Katrina JUMADIAO

DATA ANALYTICS

Behind the Cart

A Deep Dive into Online Retail

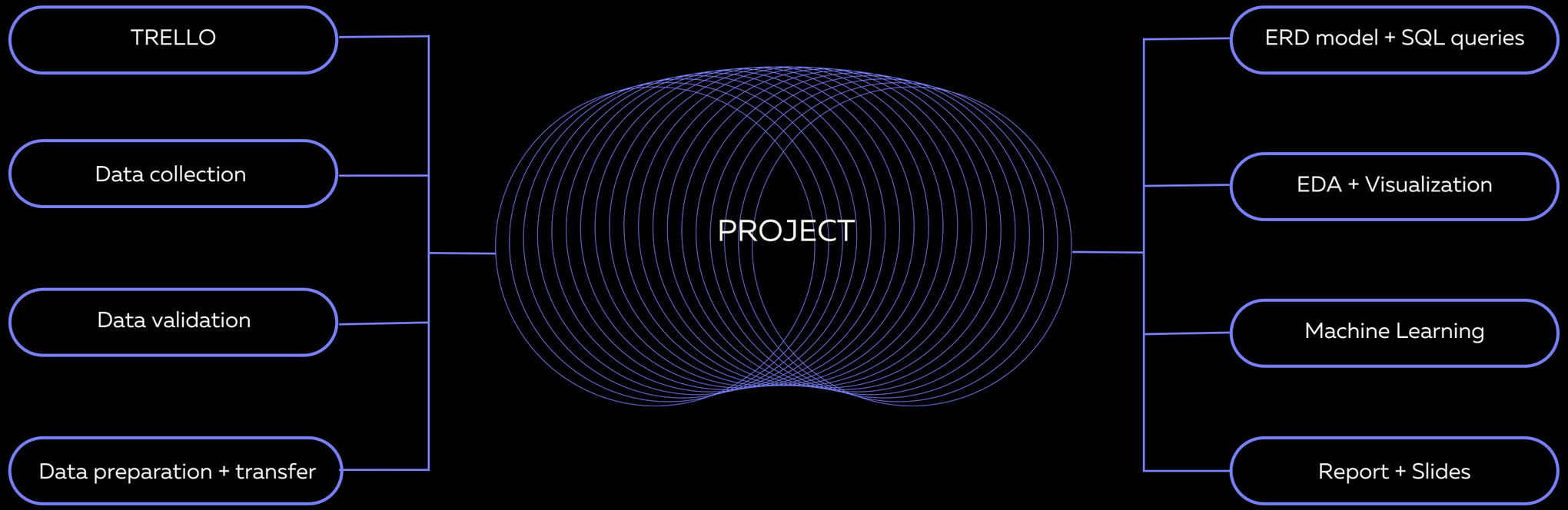
What's the project about?

DESCRIPTION.

This project involved analyzing sales data for a UK-based gift retailer. The objective was to uncover purchasing trends, identify customer segments, and inform data-driven business strategies.

PLAN

ARCHITECTING SUCCESS: THE BLUEPRINT FOR PROJECT COMPLETION



SOURCE

Online Retail Data Set

UCI Machine Learning Repository

Donated by Dr. Daqing Chen

Director of the Public Analytics group

Classification and clustering

PHASES

This project have undertaken a comprehensive journey through data analysis and machine learning in the context of sales and customer behavior in the retail industry.

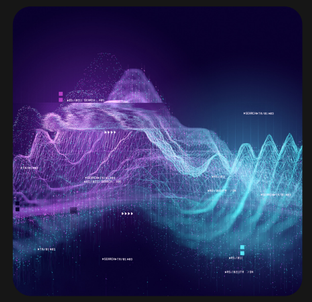
01
DATA VALIDATION

Reviewed and cleaned data to remove inconsistencies, duplicates, and handle missing values.

02
EDA AND VISUALS

Uncovered key patterns through analysis and visualized insights for a better understanding.

03
MACHINE LEARNING



Applied K-Means clustering to segment customers into high, mid, and low-value groups.

Data Cleaning

01 Duplicates

02 Missing values

03 Fix values

04 Data types

Data Cleaning

```
No. of values in the dataset : 4,335,272
Total rows in the dataset : 541,909
Total columns in the dataset : 8

Total null values : 136,534
Total duplicated rows : 5,268

RATIO OF MISSING AND DUPLICATED VALUES IN OUR DATA :
-----------------------------------------------------------------------------

Percentage of null values in the data : 3.15%
Percentage of duplicates in the data : 0.97%
```

01 Duplicates

02 Missing values

Data Cleaning

01 Duplicates

02 Missing values

```
No. of values in the dataset : 4,335,272
Total rows in the dataset : 541,909
Total columns in the dataset : 8

Total null values : 136,534
Total duplicated rows : 5,268

RATIO OF MISSING AND DUPLICATED VALUES IN OUR DATA :
-----------------------------------------------------------------------------------------

Percentage of null values in the data : 3.15%
Percentage of duplicates in the data : 0.97%
```

InvoiceNo	InvoiceDate	StockCode	Description	Quantity	UnitPrice	Revenue	CustomerID	Country
537051	2010-12-05 11:12:00	21916	retro white sticks chalk 12 set	1	0.42	0.42	15708	United Kingdom
537051	2010-12-05 11:12:00	21916	retro white sticks chalk 12 set	4	0.42	1.68	15708	United Kingdom
537051	2010-12-05 11:12:00	22725	alarm bakelike chocolate clock	1	3.75	3.75	15708	United Kingdom
537051	2010-12-05 11:12:00	22725	alarm bakelike chocolate clock	2	3.75	7.50	15708	United Kingdom

Data Cleaning

01 Duplicates

02 Missing values

03 Fix values

	StockCode	Description
0	16156L	2
1	17107D	3
2	20622	2
3	20725	2
4	20914	2
...	...	...
208	85184C	2
209	85185B	2
210	90014A	2
211	90014B	2
212	90014C	2

213 rows × 2 columns

		Revenue
StockCode	**Description**	
16156L	WRAP CAROUSEL	157.50
	WRAP, CAROUSEL	42.00
17107D	FLOWER FAIRY 5 DRAWER LINERS	150.45
	FLOWER FAIRY 5 SUMMER DRAW LINERS	15.30
	FLOWER FAIRY,5 SUMMER B'DRAW LINERS	267.75

Combine all the words in one.
For example, from:

```
22199   frying pan red polkadot
22199   frying pan red retrospot
```

To:

```
22129 frying pan red polkadot retrospot
```

Put necessary spaces.
For example, from:

```
20622 vippassport cover
```

To:

```
20622 vip passport cover
```

Put space before and after a digit.
For example, from:

```
17107D  flower fairy5 drawer liners
```

To:

```
17107D flower fairy 5 drawer liners
```

Delete unnecessary spaces before and after each sentence (only 1 space between words).
For example, from:

```
21175   gin tonic diet   metal sign
```

To:

```
21175 gin tonic diet metal sign
```

Data Cleaning

01 Duplicates

02 Missing values

03 Fix values

I made the following corrections for clarity:

- Renamed "EIRE" to "Ireland"
- Renamed "RSA" to "Republic of South Africa"

invoiceDate_norm	Year	YearMonth	Month	MonthDate	DayofMonth	Hour	DayofWeek	Month_name	Day_name
2010-12-01	2010	2010-12	12	12-01	1	8	2	December	Wednesday
2010-12-01	2010	2010-12	12	12-01	1	9	2	December	Wednesday
2010-12-01	2010	2010-12	12	12-01	1	9	2	December	Wednesday
2010-12-01	2010	2010-12	12	12-01	1	10	2	December	Wednesday
2010-12-01	2010	2010-12	12	12-01	1	10	2	December	Wednesday

Data Cleaning

01 Duplicates

02 Missing values

03 Fix values

Before

InvoiceNo	InvoiceDate	StockCode	Description	Quantity	UnitPrice	Revenue	CustomerID	Country
537051	2010-12-05 11:12:00	21916	retro white sticks chalk 12 set	1	0.42	0.42	15708	United Kingdom
537051	2010-12-05 11:12:00	21916	retro white sticks chalk 12 set	4	0.42	1.68	15708	United Kingdom
537051	2010-12-05 11:12:00	22725	alarm bakelike chocolate clock	1	3.75	3.75	15708	United Kingdom
537051	2010-12-05 11:12:00	22725	alarm bakelike chocolate clock	2	3.75	7.50	15708	United Kingdom

After

InvoiceNo	InvoiceDate	StockCode	Description	Quantity	UnitPrice	Revenue	CustomerID	Country
537051	2010-12-05 11:12:00	21916	retro white sticks chalk 12 set	5	0.42	2.10	15708	United Kingdom
537051	2010-12-05 11:12:00	22725	alarm bakelike chocolate clock	3	3.75	11.25	15708	United Kingdom

Data Cleaning

01 Duplicates

02 Missing values

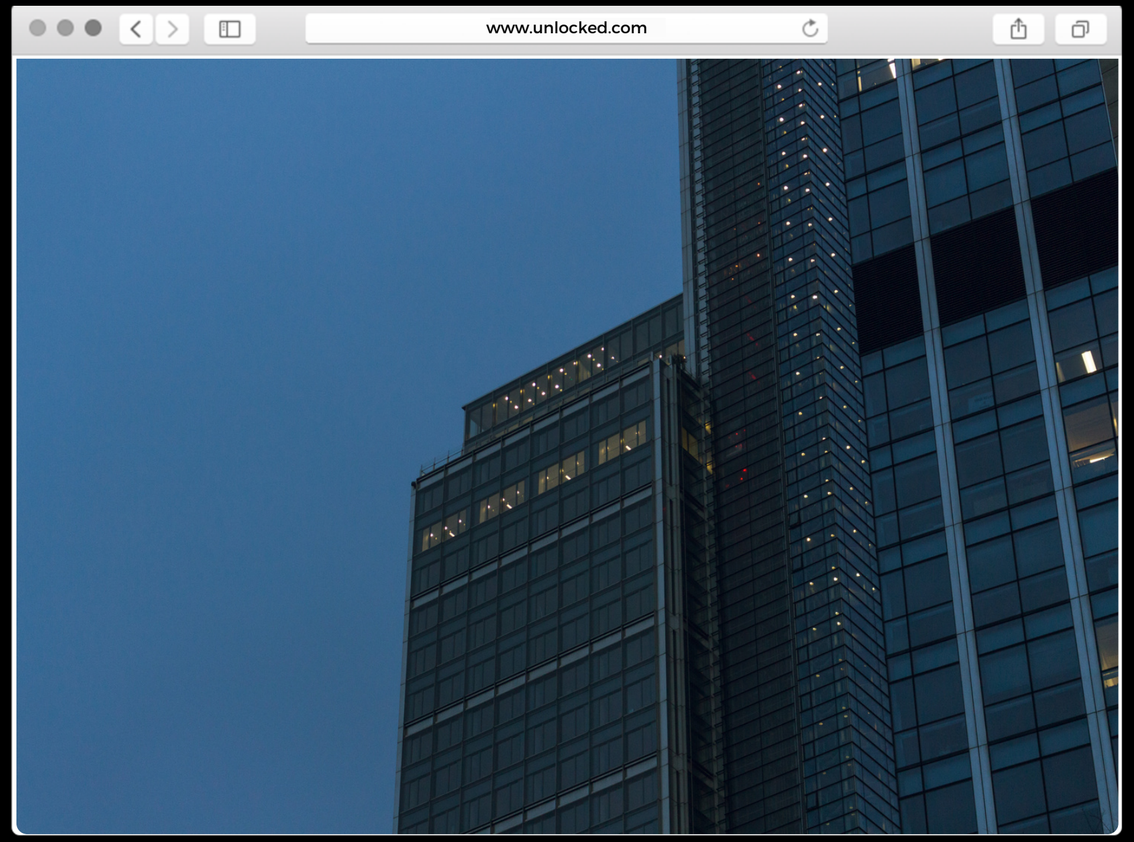
03 Fix values

04 Data types

```
InvoiceNo       object
InvoiceDate     object
StockCode       object
Description     object
Quantity         int64
UnitPrice      float64
CustomerID     float64
Country         object
dtype: object
```

There are few changes to do in our data:

- InvoiceNo to int64
- InvoiceDate to datetime[ns]
- Add Revenue column
- CustomerID to int64

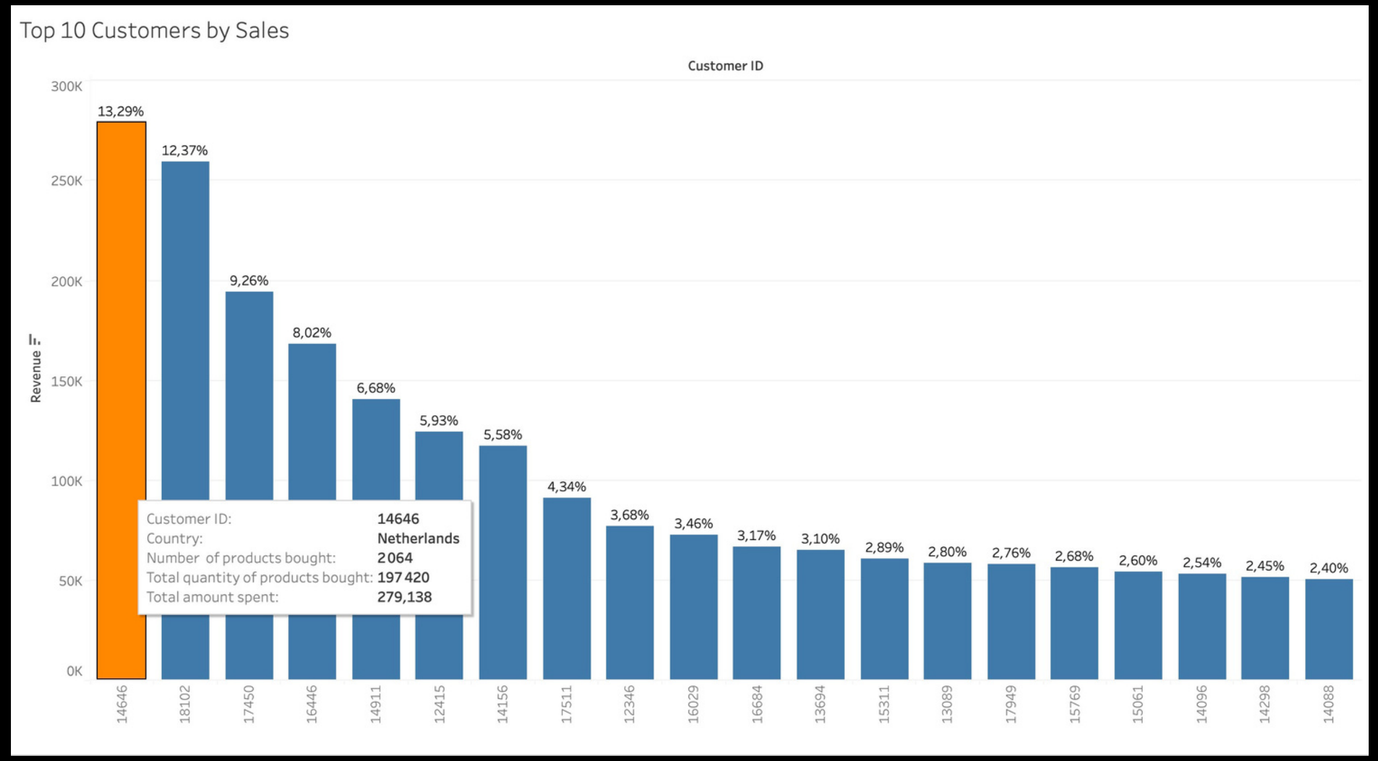


Illuminating Information Through Design

DATA VISUALIZATION: From Numbers to Narratives

TOP REVENUE CONTRIBUTING CUSTOMERS

Understanding which customers contribute the most to revenue
helps segment customers based on their value to the business



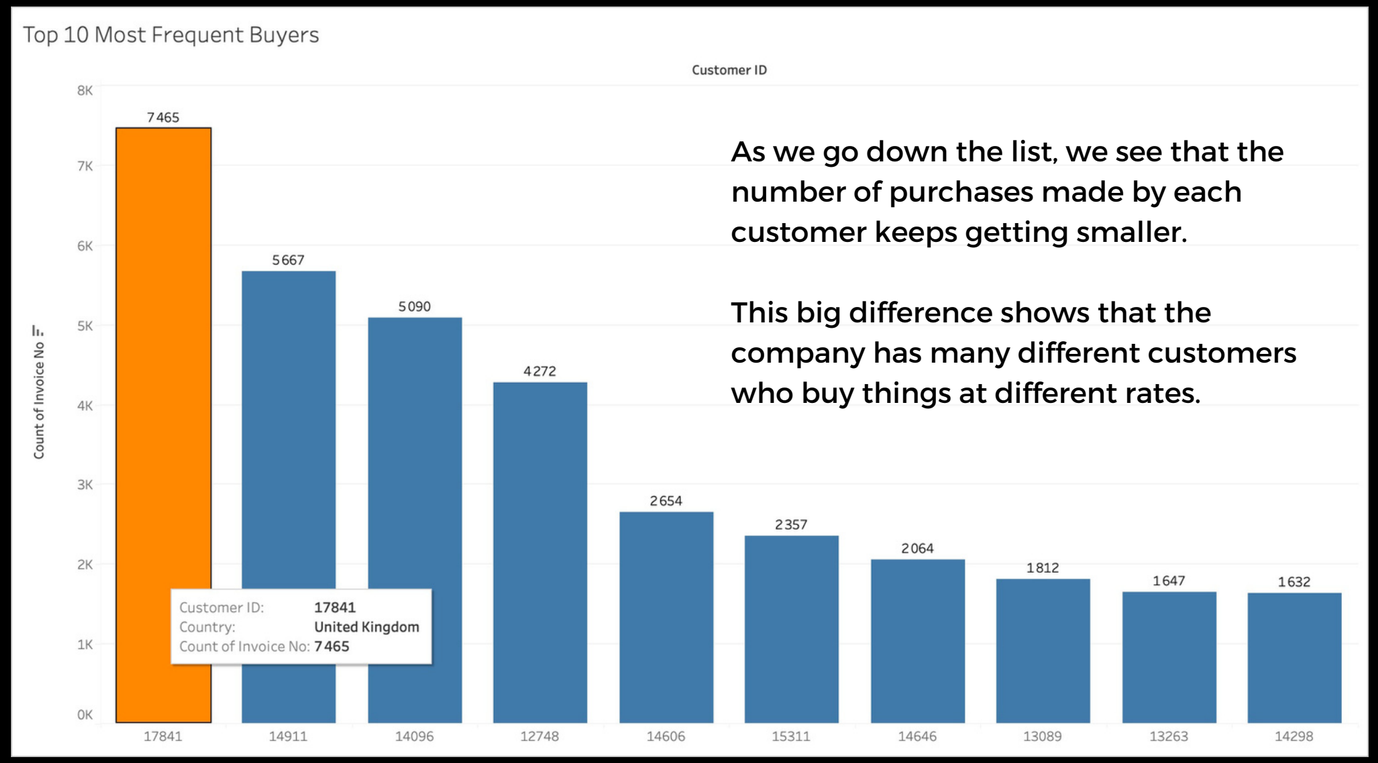
Majority of top revenue contributors are based in the UK, reflecting the strong customer base in this location.

Customer 14646 tops the list demonstrating the presence of high-value customers outside of the UK.

The revenue distribution among the top 20 customers varies from approximately 50K to 280K, which signifies the need for different customer engagement strategies based on their contribution.

MOST FREQUENT BUYERS

Frequency of purchase can indicate customer loyalty and engagement, useful for creating segments



The biggest buyer bought +7K items (a lot more than what other customers bought).

The next biggest buyer is about 76% of what the biggest customer bought. The big drop between the first and second buyers shows that the top customer buys an unusually high number of items.

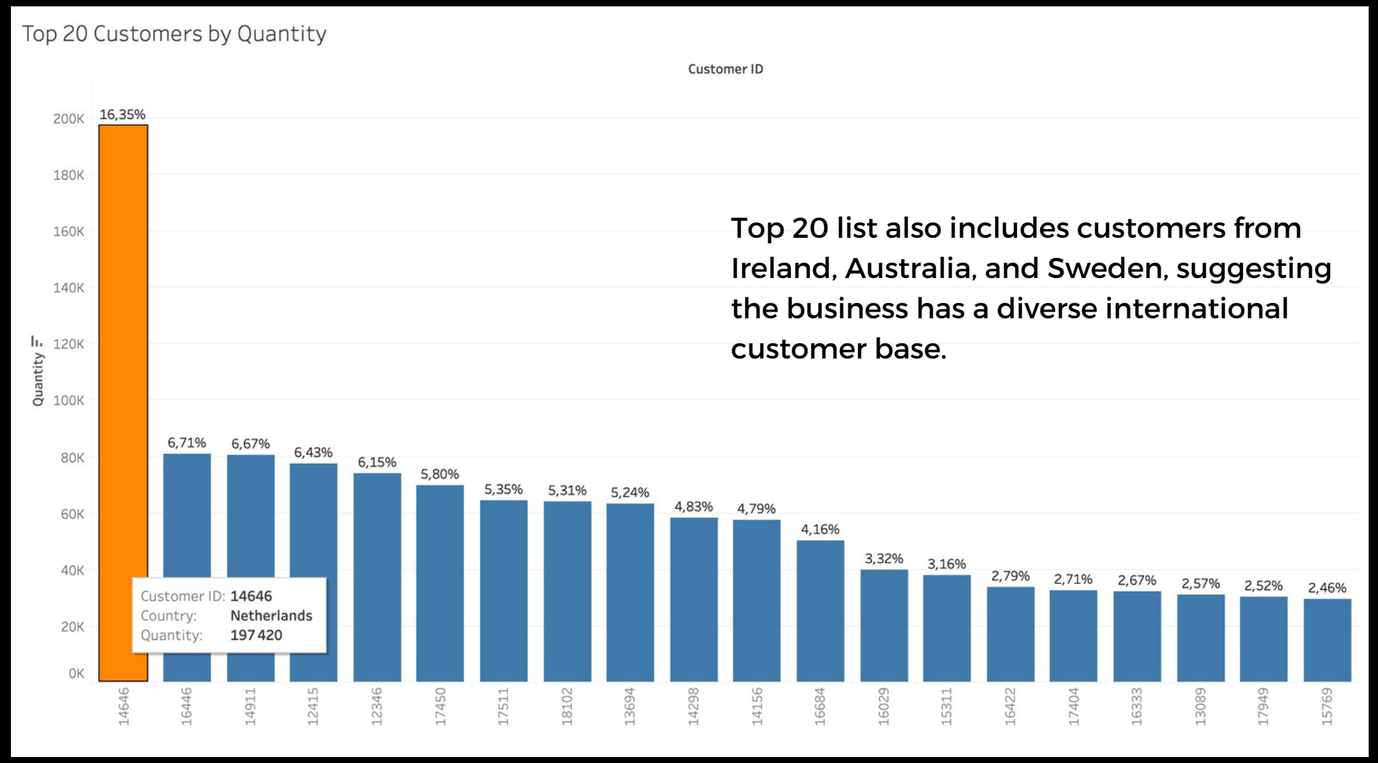
This study shows that customers who buy a lot are very important for keeping sales steady.

The company could start loyalty programs or special sales for these customers to keep them buying and maybe even get them to buy more.

It would also be good to find ways to get customers who don't buy as much to start buying more often, which would help increase total sales.

TOP CUSTOMERS BY QUANTITY

Quantity purchased helps identify high volume buyers, a potential segment



Customer 14646 from the Netherlands leads the pack with a total of +190K units purchased, exhibiting a tremendous buying power. This is significant considering the business's wholesaling nature.

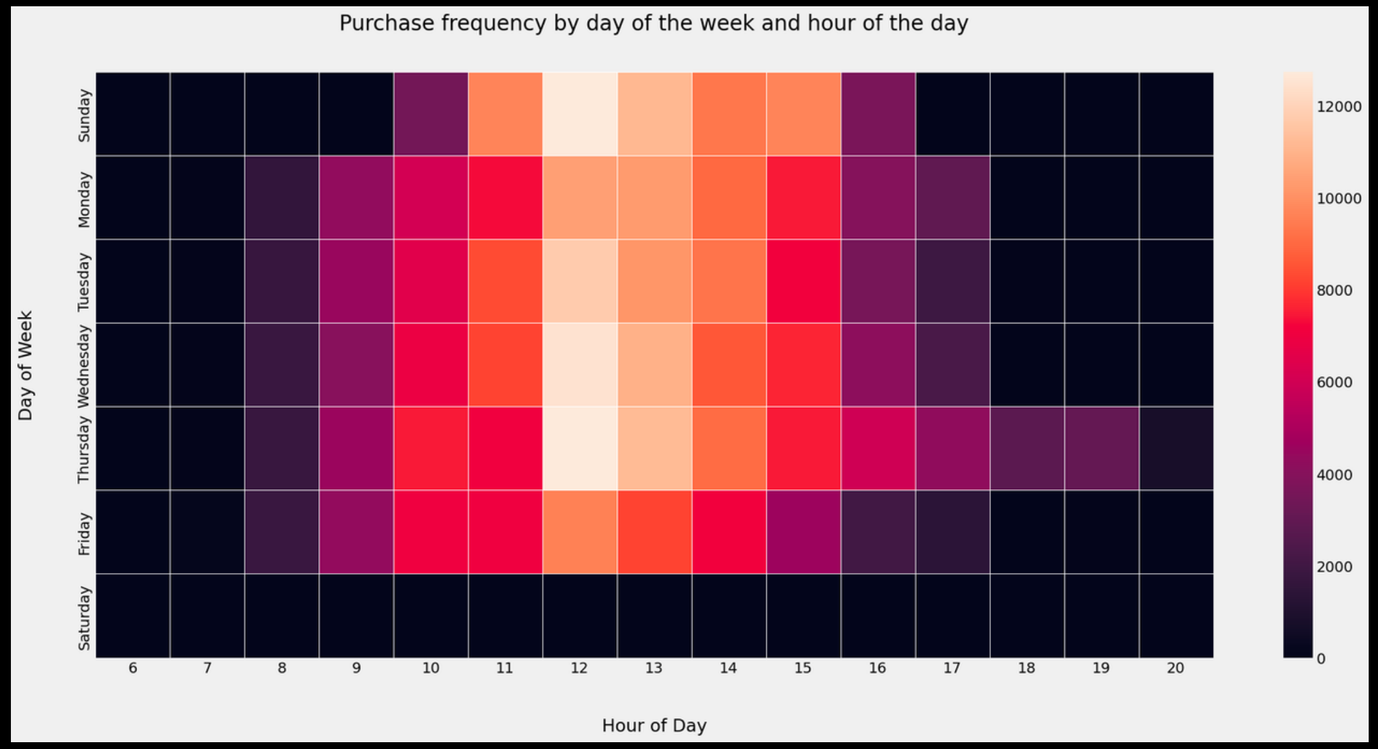
The UK seems to be a crucial market for the business, with a significant number of top buyers originating from this country. Among the top 20 customers, 14 are from the UK.

The second-highest quantity of purchases (+80K units) is attributed to customer 16446 from the UK.

However, this is less than half of the total quantity purchased by the top customer. This substantial difference underscores the unique buying power of customer 14646.

CUSTOMER PURCHASE TIMING

When do customers make most of their purchases?



There aren't many customers early in the morning or late at night. The business could consider this when deciding when to open and close and when to schedule staff.

Most customers buy things during regular work hours (10 am to 3 pm). This is when the business is busiest, so it's important to have enough customer service staff during these hours.

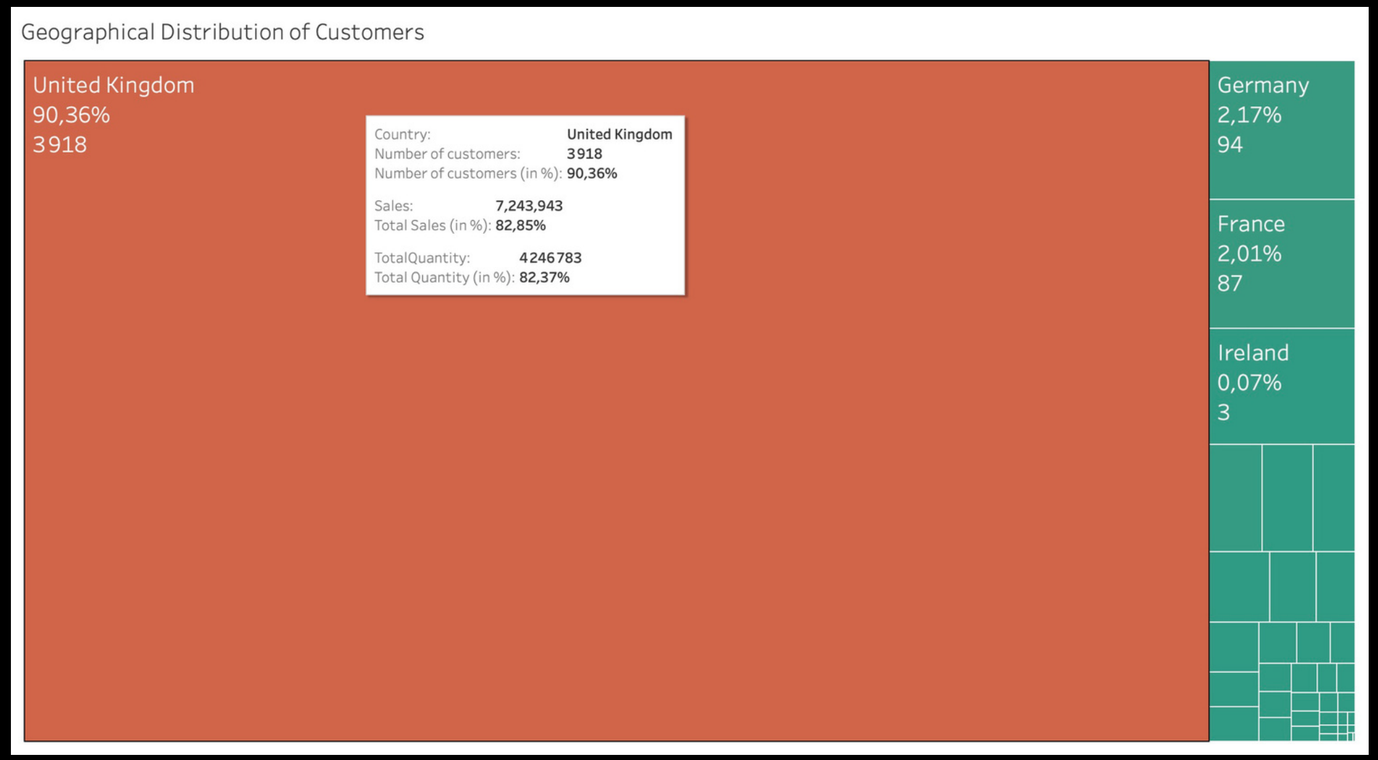
There's a sudden increase in shopping on Thursday evenings. This might mean customers are getting ready for the weekend. Because of this, Thursday might be a good day to advertise products or have special sales.

Shopping slows down quickly on Fridays, which might mean the weekend is usually slower.

Even though Sunday is a weekend day, its shopping pattern is like a weekday's.

GEOGRAPHICAL DISTRIBUTION OF CUSTOMERS

Understanding where customers are located can form the basis for geographical segmentation



Most of the company's customers are in the UK, with over 3K customers. This is not surprising given the company's location. There are also quite a few customers in Germany and France, with 94 and 87 customers respectively.

In other countries, the number of customers is much smaller, ranging from 30 customers in Spain to only one in several countries, including Singapore, Saudi Arabia, Brazil, and Iceland.

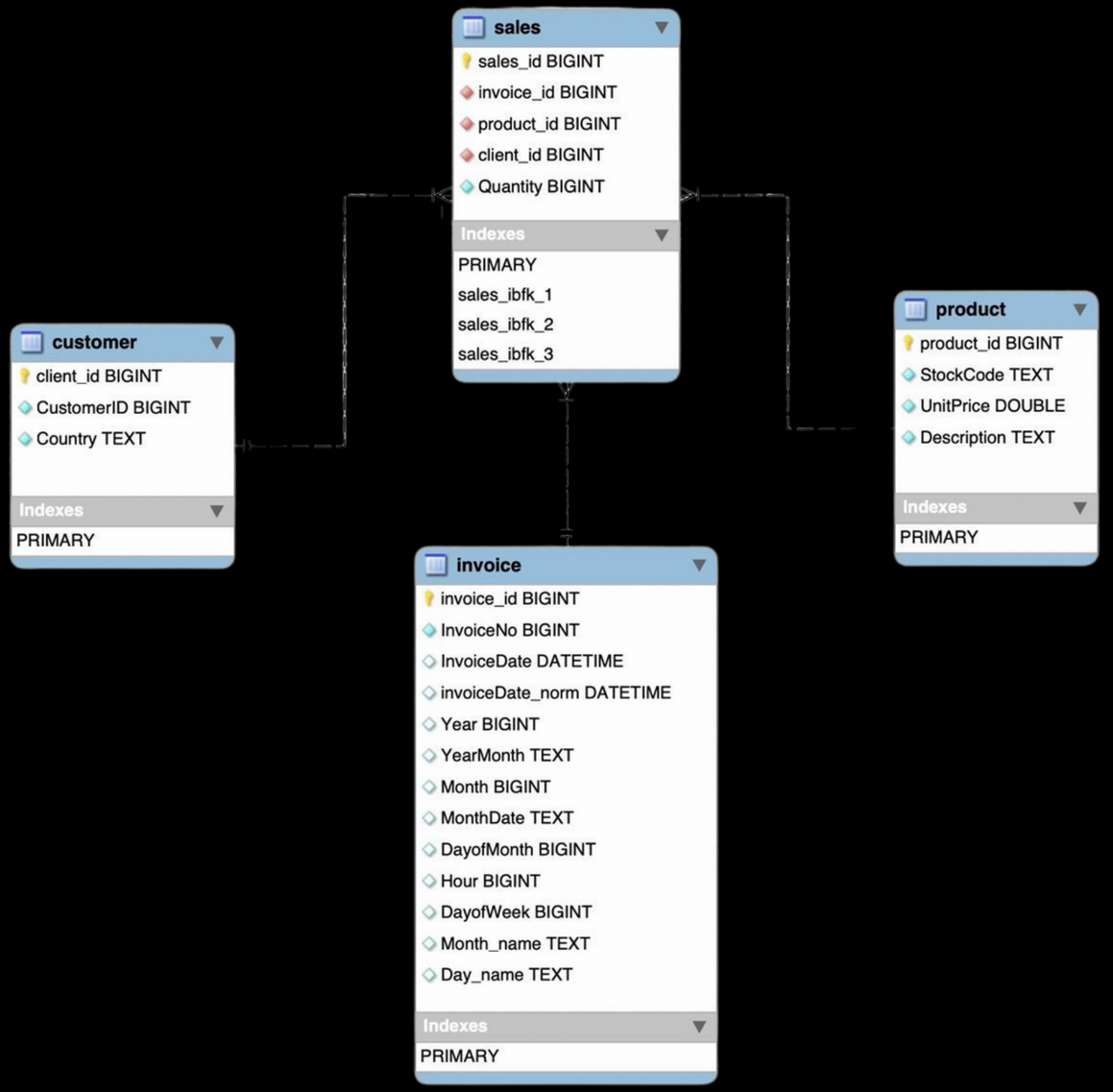
Even so, the fact that there are customers from all around the world, shows that the company's gift products have international appeal. It also means there's a chance for the company to grow in these international markets.

SQL

ER Diagram

Weaving the Web of Data:

Unfolding the Intricacies
of
ER Diagrams



sales_id	invoice_id	product_id	client_id	Quantity
0	0	0	0	6
1	0	1	0	2
2	0	2	0	6
3	0	3	0	6
4	0	4	0	6
5	0	5	0	8
6	0	6	0	6
7	1	7	0	6
8	1	8	0	6

main table

...

sales_id	invoice_id	product_id	client_id	Quantity
0	0	0	0	6
1	0	1	0	2
2	0	2	0	6
3	0	3	0	6
4	0	4	0	6
5	0	5	0	8
6	0	6	0	6
7	1	7	0	6
8	1	8	0	6

main table

...

invoice table

invoice_id	InvoiceNo	InvoiceDate	invoiceDate_norm	Year	YearMonth	Month	MonthDate	DayofMonth	Hour	DayofWeek	Month_name	Day_name
0	536365	2010-12-01 08:26:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
1	536366	2010-12-01 08:28:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
2	536367	2010-12-01 08:34:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
3	536368	2010-12-01 08:34:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
4	536369	2010-12-01 08:35:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
5	536370	2010-12-01 08:45:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
6	536371	2010-12-01 09:00:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday
7	536372	2010-12-01 09:01:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday
8	536373	2010-12-01 09:02:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday

sales_id	invoice_id	product_id	client_id	Quantity
0	0	0	0	6
1	0	1	0	2
2	0	2	0	6
3	0	3	0	6
4	0	4	0	6
5	0	5	0	8
6	0	6	0	6
7	1	7	0	6
8	1	8	0	6

main table

...

product table

product_id	StockCode	UnitPrice	Product
0	21730	4.25	holder star glass frosted tlight
1	22752	7.65	7 nesting babushka set boxes
2	71053	3.39	lantern white metal moroccan
3	84029E	3.39	red heart hottie white woolly
4	84029G	3.39	flag knitted hot union water bottle
5	84406B	2.75	hanger cupid cream coat hearts
6	85123A	2.55	holder heart cream hanging white tlight
7	22632	1.85	red warmer dot hand retrospot polka
8	22633	1.85	jack warmer union hand

invoice table

invoice_id	InvoiceNo	InvoiceDate	invoiceDate_norm	Year	YearMonth	Month	MonthDate	DayofMonth	Hour	DayofWeek	Month_name	Day_name
0	536365	2010-12-01 08:26:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
1	536366	2010-12-01 08:28:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
2	536367	2010-12-01 08:34:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
3	536368	2010-12-01 08:34:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
4	536369	2010-12-01 08:35:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
5	536370	2010-12-01 08:45:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
6	536371	2010-12-01 09:00:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday
7	536372	2010-12-01 09:01:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday
8	536373	2010-12-01 09:02:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday

sales_id	invoice_id	product_id	client_id	Quantity
0	0	0	0	6
1	0	1	0	2
2	0	2	0	6
3	0	3	0	6
4	0	4	0	6
5	0	5	0	8
6	0	6	0	6
7	1	7	0	6
8	1	8	0	6

main table

...

product table

product_id	StockCode	UnitPrice	Product
0	21730	4.25	holder star glass frosted tlight
1	22752	7.65	7 nesting babushka set boxes
2	71053	3.39	lantern white metal moroccan
3	84029E	3.39	red heart hottie white woolly
4	84029G	3.39	flag knitted hot union water bottle
5	84406B	2.75	hanger cupid cream coat hearts
6	85123A	2.55	holder heart cream hanging white tlight
7	22632	1.85	red warmer dot hand retrospot polka
8	22633	1.85	jack warmer union hand

client table

client_id	CustomerID	Country
0	17850	United Kingdom
1	13047	United Kingdom
2	12583	France
3	13748	United Kingdom
4	15100	United Kingdom
5	15291	United Kingdom
6	14688	United Kingdom
7	17809	United Kingdom
8	15311	United Kingdom

invoice table

invoice_id	InvoiceNo	InvoiceDate	invoiceDate_norm	Year	YearMonth	Month	MonthDate	DayofMonth	Hour	DayofWeek	Month_name	Day_name
0	536365	2010-12-01 08:26:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
1	536366	2010-12-01 08:28:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
2	536367	2010-12-01 08:34:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
3	536368	2010-12-01 08:34:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
4	536369	2010-12-01 08:35:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
5	536370	2010-12-01 08:45:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	8	2	December	Wednesday
6	536371	2010-12-01 09:00:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday
7	536372	2010-12-01 09:01:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday
8	536373	2010-12-01 09:02:00	2010-12-01 00:00:00	2010	2010-12	12	12-01	1	9	2	December	Wednesday

A LOOK BACK AT PROJECT MILESTONES

Highlights

01	Data cleaning	Null values addressed, irrelevant data excluded, data types fixed.
02	EDA + Visualization	Detected customer trends and patterns through comprehensive data visualizations.
03	ERD model	Developed an effective Entity-Relationship Diagram showcasing data structure.
04	Database schema	Constructed a robust database schema ensuring data integrity.
05	SQL queries	Performed SQL queries for extracting actionable business insights.

ML KEYWORDS

...

cluster

principal components

feature selection

kmeans

assumptions

algorithm

pca

evr

centroids

robustscaler

gridsearch

elbow

standardscaler

fit_transform

silhouette

dbscan

PREDICT

wonderful

preprocessing

feature engineering

CLUSTERING

Clustering is like organizing a big, mixed-up box of Lego blocks into groups by color, size, or shape. In our project, we used clustering to sort our customers into different groups based on their shopping behavior.

I chose this method because it helps us understand our customers better, so we can give them what they need and keep them happy!

CUSTOMER SEGMENTATION

The ABCs of AI

01 Feature Engineering

02 Preprocessing

03 Choosing number of k clusters

04 PCA

05 K-means

06 Update original data with new features

01 Feature engineering

```
snapshot_date = data["InvoiceDate"].max() + dt.timedelta(days=1)
data_rfm = data.groupby(["CustomerID"]).agg({
    "InvoiceDate": lambda x: (snapshot_date - x.max()).days,
    "InvoiceNo": "count",
    "Revenue": "sum"})

data_rfm.head()
```

✓ 0.2s

CustomerID	InvoiceDate	InvoiceNo	Revenue
12346	326	1	77183.60
12347	2	182	4310.00
12348	75	27	1437.24
12349	19	72	1457.55
12350	310	16	294.40

- Group customers by their ID

- Get the maximum date in the InvoiceDate column
- Add one day to calculate the next day after the latest invoice date
- Get the difference between client's date of purchase and the latest date

- Count number of purchases per customer

- Sum the total amount spent per customer

```
data_rfm.rename(columns={"InvoiceDate": "Recency",
                         "InvoiceNo": "Frequency",
                         "Revenue": "MonetaryValue"}, inplace=True)

data_rfm.head()
```

✓ 0.0s

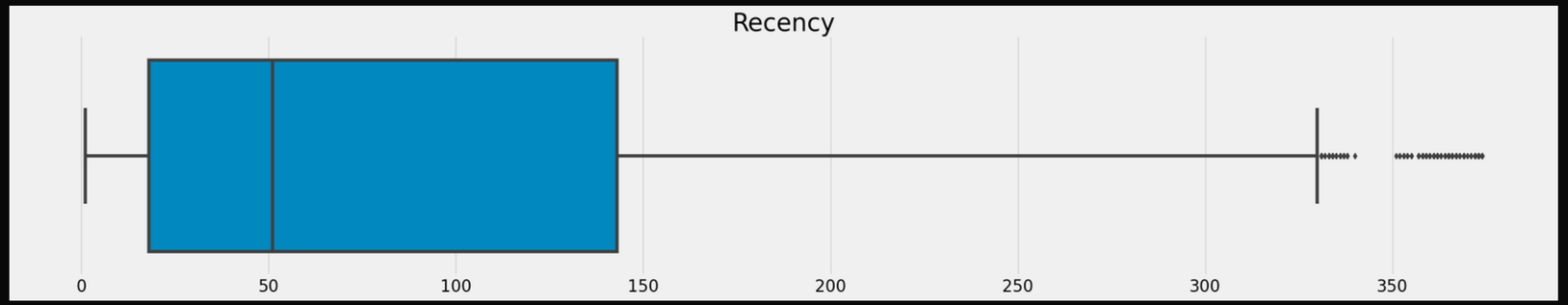
CustomerID	Recency	Frequency	MonetaryValue
12346	326	1	77183.60
12347	2	182	4310.00
12348	75	27	1437.24
12349	19	72	1457.55
12350	310	16	294.40

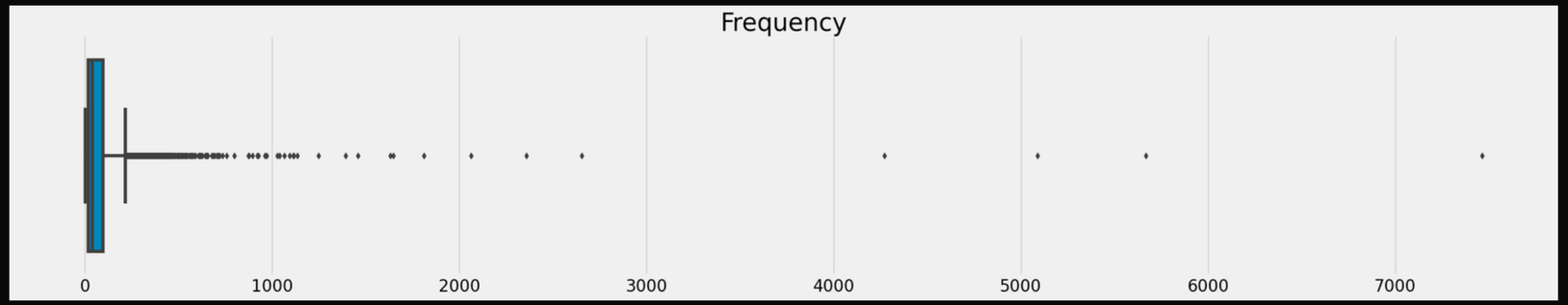
- **RECENCY**: This tells us when a customer last shopped with us

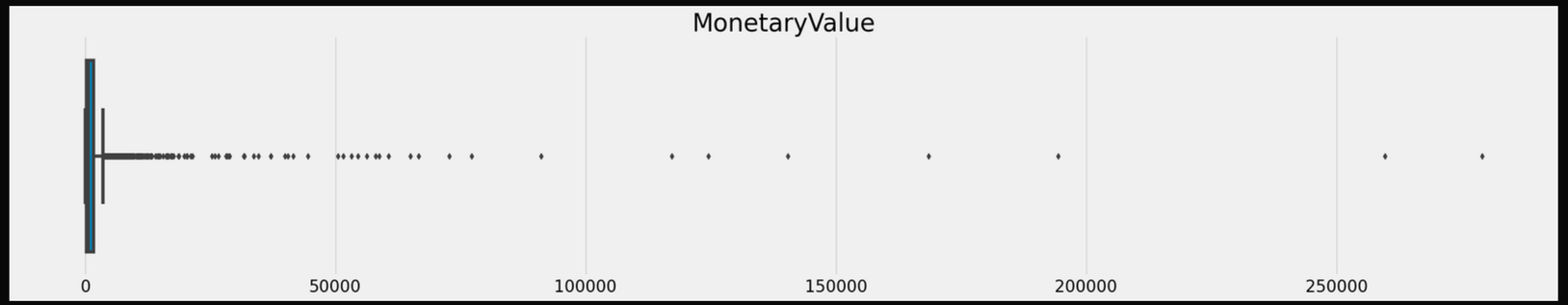
- **FREQUENCY**: This shows how often a customer shops with us

- **MONETARY VALUE**: This is how much a customer spends in our store

02 Preprocessing







02 Preprocessing

```
from sklearn.preprocessing import RobustScaler

feats = ["Recency", "Frequency", "MonetaryValue"]

scaler_rs = RobustScaler()
scaledRFM_rs = scaler_rs.fit_transform(data_rfm)

data_scaledRFM_rs = pd.DataFrame(scaledRFM_rs, columns=feats)
data_scaledRFM_rs.head()
```

✓ 0.0s

	Recency	Frequency	MonetaryValue
0	2.20	-0.49	57.66
1	-0.39	1.77	2.75
2	0.19	-0.16	0.58
3	-0.26	0.40	0.60
4	2.07	-0.30	-0.28

RobustScaler() was chosen because it's really good at dealing with data that has a lot of unusual or extreme values, which we call "outliers".

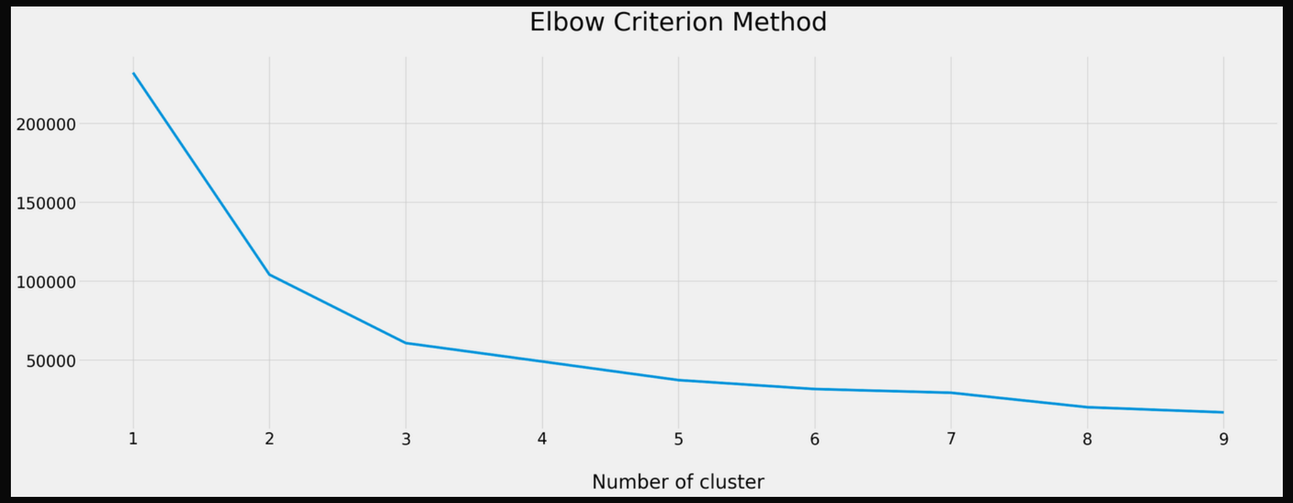
When we're getting our data ready for machine learning, we want all our measurements to be on a similar scale so that one doesn't overpower the others. But if we have outliers, they can really mess up this process.

Imagine you and your friends are ordering a pizza and everyone else wants just one slice, but one friend wants to eat the whole thing! It's not fair, right? That's what outliers can do to our data.

RobustScaler() doesn't let those outliers ruin the party. It scales the data using the median and quartiles, not the mean and standard deviation like many other scalers. This makes it robust to outliers!

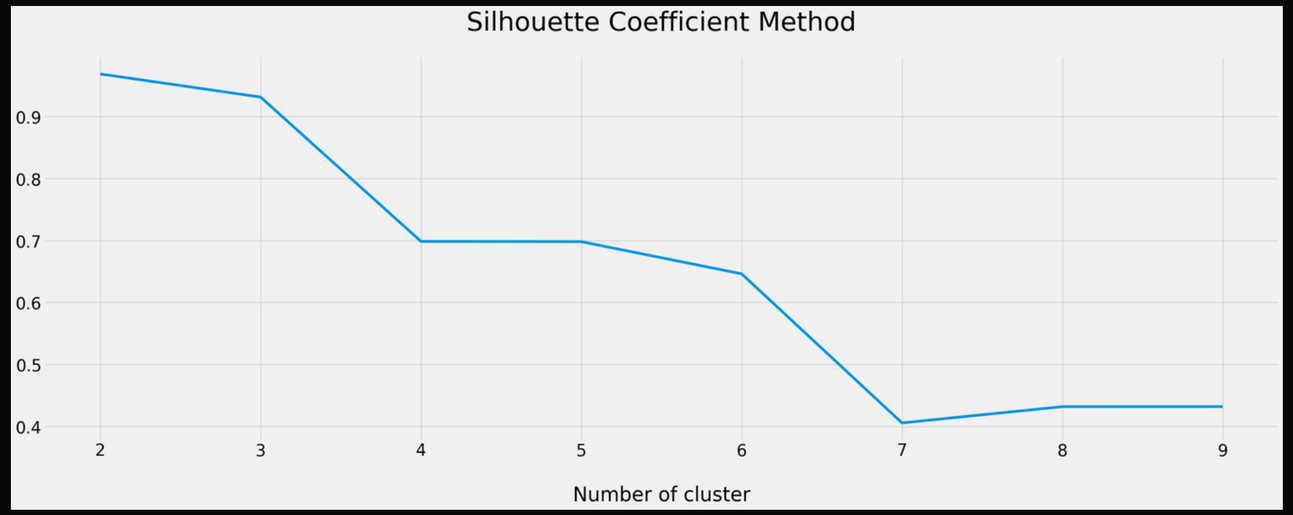
So, it helps us get the most accurate and fair results from our machine learning.

03 Choosing number of k clusters



We should look for a kink or bend in the plot - this is the "elbow." This bend often indicates a rate of diminishing returns where an increase in clusters does not result in a substantial decrease in WCSS*.

*WCSS = within-cluster sum of squares



For the silhouette scores, the optimal number of clusters is typically where the silhouette score reaches its maximum value.

The optimal k might be evident from just one of the methods or might require comparing the results from both. While the Elbow Method is more of a visual and somewhat subjective method, the Silhouette Method provides a more quantifiable metric for cluster validation.

04 PCA

01 Standardize the data (done!)

02 Apply PCA (data fitting)

03 Transform data

04 Calculate EVR (Explained Variance Ratio)

05 Calculate centroids

06 Visualize data in 2-dimensional space

FIT.
In PCA, 'fitting' is like studying for an exam, you're learning and understanding the material.

TRANSFORM.
If 'fitting' is studying for an exam, 'transforming' is like taking the exam, you're applying what you've learned.

Fit and Transform make easier for us to work with in the next steps of our project, like creating customer groups (clustering).

EVR.
In the context of our project, knowing the explained variance ratio helps us decide how many principal components we need to keep.

CENTROIDS.
Centroids help us identify the 'middle' of each cluster, providing a reference point that makes understanding the spread of data points within the cluster easier.

04 PCA

```
evr_rs = pca.explained_variance_ratio_
print(f"Explained variance ratio: {evr_rs}")
```

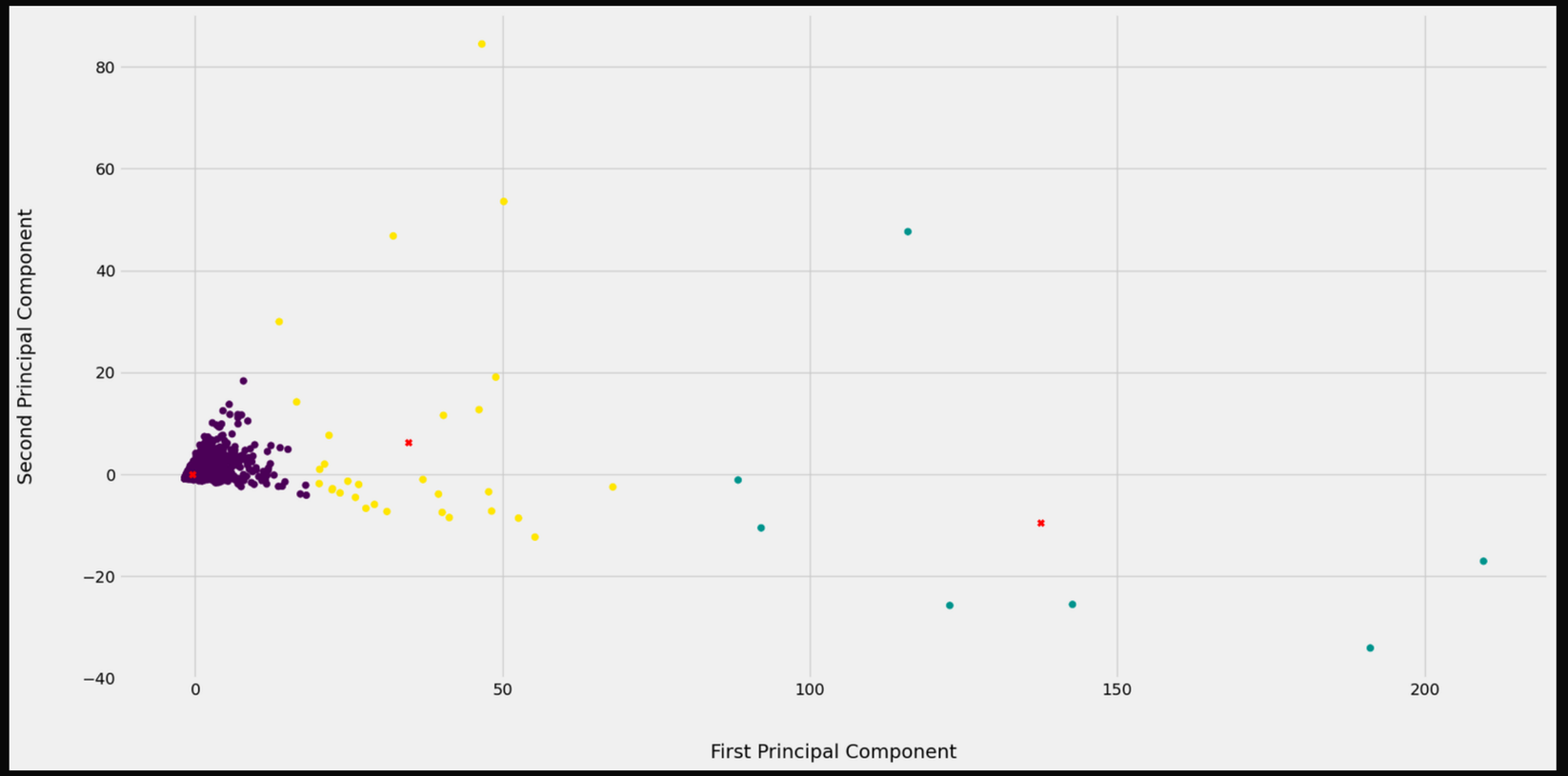
✓ 0.0s

```
Explained variance ratio: [0.87383766 0.11471658]
```

PC1: explains 87% of the variance
PC2: explains approximately 11%

Together: 98% of the total variance in our data

This is an excellent result!
Results are above the typical satisfactory threshold (70-90%)



05 K-means

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3, random_state=0, n_init="auto")
kmeans_clusters = kmeans.fit_predict(pca_rs)
kmeans_clusters
```

✓ 0.0s

	Principal Component 1	Principal Component 2
0	55.27	-12.40
1	1.93	0.84
2	-0.58	-0.66
3	-0.45	-0.09
4	-1.48	-0.74
...	...	...
4331	-1.57	-0.78
4332	-1.64	-0.76
4333	-1.54	-0.63
4334	1.50	7.35
4335	-0.18	-0.20

4336 rows × 2 columns

05 K-means

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3, random_state=0, n_init="auto")
kmeans_clusters = kmeans.fit_predict(pca_rs)
kmeans_clusters
```

✓ 0.0s

	Principal Component 1	Principal Component 2
0	55.27	-12.40
1	1.93	0.84
2	-0.58	-0.66
3	-0.45	-0.09
4	-1.48	-0.74
...	...	...
4331	-1.57	-0.78
4332	-1.64	-0.76
4333	-1.54	-0.63
4334	1.50	7.35
4335	-0.18	-0.20

4336 rows × 2 columns

```
rfm = data_rfm.copy()
rfm["Cluster"] = kmeans_clusters
rfm
```

✓ 0.0s

CustomerID	Recency	Frequency	MonetaryValue	Cluster
12346	326	1	77183.60	2
12347	2	182	4310.00	0
12348	75	27	1437.24	0
12349	19	72	1457.55	0
12350	310	16	294.40	0
...	...	...	...	...
18280	278	10	180.60	0
18281	181	7	80.82	0
18282	8	12	178.05	0
18283	4	687	2039.58	0
18287	43	69	1837.28	0

4336 rows × 4 columns

06 Updating original data with new features

```python
# Create mapping from Cluster to Status
cluster_to_status = {0: "Low value", 1: "High value", 2: "Mid value"}

# Create a backup
rfm_backup = rfm.copy()

# Map the cluster to the status and add it as a new column
rfm["Status"] = rfm["Cluster"].map(cluster_to_status)

# Reset index before merge
rfm_reset = rfm.reset_index()

# Create backup
beforeRFM_data = data.copy()

# Merge the two dataframes on CustomerID
cluster_data = pd.merge(data, rfm_reset[["CustomerID", "Status"]], on="CustomerID", how="left")

# Save data as a csv file
cluster_data.to_csv("Cleaned Dataset/cluster.csv")

cluster_data.head()
```

✓ 2.6s Python

	InvoiceNo	InvoiceDate	StockCode	Description	Quantity	UnitPrice	Revenue	CustomerID	Country	invoiceDate_norm	Year	YearMonth	Month	MonthDate	DayofMonth	Hour	DayofWeek	Month_name	Day_name	Quarterly	Status
0	536365	2010-12-01 08:26:00	21730	holder star frosted tlight glass	6	4.25	25.50	17850	United Kingdom	2010-12-01	2010	2010-12	12	12-01	1	8	2	December	Wednesday	2010Q4	Low value
1	536365	2010-12-01 08:26:00	22752	7 set nesting babushka boxes	2	7.65	15.30	17850	United Kingdom	2010-12-01	2010	2010-12	12	12-01	1	8	2	December	Wednesday	2010Q4	Low value
2	536365	2010-12-01 08:26:00	71053	moroccan white lantern metal	6	3.39	20.34	17850	United Kingdom	2010-12-01	2010	2010-12	12	12-01	1	8	2	December	Wednesday	2010Q4	Low value
3	536365	2010-12-01 08:26:00	84029E	red hottie white heart woolly	6	3.39	20.34	17850	United Kingdom	2010-12-01	2010	2010-12	12	12-01	1	8	2	December	Wednesday	2010Q4	Low value
4	536365	2010-12-01 08:26:00	84029G	water flag bottle hot knitted union	6	3.39	20.34	17850	United Kingdom	2010-12-01	2010	2010-12	12	12-01	1	8	2	December	Wednesday	2010Q4	Low value

BONUS Understanding the clusters

1

BEST CUSTOMERS
HIGH VALUE

- They bought something very recently, with an average of just 6 days since their last purchase
- They buy often, around 1.5K purchases on average
- They spend a lot, about +180K on average

We should focus on keeping these customers happy by giving them special deals, top-notch customer service, and first access to new products.

2

GOOD CUSTOMERS
MID VALUE

- These customers buy regularly
- They make a lot of purchases, about +1K on average
- Spend a good amount, around +45K on average

We could encourage them to buy more often and spend more by making them feel valued, through marketing campaigns based on their past purchases and suggest recommendations.

0

OCCASSIONAL CUSTOMERS
LOW VALUE

- These customers haven't bought anything for a while, with an average of 93 days since their last purchase
- They don't buy often or spend much, with averages around 80 purchases and +1K spent.

We should try to get them to buy again by giving them special offers or looking at what they bought before to understand what they like.

We might also want to ask them for feedback to find out why they haven't been buying and what could get them to buy again.

CHALLENGES

OBSTACLE 1: The Paradox of Choice in Data Selection

Overwhelmed by abundant data options, I focused on identifying an industry in need of nuanced insights. The UK gift retail industry fit the bill, and this clear focus made data selection easier.

OBSTACLE 2: Diverse Analytical Possibilities

While multiple machine learning techniques seemed applicable, I prioritized customer segmentation as it presented significant business value. This guided me in choosing the most relevant methods.

OBSTACLE 3: Debugging Machine Learning Models

The most demanding aspect was troubleshooting machine learning algorithms that weren't performing as expected. Persistent iteration, testing, and finally switching from StandardScaler to RobustScaler ultimately led to more robust results.

The last chapter

Drawing Conclusions from Our Journey

Through extensive EDA, we've unveiled key trends in sales performance, customer behaviour, geographic markets, temporal sales, product and pricing strategies. Each insight provides an opportunity to optimize business operations and strategies.

The complexities of tuning hyperparameters and debugging models provided a hands-on experience with the intricacies of data science. Utilizing clustering algorithms to segment customers demonstrated the power of machine learning in transforming raw data into valuable business insights.

Overall, this project underlines the transformative role of data analysis and machine learning in business decision-making. By converting raw data into actionable insights, businesses can make informed decisions, identify new opportunities, and gain a competitive edge.

THANK YOU!

PLEASE TO MEET YOU.