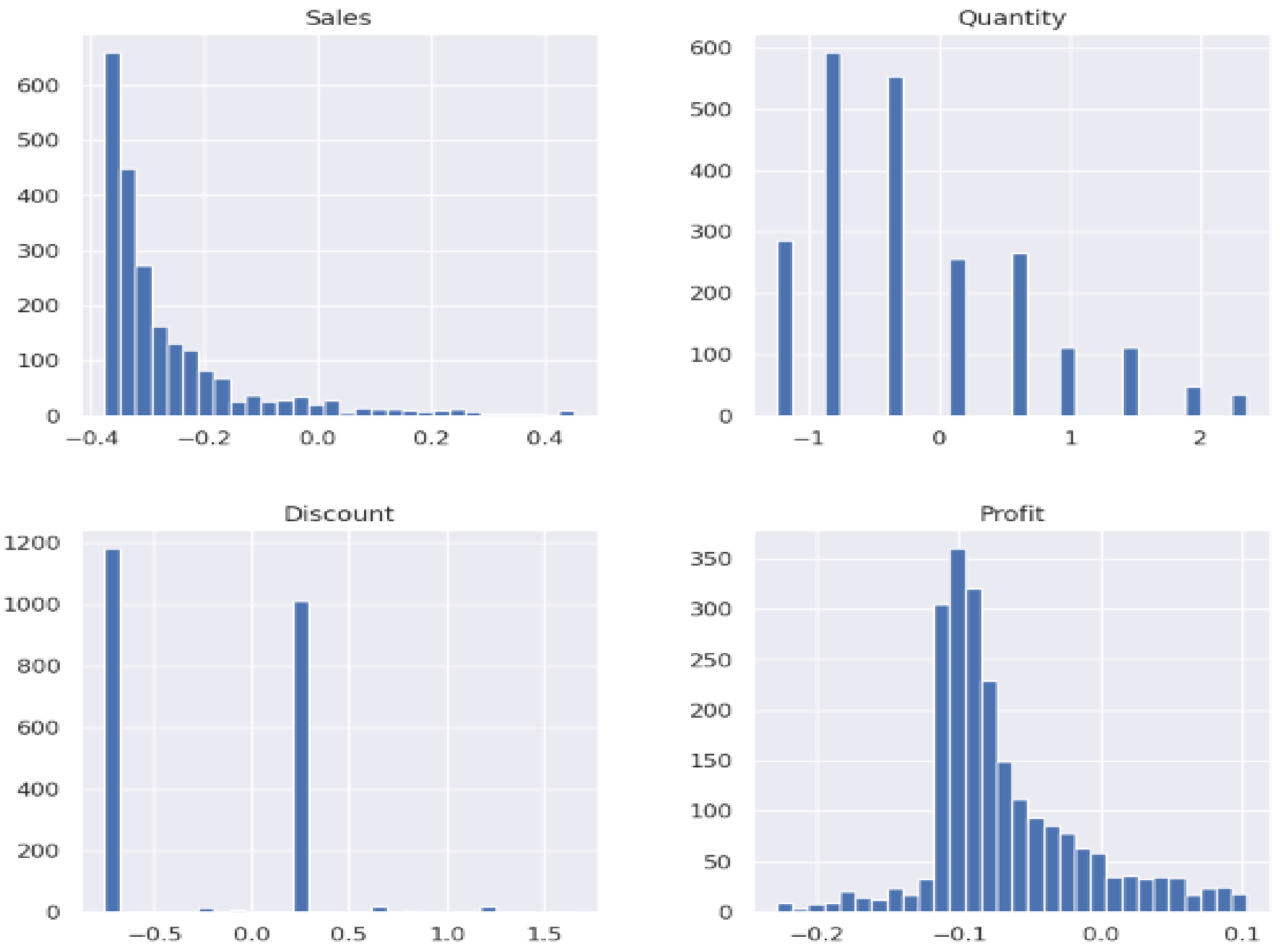
```
# Boxplot for numerical columns
numerical_cols =
df.select_dtypes(include=np.number).columns.tolist()
fig, ax = plt.subplots(figsize=(10, 8))
df.boxplot(column=numerical_cols, ax=ax, patch_artist=True,
boxprops=dict(facecolor='lightblue'))
plt.title("Boxplots of Numerical Columns")
plt.show()
```

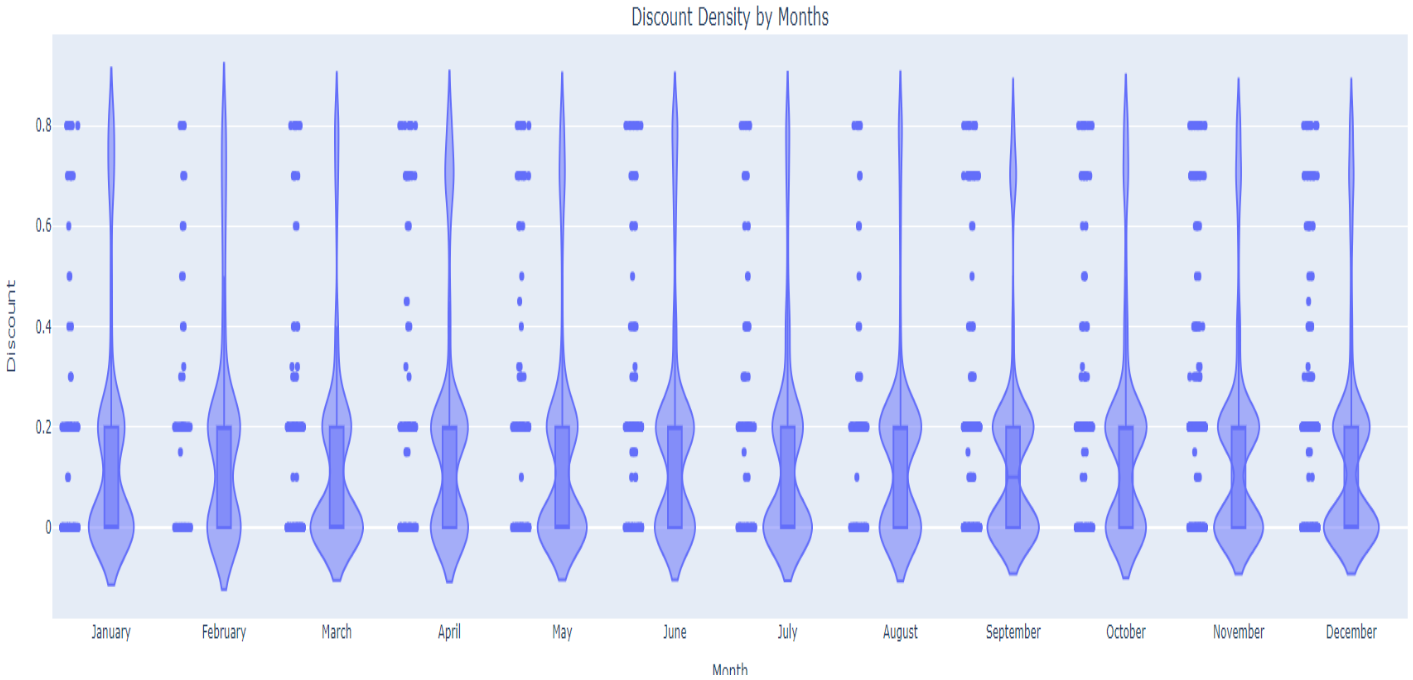
```
# Violin plot for numerical columns
sns.violinplot(data=df[numerical_cols])
plt.title("Violin Plots of Numerical Columns")
plt.show()
```

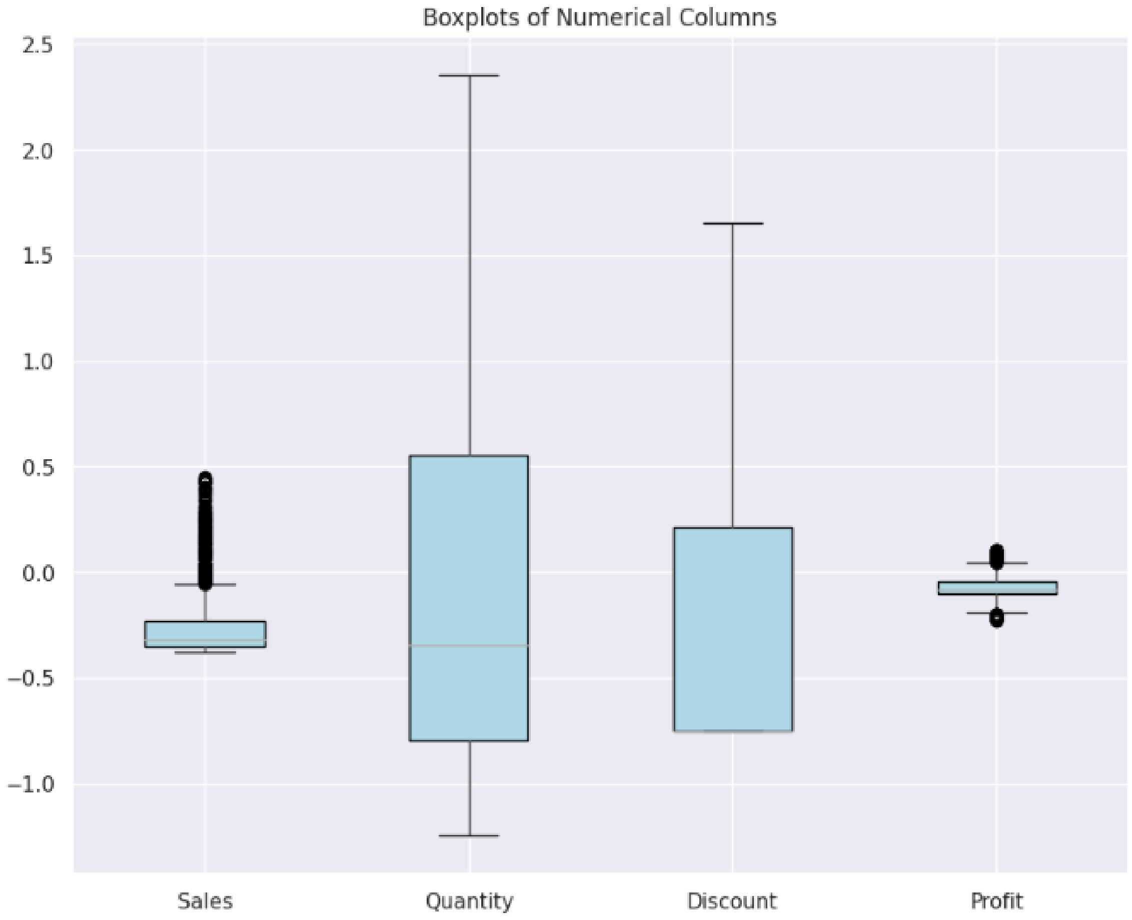
```
# Scatter plot of Sales and Profit
sns.scatterplot(x='Sales', y='Profit', data=df)
plt.title("Scatter Plot of Sales vs. Profit")
plt.show()
```

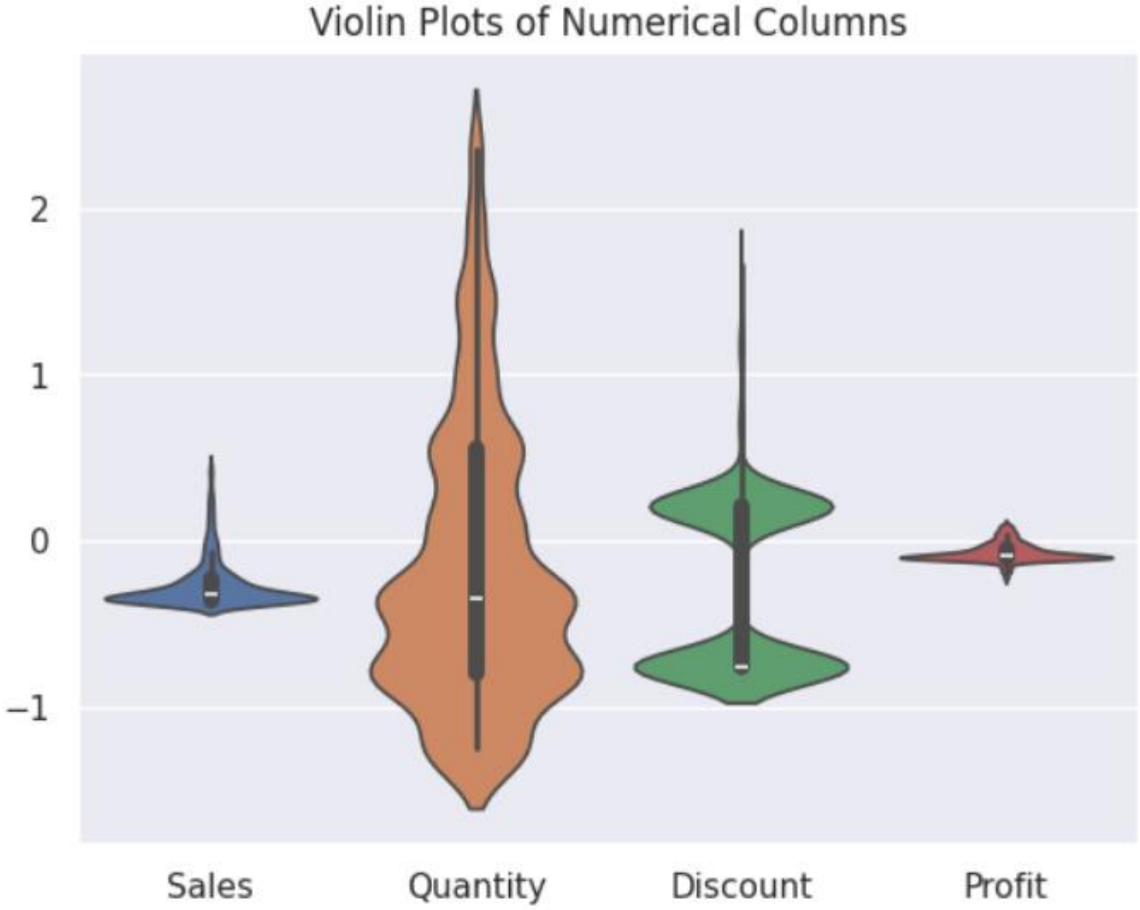
```
# Strip plot of Quantity and Sales
fig = px.strip(df, x="Quantity", y="Sales", orientation="h",
color="Ship Mode")
fig.show()
```

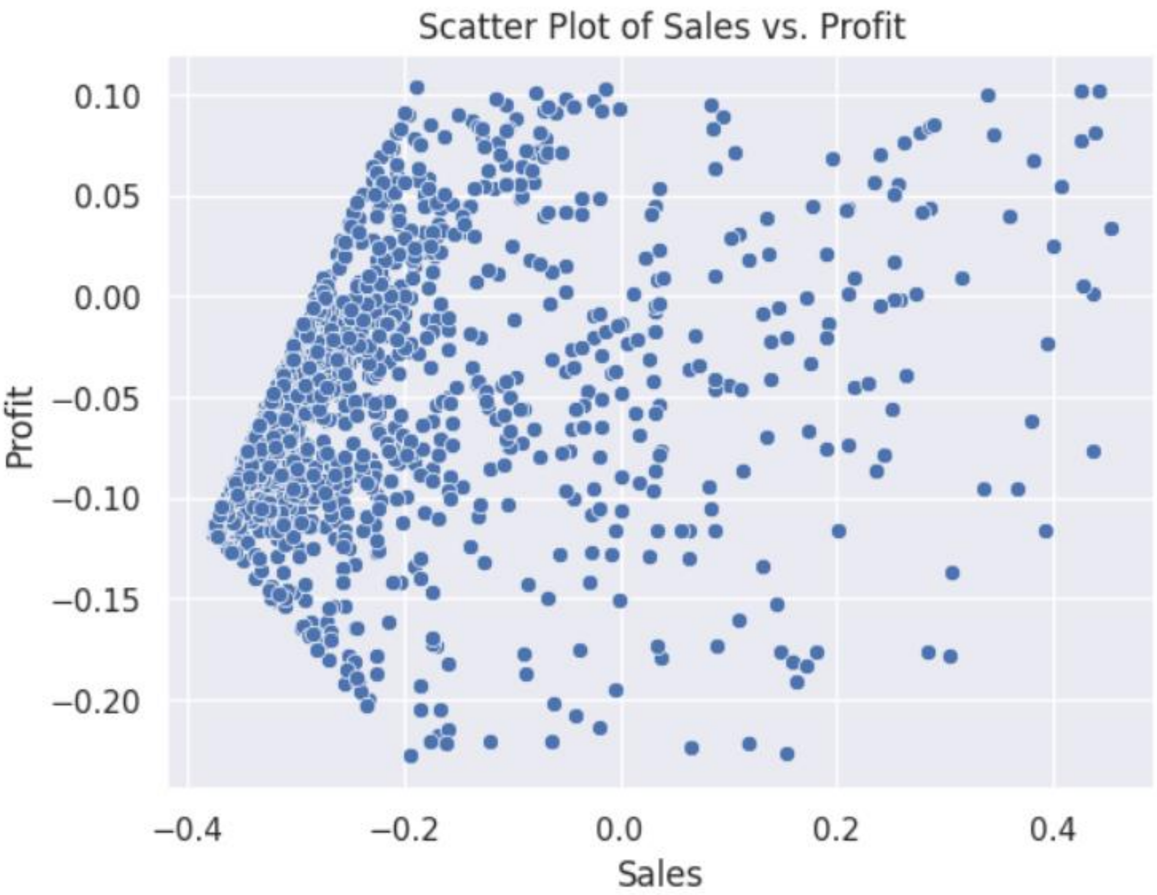
Output:

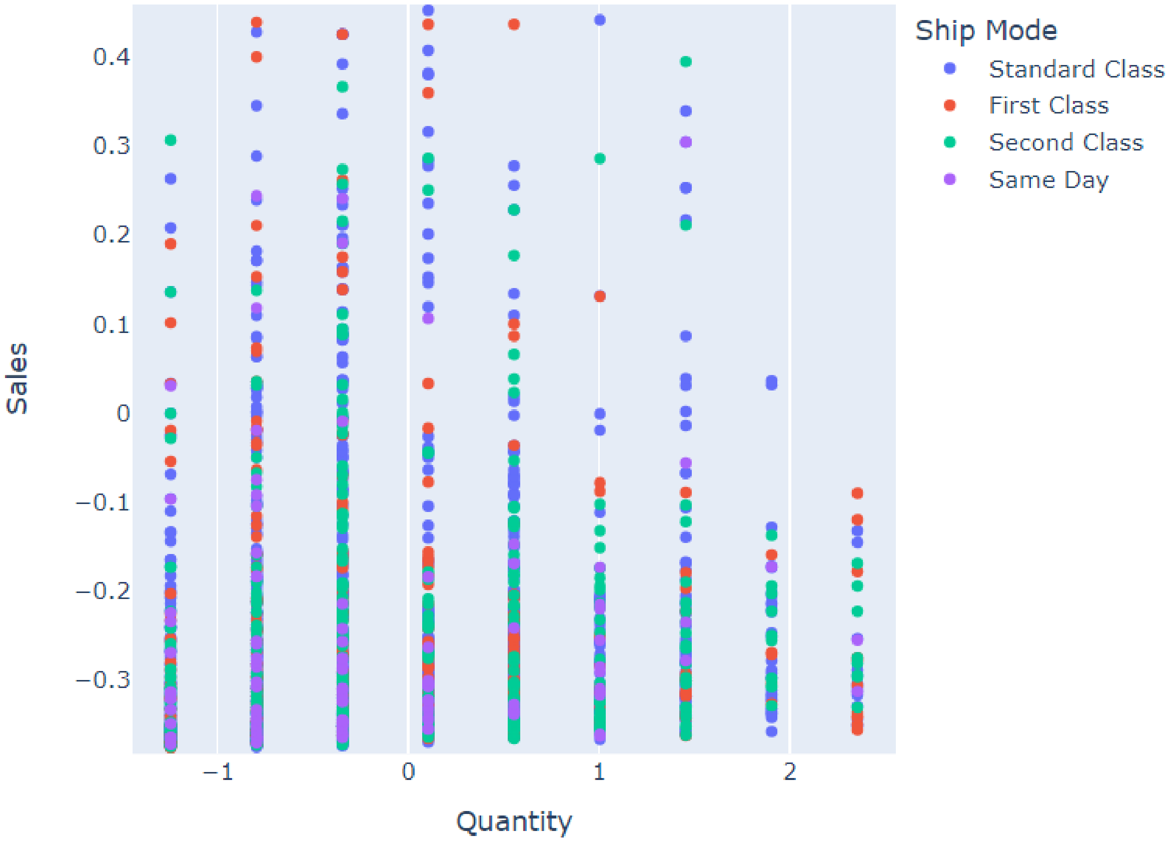

Sales
600
500
400
300
200
100
0
−0.4
−0.2
0.0
0.2
0.4
Quantity
600
500
400
300
200
100
0
−1
0
1
2
Discount
1200
1000
800
600
400
200
0
−0.5
0.0
0.5
1.0
1.5
Profit
350
300
250
200
150
100
50
0
−0.2
−0.1
0.0
0.1


Discount Density by Months
Discount
0.8
0.6
0.4
0.2
0
January
February
March
April
May
June
July
August
September
October
November
December
Month


Boxplots of Numerical Columns
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
Sales
Quantity
Discount
Profit


Violin Plots of Numerical Columns
2
1
0
−1
Sales
Quantity
Discount
Profit


Scatter Plot of Sales vs. Profit
Profit
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.4
−0.2
0.0
0.2
0.4
Sales


Sales
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−1
0
1
2
Quantity
Ship Mode
● Standard Class
● First Class
● Second Class
● Same Day

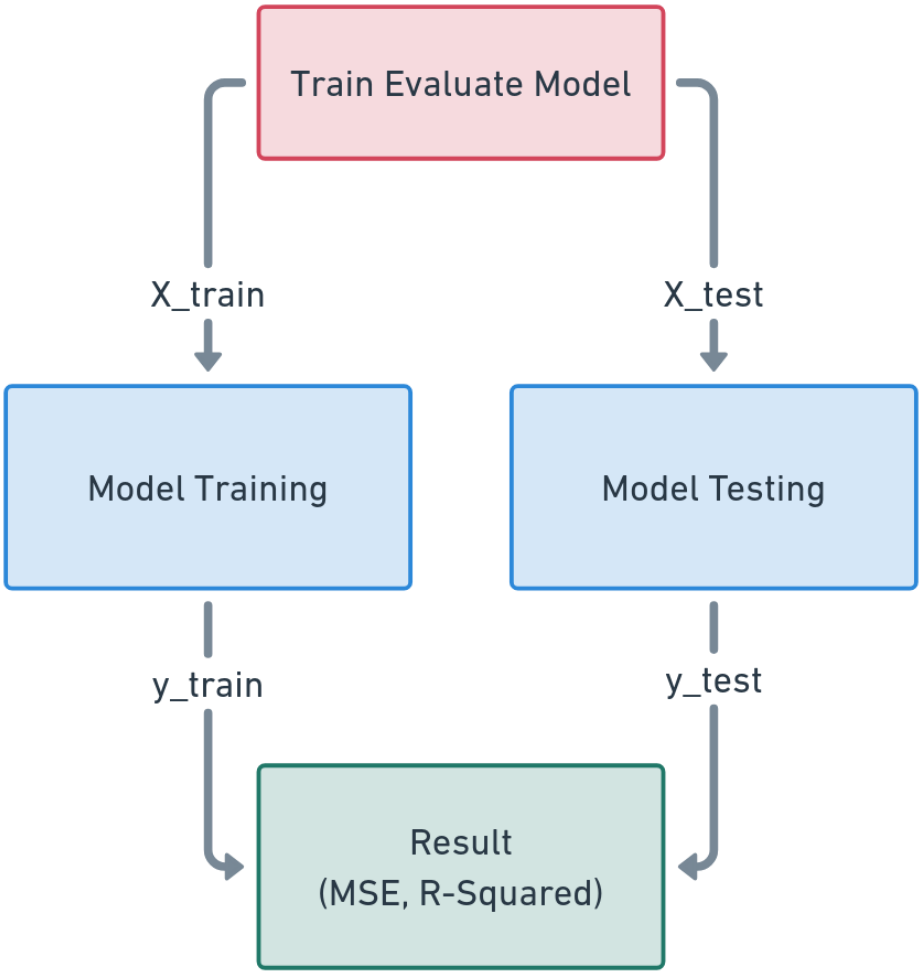
Model output:

LR :

Code:

```
# Linear Regression (LR)
# Define a function for training and evaluating models
def train_evaluate_model(model, X_train, y_train, X_test, y_test,
model_name):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    return {
        'Model': model_name,
        'Mean Squared Error': mse,
        'R-squared': r2
    }
```

How it working (Diagram):

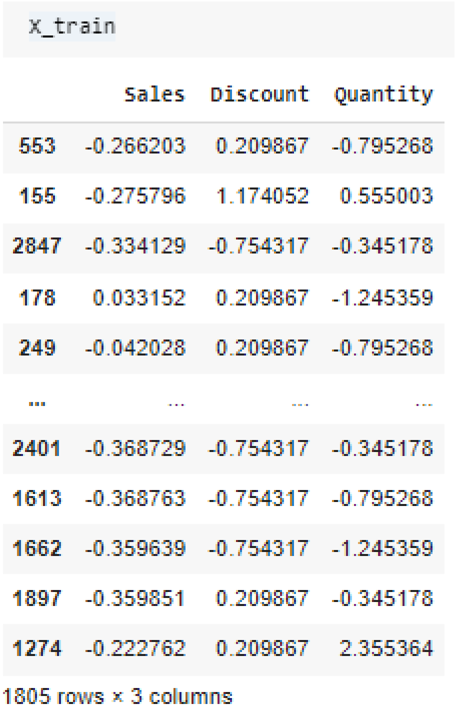


Training data – Model performance:

Code:

```
# Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

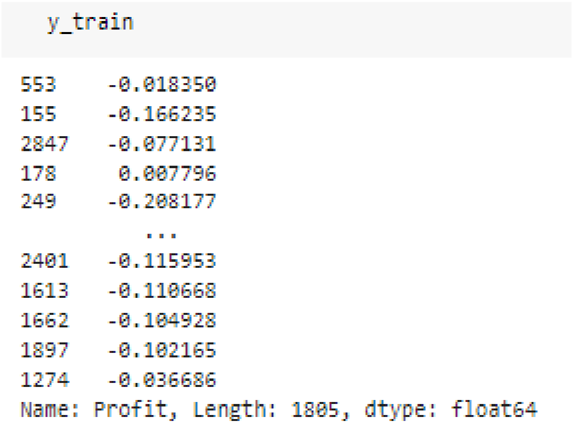
Output:



```
X_train
```

	Sales	Discount	Quantity
553	-0.266203	0.209867	-0.795268
155	-0.275796	1.174052	0.555003
2847	-0.334129	-0.754317	-0.345178
178	0.033152	0.209867	-1.245359
249	-0.042028	0.209867	-0.795268
...	...	...	...
2401	-0.368729	-0.754317	-0.345178
1613	-0.368763	-0.754317	-0.795268
1662	-0.359639	-0.754317	-1.245359
1897	-0.359851	0.209867	-0.345178
1274	-0.222762	0.209867	2.355364

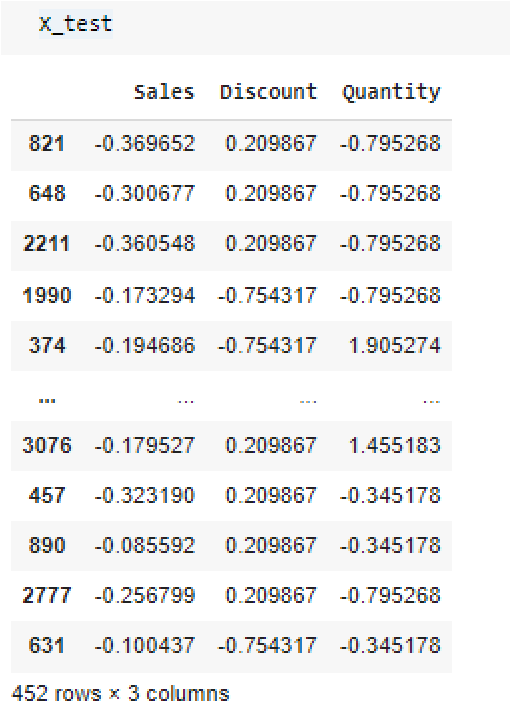
1805 rows × 3 columns



```
y_train
```

```
553     -0.018350
155     -0.166235
2847    -0.077131
178      0.007796
249     -0.208177
           ...
2401    -0.115953
1613    -0.110668
1662    -0.104928
1897    -0.102165
1274    -0.036686
Name: Profit, Length: 1805, dtype: float64
```

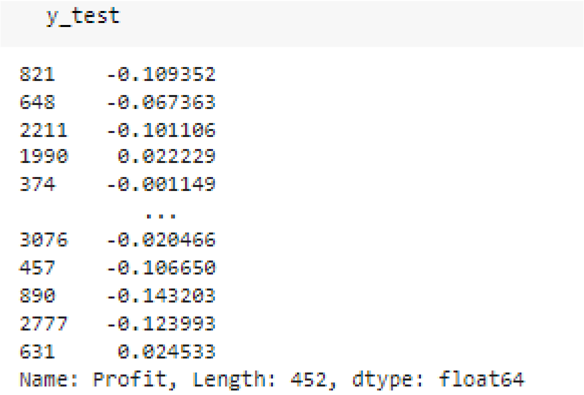
Output – Testing output:



```
x_test
```

	Sales	Discount	Quantity
821	-0.369652	0.209867	-0.795268
648	-0.300677	0.209867	-0.795268
2211	-0.360548	0.209867	-0.795268
1990	-0.173294	-0.754317	-0.795268
374	-0.194686	-0.754317	1.905274
...	...	...	...
3076	-0.179527	0.209867	1.455183
457	-0.323190	0.209867	-0.345178
890	-0.085592	0.209867	-0.345178
2777	-0.256799	0.209867	-0.795268
631	-0.100437	-0.754317	-0.345178

452 rows × 3 columns



```
y_test
```

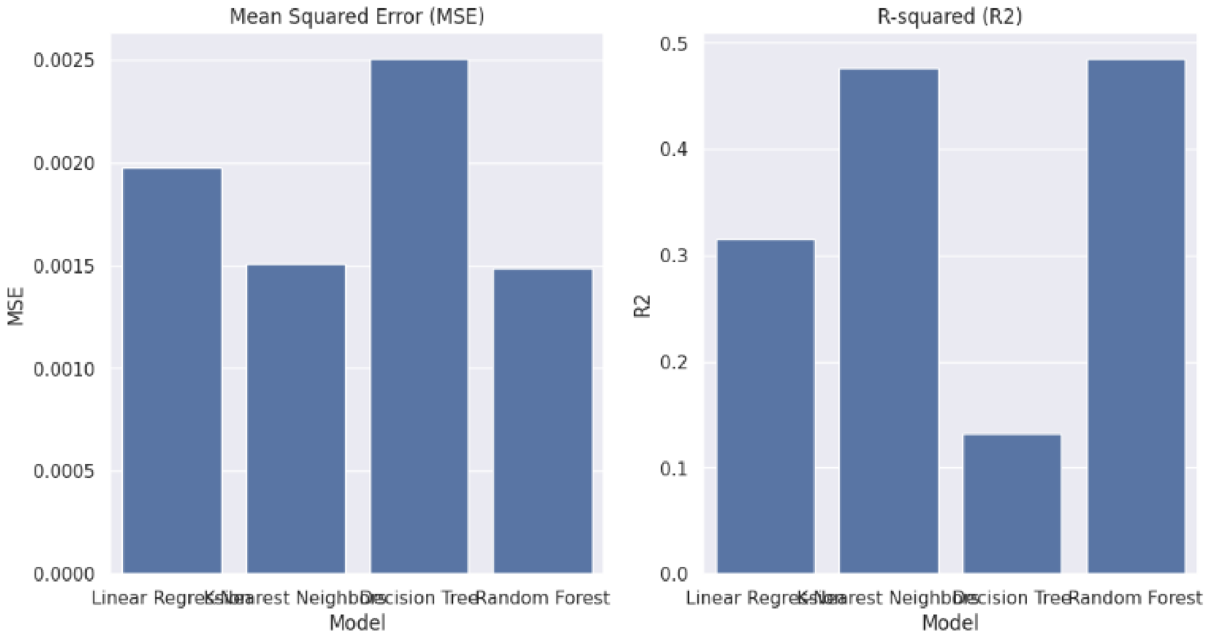
```
821    -0.109352
648    -0.067363
2211   -0.101106
1990    0.022229
374    -0.001149
          ...   
3076   -0.020466
457    -0.106650
890    -0.143203
2777   -0.123993
631     0.024533
Name: Profit, Length: 452, dtype: float64
```

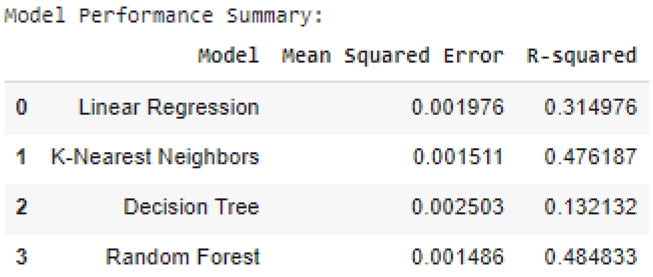
Model Performance Analysis :

Code:

```
# Define models
linear_regression_model = LinearRegression()
knn_model = KNeighborsRegressor()
decision_tree_model = DecisionTreeRegressor()
random_forest_model = RandomForestRegressor()
# Train and evaluate models
results = []
results.append(train_evaluate_model(linear_regression_model,
X_train, y_train, X_test, y_test, 'Linear Regression'))
results.append(train_evaluate_model(knn_model, X_train, y_train,
X_test, y_test, 'K-Nearest Neighbors'))
results.append(train_evaluate_model(decision_tree_model, X_train,
y_train, X_test, y_test, 'Decision Tree'))
results.append(train_evaluate_model(random_forest_model, X_train,
y_train, X_test, y_test, 'Random Forest'))
# Create a DataFrame to store the performance results
results_df = pd.DataFrame(results, columns=['Model', 'Mean Squared
Error', 'R-squared'])
# Plot the performance of different models
fig, axes = plt.subplots(1, 2, figsize=(12, 6))
# Plot Mean Squared Error for each model
sns.barplot(x='Model', y='Mean Squared Error', data=results_df,
ax=axes[0])
axes[0].set_title('Mean Squared Error (MSE)')
axes[0].set_ylabel('MSE')
# Plot R-squared for each model
sns.barplot(x='Model', y='R-squared', data=results_df, ax=axes[1])
axes[1].set_title('R-squared (R2)')
axes[1].set_ylabel('R2')
plt.show()

# Model Performance Summary:
print("\nModel Performance Summary:")
results_df
```

Output:




Model Performance Summary:

	Model	Mean Squared Error	R-squared
0	Linear Regression	0.001976	0.314976
1	K-Nearest Neighbors	0.001511	0.476187
2	Decision Tree	0.002503	0.132132
3	Random Forest	0.001486	0.484833

Manage relationship:

There are several aspects to consider, especially given the various transformations, data visualizations, and analyses you aim to perform on the dataset. Here are some considerations for managing relationships in the dataset:

1. Feature Relationships:

- **Correlation Analysis:** You can use correlation heatmaps to analyze the relationships between numerical features in the dataset. Understanding which features are highly correlated can guide feature selection and modeling decisions.
- **Category and Subcategory Relationships:** Investigate how different categories and subcategories relate to one another, especially in terms of sales and quantity. Grouping similar categories and subcategories together might provide better insights and improve modeling.
- **Temporal Relationships:** Examine how data trends change over time, such as monthly changes in sales or quantities. Analyzing temporal patterns helps in understanding seasonality and can influence future predictions.

2. User and Customer Relationships:

- **Customer Purchase Patterns:** Analyze how different customer segments (e.g., Consumer, Corporate, Home Office) interact with different product categories and subcategories. This can reveal patterns in purchasing behavior.
- **Order and Customer Relationships:** Investigate the relationship between order volume and customer retention to identify high-value customers and strategize personalized marketing efforts.

3. Location Relationships:

- **Regional and State Relationships:** Examine how sales and quantities vary across different regions and states. This can help identify regional preferences and guide targeted marketing strategies.
- **City-level Analysis:** Consider city-level relationships, such as how different cities perform in terms of sales and quantity of orders. This may help optimize inventory management and supply chain operations.

4. Visual Relationships:

- **Pair Plot Analysis:** Visualizing pair plots for key features allows you to see relationships between different pairs of features, revealing potential trends or clusters.
- **Violin and Box Plots:** Use these plots to visualize data distributions and relationships, particularly in categories and subcategories, as well as different segments or other categorical features.

5. Predictive Model Relationships:

- **Model Evaluation:** By evaluating model performance metrics, you can understand how different features contribute to the model and whether there are any dependencies or relationships affecting model accuracy.
- **Prediction Confidence:** Understanding the confidence in predictions can highlight areas where the model performs well and areas where further investigation may be needed.

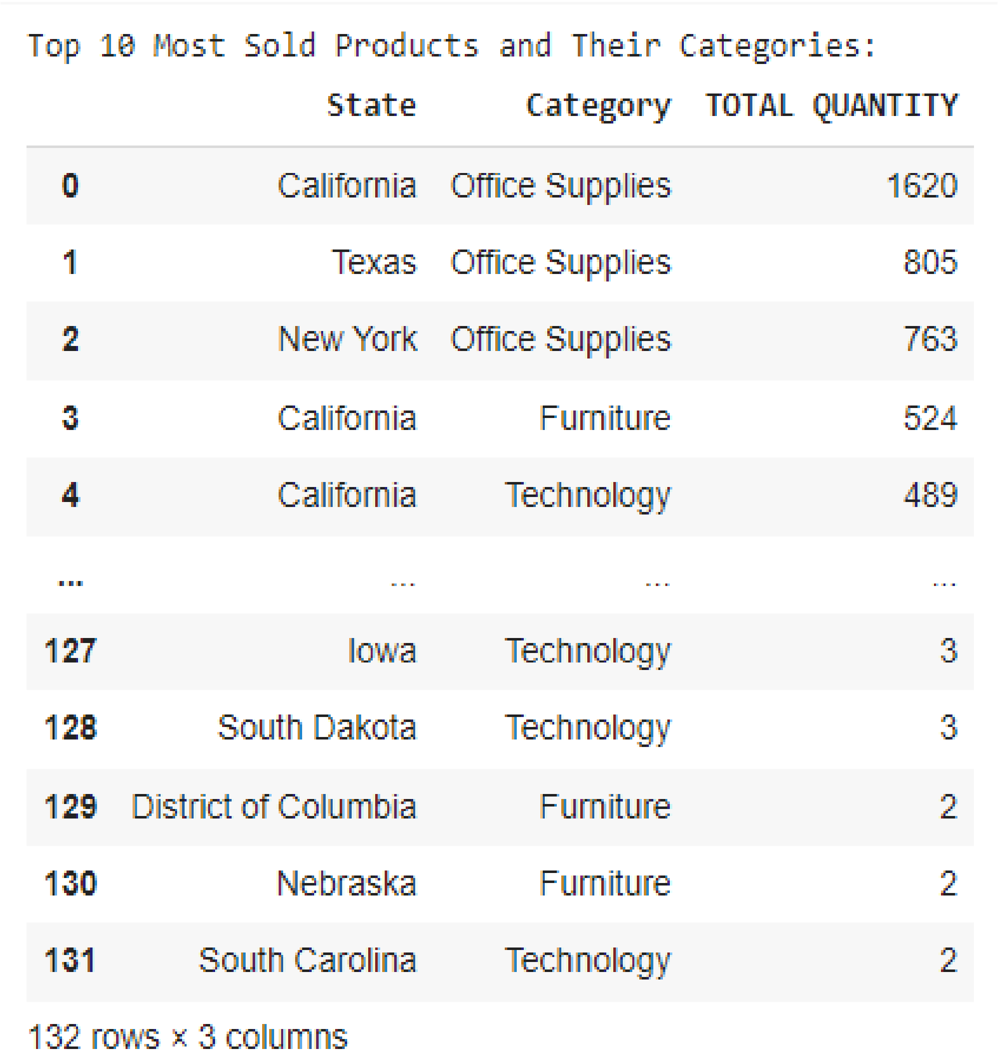
Managing Relationships Effectively

- **Data Cleaning and Preprocessing:** Ensure that relationships are accurately represented by handling missing values, duplicates, and data inconsistencies.
- **Feature Engineering:** Create derived features that capture relationships between different data elements to improve modeling.
- **Feedback Loop:** Continuously monitor the model's performance and adjust relationships as needed based on new insights and changing data patterns.

Code Organization

- **Separation of Concerns:** Organize your code into distinct sections, each responsible for a specific aspect of the analysis, such as data preprocessing, feature engineering, and model training.
- **Use Functions:** Encapsulate tasks such as data cleaning, feature engineering, and visualization into functions to manage complexity and improve code readability.

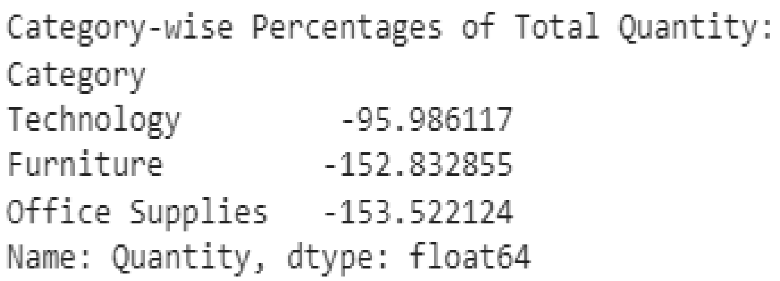
Project result:



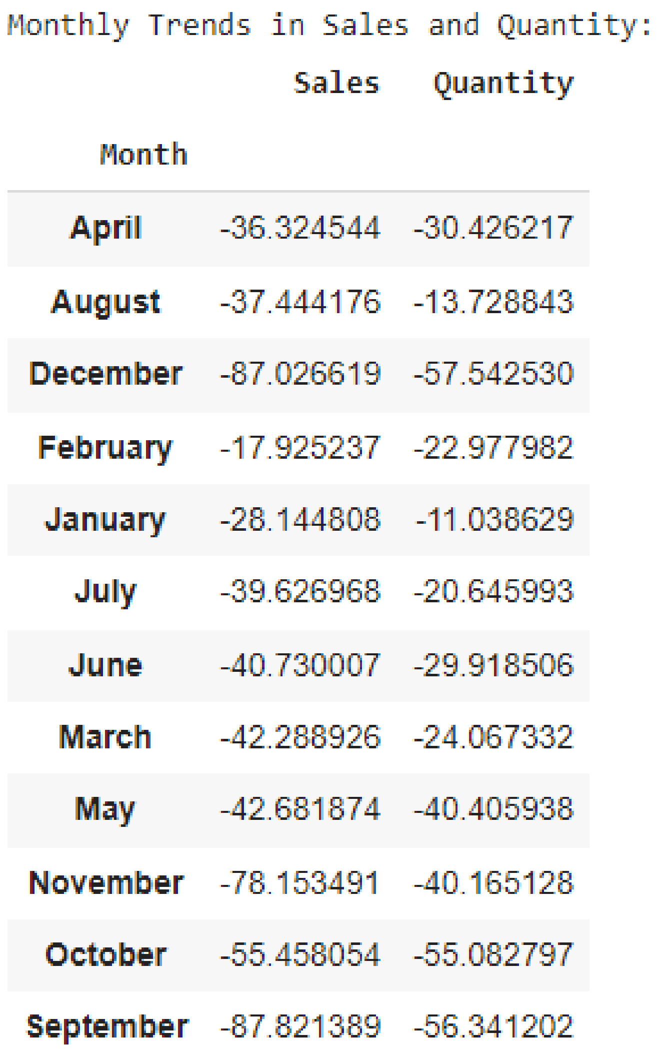
Top 10 Most Sold Products and Their Categories:

	State	Category	TOTAL QUANTITY
0	California	Office Supplies	1620
1	Texas	Office Supplies	805
2	New York	Office Supplies	763
3	California	Furniture	524
4	California	Technology	489
...	...	...	...
127	Iowa	Technology	3
128	South Dakota	Technology	3
129	District of Columbia	Furniture	2
130	Nebraska	Furniture	2
131	South Carolina	Technology	2

132 rows × 3 columns

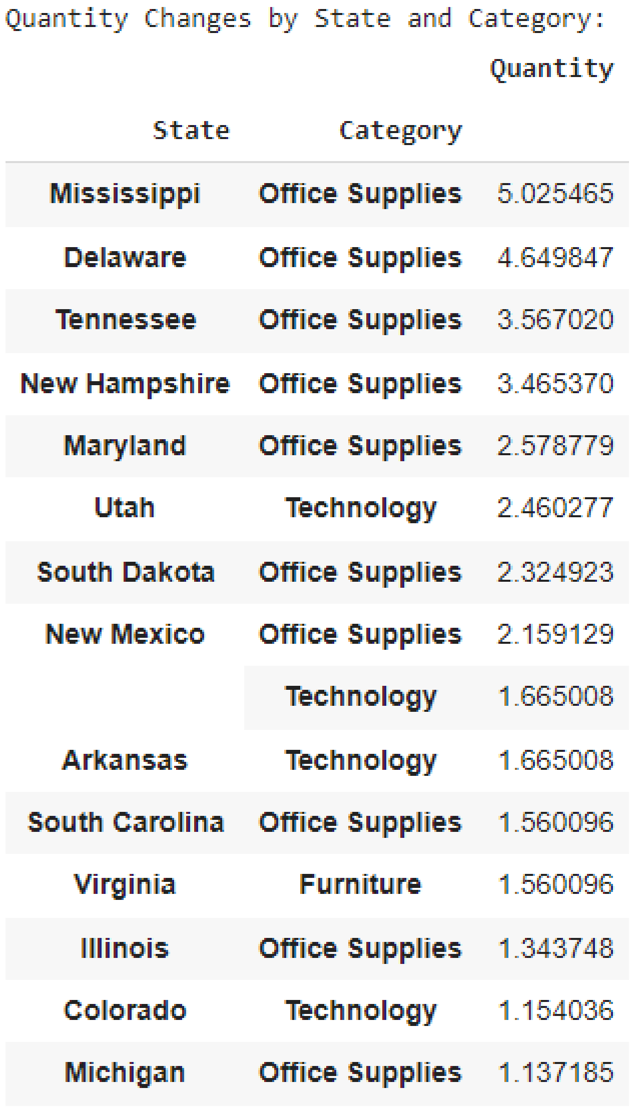


```
Category-wise Percentages of Total Quantity:
Category
Technology         -95.986117
Furniture         -152.832855
Office Supplies   -153.522124
Name: Quantity, dtype: float64
```



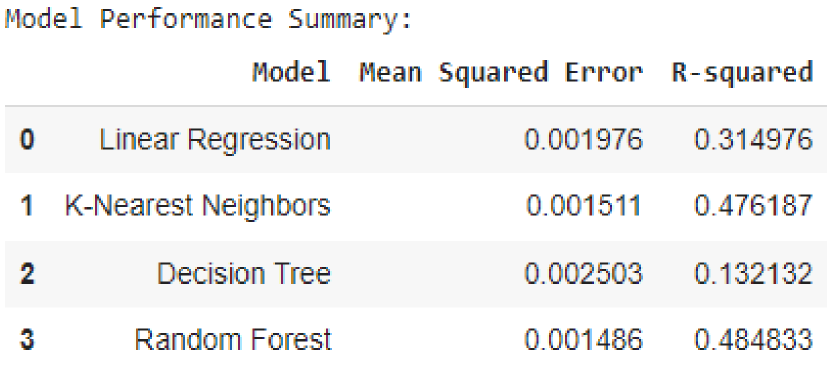
```
Monthly Trends in Sales and Quantity:
```

Month	Sales	Quantity
April	-36.324544	-30.426217
August	-37.444176	-13.728843
December	-87.026619	-57.542530
February	-17.925237	-22.977982
January	-28.144808	-11.038629
July	-39.626968	-20.645993
June	-40.730007	-29.918506
March	-42.288926	-24.067332
May	-42.681874	-40.405938
November	-78.153491	-40.165128
October	-55.458054	-55.082797
September	-87.821389	-56.341202



Quantity Changes by State and Category:

State	Category	Quantity
Mississippi	Office Supplies	5.025465
Delaware	Office Supplies	4.649847
Tennessee	Office Supplies	3.567020
New Hampshire	Office Supplies	3.465370
Maryland	Office Supplies	2.578779
Utah	Technology	2.460277
South Dakota	Office Supplies	2.324923
New Mexico	Office Supplies	2.159129
	Technology	1.665008
Arkansas	Technology	1.665008
South Carolina	Office Supplies	1.560096
Virginia	Furniture	1.560096
Illinois	Office Supplies	1.343748
Colorado	Technology	1.154036
Michigan	Office Supplies	1.137185



Model Performance Summary:

	Model	Mean Squared Error	R-squared
0	Linear Regression	0.001976	0.314976
1	K-Nearest Neighbors	0.001511	0.476187
2	Decision Tree	0.002503	0.132132
3	Random Forest	0.001486	0.484833

CONCLUSION

The project presents a comprehensive approach to analyzing and modeling data from a U.S. e-commerce dataset. Through exploratory data analysis, we gain valuable insights into patterns and relationships within the data, such as seasonal trends in sales and quantities, customer purchasing behaviors, and regional preferences. This analysis guides the development of predictive models, including linear regression and other machine learning techniques, to make informed predictions and improve business operations. By evaluating model performance using metrics such as accuracy, precision, and recall, we identify the most effective models for deployment. The feedback loop ensures continuous improvement of the models and data quality. Overall, the project's systematic approach to data management, analysis, and model development provides actionable insights that support strategic decision-making and enhance customer experiences in the e-commerce industry.

FUTURE SCOPE

The future scope of this project lies in expanding its capabilities to accommodate evolving data sources and business needs. By integrating real-time data streams, the system can provide up-to-date insights and predictive analytics to support dynamic decision-making. Enhancing the model's ability to handle larger datasets and more complex features will improve its predictive accuracy and robustness. Additionally, incorporating advanced machine learning algorithms such as deep learning and ensemble methods can lead to better performance and generalization across various use cases. As the e-commerce landscape continues to evolve, incorporating customer feedback and market trends into the model can help anticipate future demands and preferences. Lastly, expanding the user interface to offer more intuitive data visualizations and interactive features will facilitate broader adoption and enhance the overall user experience. By continuously refining the project and exploring new avenues for data integration and analysis, the system can stay at the forefront of e-commerce analytics and offer valuable insights for sustained growth and success.

REFERENCES

1. Project Github link (https://github.com/Loki-30G/E-Commerce.git), Lokesh G, 2024.
2. Project video recorded link (https://youtu.be/fA7Xy_GIN1U), Lokesh G, 2024.
3. Project PPT & Report github link (https://github.com/Loki-30G/E-Commerce/tree/main/Report%20and%20PPT), Lokesh G , 2024.
4. Project Dataset Github link (https://github.com/Loki-30G/E-Commerce/tree/main/Dataset), Lokesh G, 2024.

GIT Hub Link of Project Code:

https://github.com/Loki-30G/E-Commerce/tree/main/Code

THANK YOU