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

Unit Price and Quantity Relationship:

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

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×IQR or above Q3+1.5×IQR.

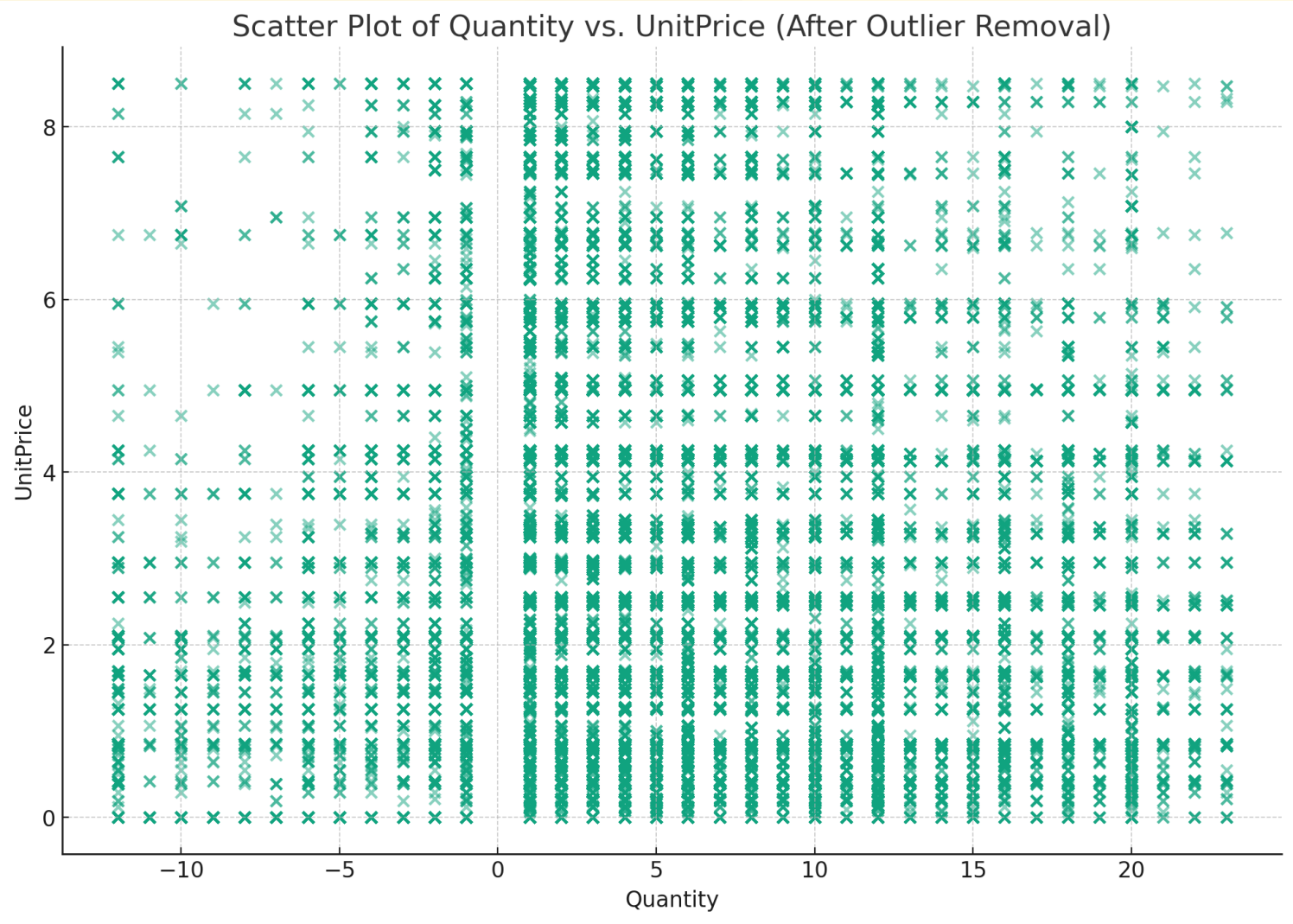
Handling Outlier

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 coeeficient
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

Total Sales and Quantity Relationship:

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

Part I: Preprocessing

Python Library

Pandas for Data Analysis and Manipulation

A Python library is a collection of related modules. It makes Python programming simpler and convenient for the programmer.

```
# importing Pandas library
import pandas as pd
```

Pandas is the backbone of most python data mining projects.

Reading Data

Usually we can use pandas library. Pandas store the imported data as DataFrame.

```
# Default sep = ','
df = pd.read_csv("iris_dirty.csv")
```

or if you want to use another separator, simply add sep='\t'

```
df = pd.read_csv("file_name.csv", sep = '\t')
```

Iris Flower Dataset

- Also known as Fisher's Iris dataset
- Introduced by Ronald Fisher in his 1936 paper.

View Data

You can have a look at the first five rows with .head():

```
# by default is 5 rows
df.head()
# you can also customize the #-rows
df.head(10)
```

or the last five rows with .tail():

```
df.tail()
```

Data Info

The shape property returns the dimensionality of the DataFrame.

```
df.shape
```

The info() method prints information about the DataFrame.

```
df.info
```

Statistical Description

All standard statistical operations are present in Pandas:

```
# Show the statistical summary on the numerical columns
df.describe()
# or individually
df.mean()
```

```
# Show the statistical summary on the categorical columns
df.describe(include = 'object')
```

Data Cleaning

Finding Missing Values

It is common to have not-a-number (NaN) values in your data set.

```
# Will give the total number of NaN in each column
df.isna().sum()
```

Data Cleaning

Handling Missing Values

```
# Remove the rows with NaN, not recommended
df.dropna()
```

```
# fill NaN with 0, also not recommended
df.fillna(0)

# fill NaN with mean, better
df1 = df.fillna(df.mean(numeric_only=True))
```

Data Cleaning

Problematic Values

Typo can be considered problematic.

```
# Count unique categorical values
df1.Species.unique()
df1['Species'].value_counts()

# View problematic values
df1.iloc[[7]]
```

Data Cleaning

Handling Problematic Values

We can replace with the correct value using replace()

```
df2 = df1.replace(['SETSA'],'setosa')
```

Cleaning done!

Check out my Kaggle post for more data cleaning example.

Data Visualization

Why?

- Visualizing data prior to analysis is a good practice.
- Statistical description do not fully depict the data set in its entirety.

Check out my video HERE explaining the importance of visualizing data when analyzing it.

Cheat Sheet

Data Visualization

Scatter plot

```
df2.plot.scatter(x = 'Petal.Length', y = 'Petal.Width')
```

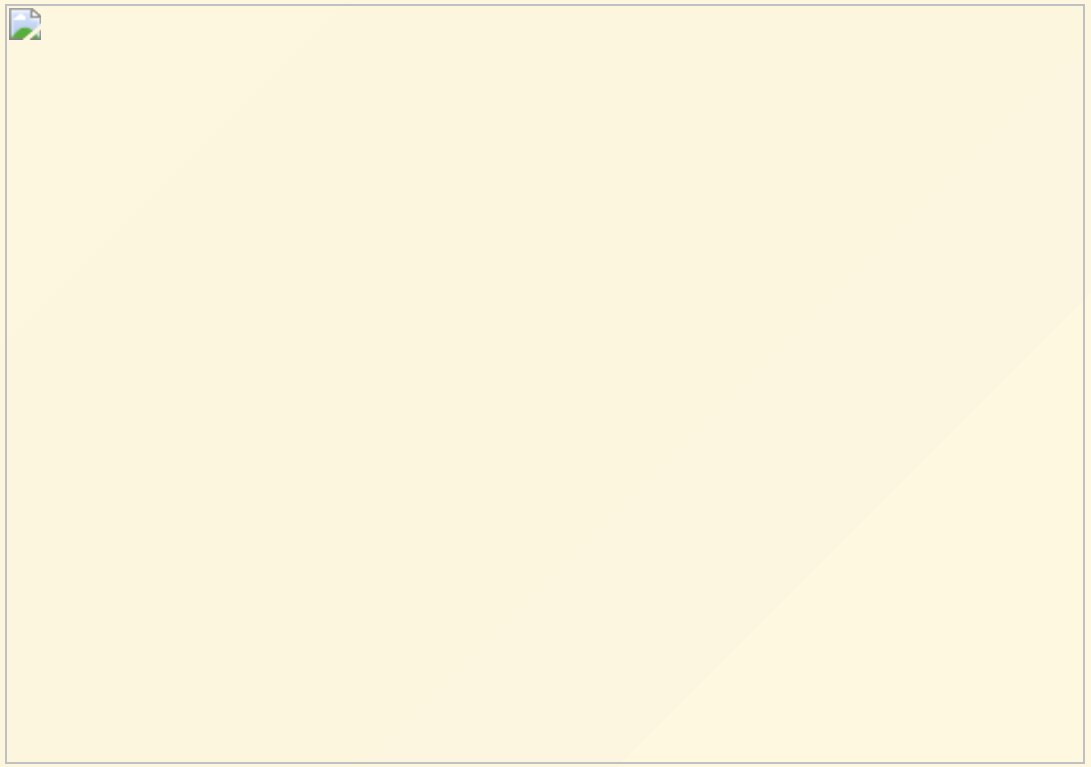
Using colour as third variable

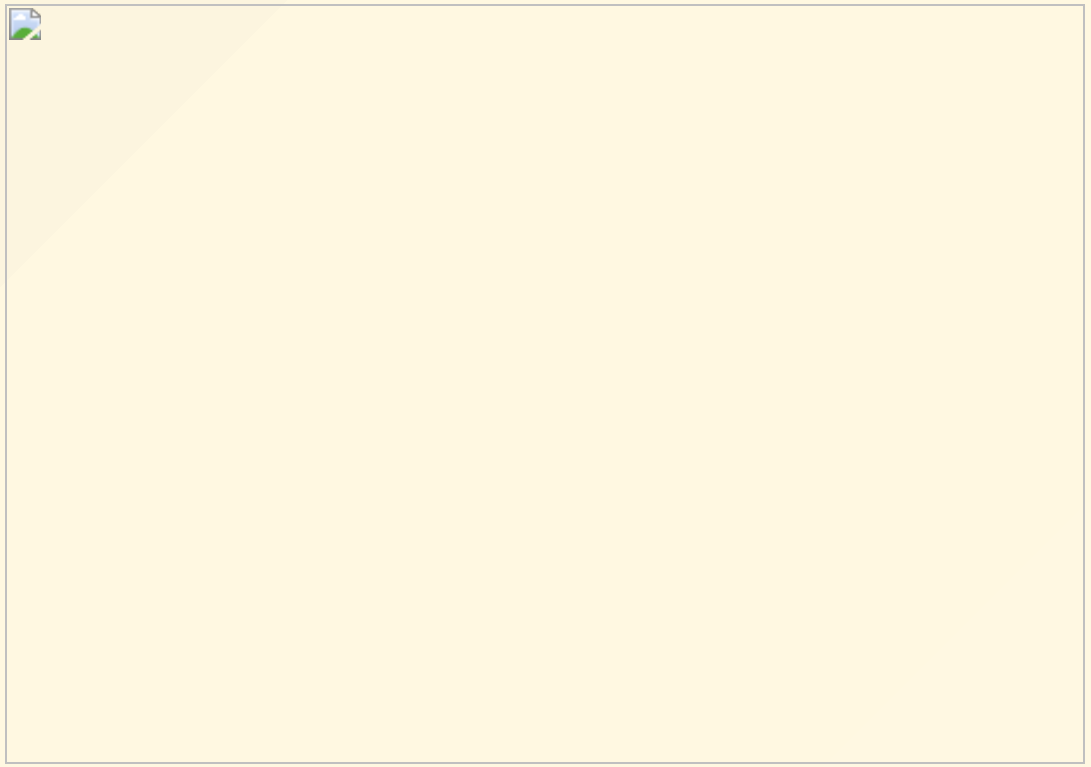
```
# Dictionary mapping colour with categorical values
colors = {'setosa':'red','virginica':'blue','versicolor':'green'}

df2.plot.scatter(x = 'Petal.Length', y = 'Petal.Width', c = df2['Species'].map(colors))
```

Data Visualization

What do you see?





Part II: Machine Learning (ML)

Machine Learning

Performing Classification

When you look at the petal measurements of the three species , what do you see?

- *It's pretty obvious to us humans that virginica has larger petals than versicolor and setosa. But machine cannot understand like we do. It needs some algorithm to do so.*

For that, we need to implement an algorithm that is able to classify the iris flowers into their corresponding classes.

Python Library

scikit-learn for Machine Learning

Scikit-learn provides various tools for model fitting, model selection, model evaluation, and many other utilities.

```
# import built-in machine learning algorithms, for example Logistic Regression

from sklearn.linear_model import LogisticRegression
```

Holdout Method

Randomly split the dataset into two sets; Training set and Test set

```
from sklearn.model_selection import train_test_split

# Separate/Assign the attributes into (X) and target (y)
X = df2.iloc[:, :-1]
y = df2.iloc[:, -1]

#split 80% training and 20 test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Algorithm

Logistic Regression is used to predict a categorical target , given a set of independent variables.

```
# import Linear Regression
from sklearn.linear_model import LogisticRegression

# create model
model = LogisticRegression(max_iter=150)

# train model
model.fit(X_train, y_train)
```

Evaluation

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance.

```
from sklearn import metrics
# Find accuracy
metrics.accuracy_score(y_test, y_pred)
```

```
#Find confusion matrix
metrics.confusion_matrix(y_test, y_pred)
```

Check out my video *HERE on how to calculate confusion matrix.

Finally!

Our model ready to be used.

```
# Lets create a new data
data = {'Sepal.Length': [4.7], 'Sepal.Width': [3.1], 'Petal.Length': [1.7], 'Petal.Width': [0.3]}

newdf = pd.DataFrame(data)
```

```
# Now we can predict using our model
ynew = model.predict(newdf)
```

Next?

Can we have a better classifier performance?

- Normalizing, scaling, feature selection, cross-validation, etc.

Which algorithm is better?

- Neural Network?, Check out my video on perceptron

How to apply?

- create a *"machine learning"* capable web/mobile-based system

Learning Materials

Textbook

- Introduction to Data Mining

Practical Books

- Python for Data Analysis
- Machine Learning with Python Cookbook

Exercise

Instructions

1. Download the dirty Iris dataset HERE
2. Perform data preprocessing such as handling missing values, handling problematic values, etc.
3. Split the dataset into training and test set
4. Perform classification using Logistic Regression
5. Evaluate the model