

Data Analyst Project

Analyzing Consumer Behavior in Online Retail: Insights from a UK E-Commerce Dataset

Presenter

Lalitha Shamugam

- [Google Scholar](#)
- [LinkedIn](#)

Task I: Topic and Data Set

Data Set Info

- The dataset chosen is "E-commerce Data".
- It is available in the following [link](#).

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	12/1/2010 8:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	12/1/2010 8:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MUG COSY	6	12/1/2010 8:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	12/1/2010 8:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	12/1/2010 8:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE ALPHABET BLOCKS	2	12/1/2010 8:34	9.95	13047	United Kingdom
536367	21754	HOME BUILDING BLOCK WORD	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	21755	LOVE BUILDING BLOCK WORD	3	12/1/2010 8:34	5.95	13047	United Kingdom
536367	21777	RECIPE BOX WITH METAL HEART	4	12/1/2010 8:34	7.95	13047	United Kingdom
536367	48187	DOORMAT NEW ENGLAND	4	12/1/2010 8:34	7.95	13047	United Kingdom
536368	22960	JAM MAKING SET WITH JARS	6	12/1/2010 8:34	4.25	13047	United Kingdom
536368	22913	RED COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom
536368	22912	YELLOW COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom
536368	22914	BLUE COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom
536369	21756	BATH BUILDING BLOCK WORD	3	12/1/2010 8:35	5.95	13047	United Kingdom

Data Set Description

1. This dataset contains actual transaction data from a UK-based online retail store.
2. It contains transaction data from November 2010 to December 2011.
3. There are approximately 500,000 records.

Data Set Attributes

- InvoiceNo: Invoice number (a unique identifier for each transaction)
- StockCode: Product code
- Description: Product description
- Quantity: Quantity of product purchased
- InvoiceDate: Date and time of purchase
- UnitPrice: Product price per unit
- CustomerID: Unique customer identifier
- Country: Country from where the order was placed

Read File

```
# import library  
import pandas as pd  
  
# read file  
data = pd.read_csv('data.csv', encoding = "ISO-8859-1")  
  
# view file  
data.head()
```

Basic Info

```
# find data info  
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 541909 entries, 0 to 541908  
Data columns (total 8 columns):  
#      Column      Non-Null Count  Dtype  
---  -  
0     InvoiceNo      541909 non-null  object  
1     StockCode      541909 non-null  object  
2     Description    540455 non-null  object  
3     Quantity      541909 non-null  int64  
4     InvoiceDate    541909 non-null  object  
5     UnitPrice      541909 non-null  float64  
6     CustomerID     406829 non-null  float64  
7     Country        541909 non-null  object  
dtypes: float64(2), int64(1), object(5)  
memory usage: 33.1+ MB
```


Statistical Description

```
# statistical description  
data.describe(include='all')
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
count	541909	541909	540455	541909.000000	541909	541909.000000	406829.000000	541909
unique	25900	4070	4223	NaN	23260	NaN	NaN	38
top	573585	85123A	WHITE HANGING HEART T- LIGHT HOLDER	NaN	10/31/2011 14:41	NaN	NaN	United Kingdom
freq	1114	2313	2369	NaN	1114	NaN	NaN	495478
mean	NaN	NaN	NaN	9.552250	NaN	4.611114	15287.690570	NaN
std	NaN	NaN	NaN	218.081158	NaN	96.759853	1713.600303	NaN
min	NaN	NaN	NaN	-80995.000000	NaN	-11062.060000	12346.000000	NaN
25%	NaN	NaN	NaN	1.000000	NaN	1.250000	13953.000000	NaN
50%	NaN	NaN	NaN	3.000000	NaN	2.080000	15152.000000	NaN
75%	NaN	NaN	NaN	10.000000	NaN	4.130000	16791.000000	NaN
max	NaN	NaN	NaN	80995.000000	NaN	38970.000000	18287.000000	NaN

Potential Business Hypothesis

Quantity and UnitPrice Relationship:

- Hypothesis: There is a relationship between the quantity of a product sold and its unit price
- Dependent Variable: Quantity
- Independent Variable: UnitPrice

Task II: Data Analysis & Prediction

Handling Missing Values

- No missing value found in Quantity and UnitPrice

```
# find missing value  
missing_values = data.isnull().sum()  
missing_values
```

```
InvoiceNo      0  
StockCode      0  
Description    1454  
Quantity       0  
InvoiceDate    0  
UnitPrice      0  
CustomerID    135080  
Country        0  
dtype: int64
```

Handling Outlier

To detect outliers in the "Quantity" and "UnitPrice" columns, we can use the Interquartile Range (IQR) method. This involves:

1. Calculating the first (Q1) and third quartiles (Q3) for each column.
2. Determining the IQR, which is the difference between Q3 and Q1.
3. Identifying outliers as values that fall below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$.

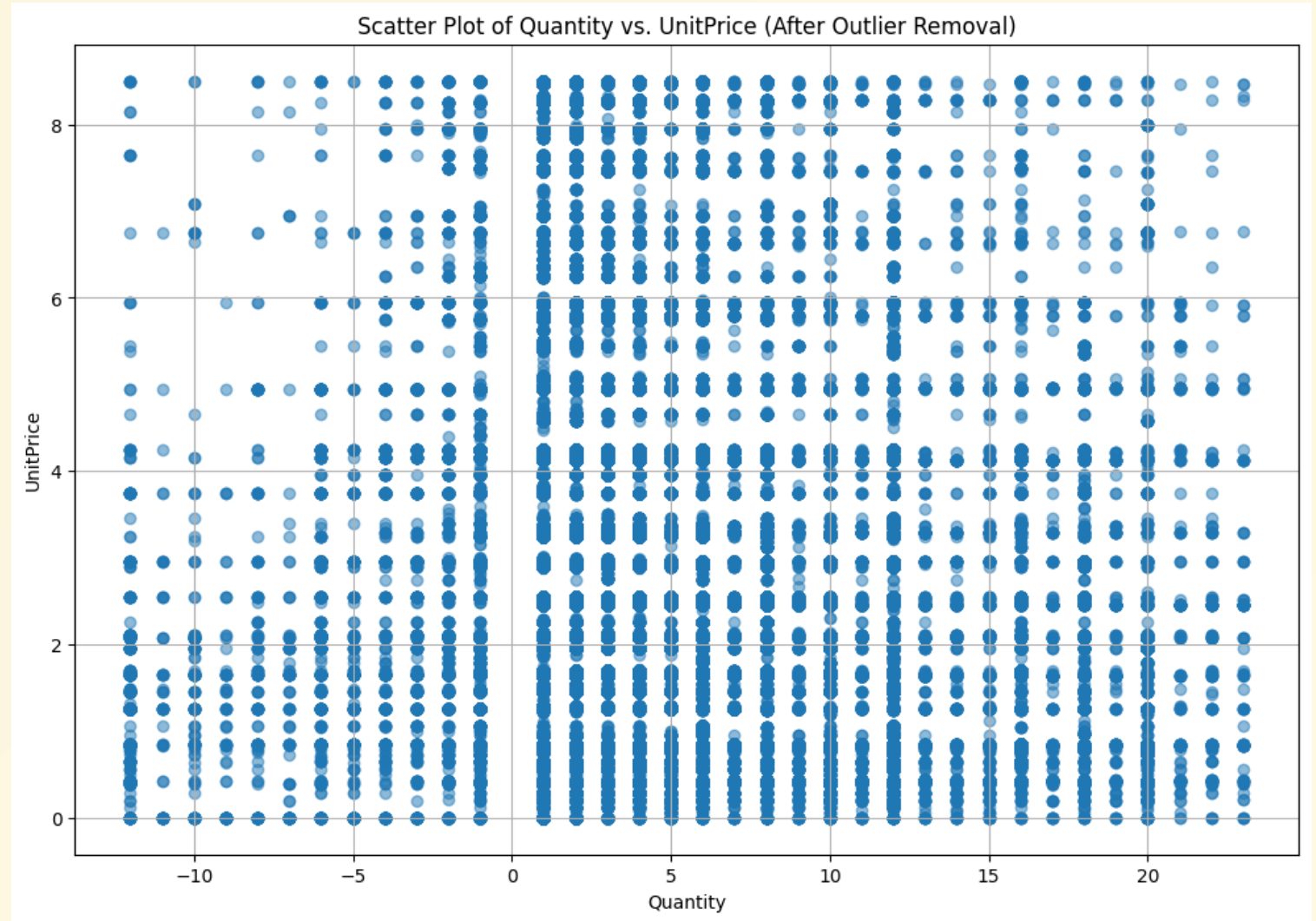
Handling Outlier

Based on the Interquartile Range (IQR) method:

- There are 58,619 outliers detected in the "Quantity" column.
- There are 39,627 outliers detected in the "UnitPrice" column.

Scatter Plot

- It is difficult to observe any pattern or linear relationship.
- Clearly this will have no or weak relationship.



Testing Relationship

```
# find Pearson correlation coefficient  
correlation_coefficient = data_cleaned["Quantity"].corr(data_cleaned["UnitPrice"])  
  
correlation_coefficient
```

- The Pearson correlation coefficient between "Quantity" and "UnitPrice" in the cleaned dataset is approximately -0.2805
- This indicates a weak negative correlation between the two variables. As the quantity increases, the unit price tends to decrease slightly (and vice versa), but the relationship is not very strong.

Updated Business Hypothesis

Quantity and TotalSales Relationship:

- Hypothesis: There is a relationship between the quantity of a product sold and its total sales
- Dependent Variable: Quantity
- Independent Variable: TotalSales

Feature Engineering: TotalSales

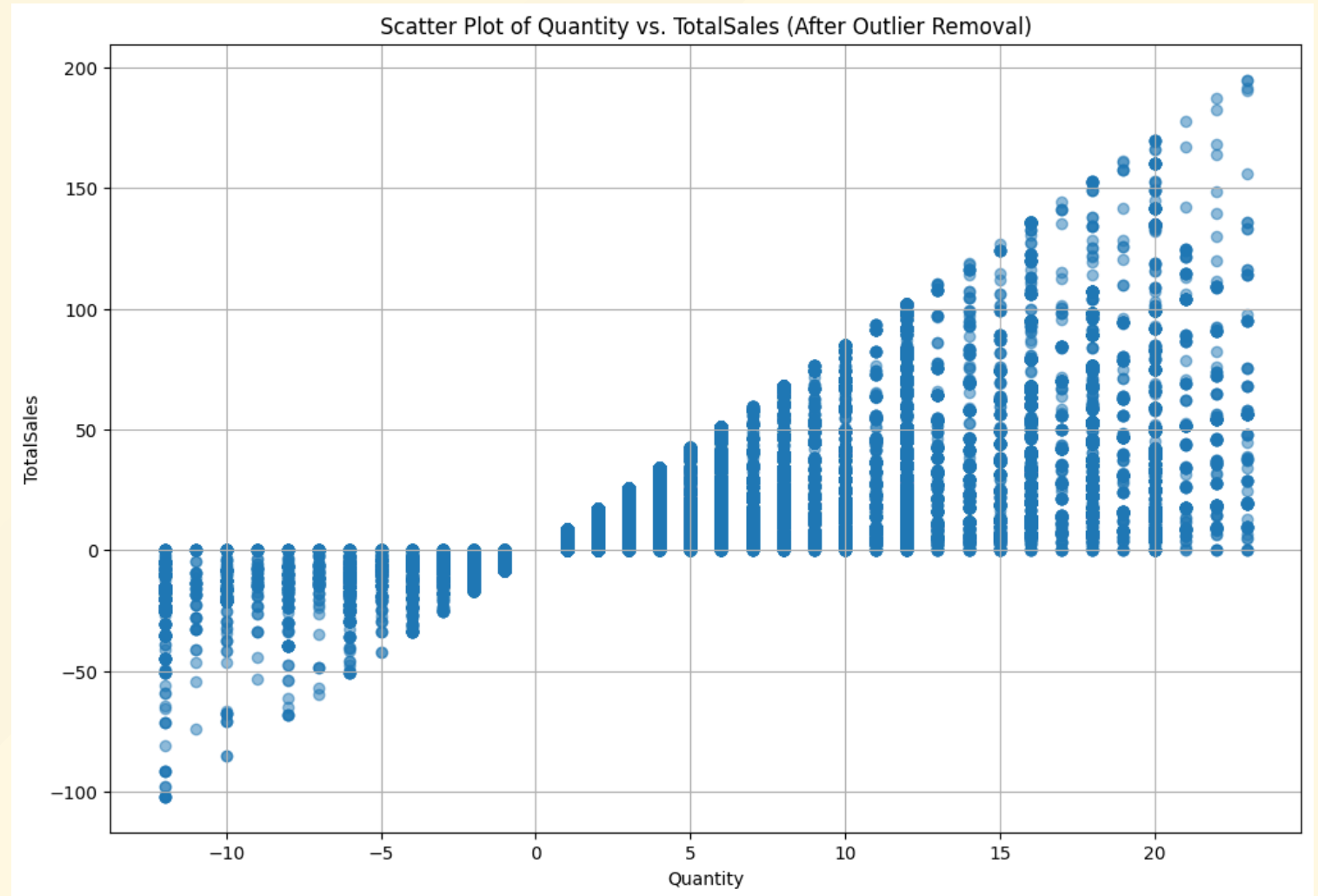
The new variable, "TotalSales", will be computed as:

$$TotalSales = Quantity \times UnitPrice$$

```
# Create the new "TotalSales" variable  
data_cleaned["TotalSales"] = data_cleaned["Quantity"] * data_cleaned["UnitPrice"]  
  
# Display the first few rows of the dataset with the new variable  
data_cleaned.head()
```

Scatter Plot

- Now, the linear relationship is much more visible.
- Clearly this will have moderate or strong positive linear relationship.



Testing Relationship

```
# Compute the Pearson correlation coefficient between "Quantity" and "TotalSales"  
correlation_total_sales_quantity = data_cleaned["TotalSales"].corr(data_cleaned["Quantity"])  
  
correlation_total_sales_quantity
```

- The Pearson correlation coefficient between "Quantity" and "TotalSales" in the cleaned dataset is approximately 0.6062.
- This indicates a moderate positive correlation between the two variables. As the quantity increases, the total sales also tend to increase, which is expected since "TotalSales" is derived from "Quantity" (among other factors).

Result

- There is moderate positive correlation between "Quantity" and "TotalSales"

New Cleaned Dataset

```
# View new cleaned data  
data_cleaned.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	TotalSales
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom	15.30
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom	20.34
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom	22.00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom	20.34
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom	20.34

Task III: Results Visualisation

New Cleaned Dataset

- We will take the cleaned data and generate few visualisation using Tableau

Bar Chart

