YEAR
2023

PRESENTER
Eva Faith Wong

SC1015 MINI PROJECT

PT1 | TEAM 4

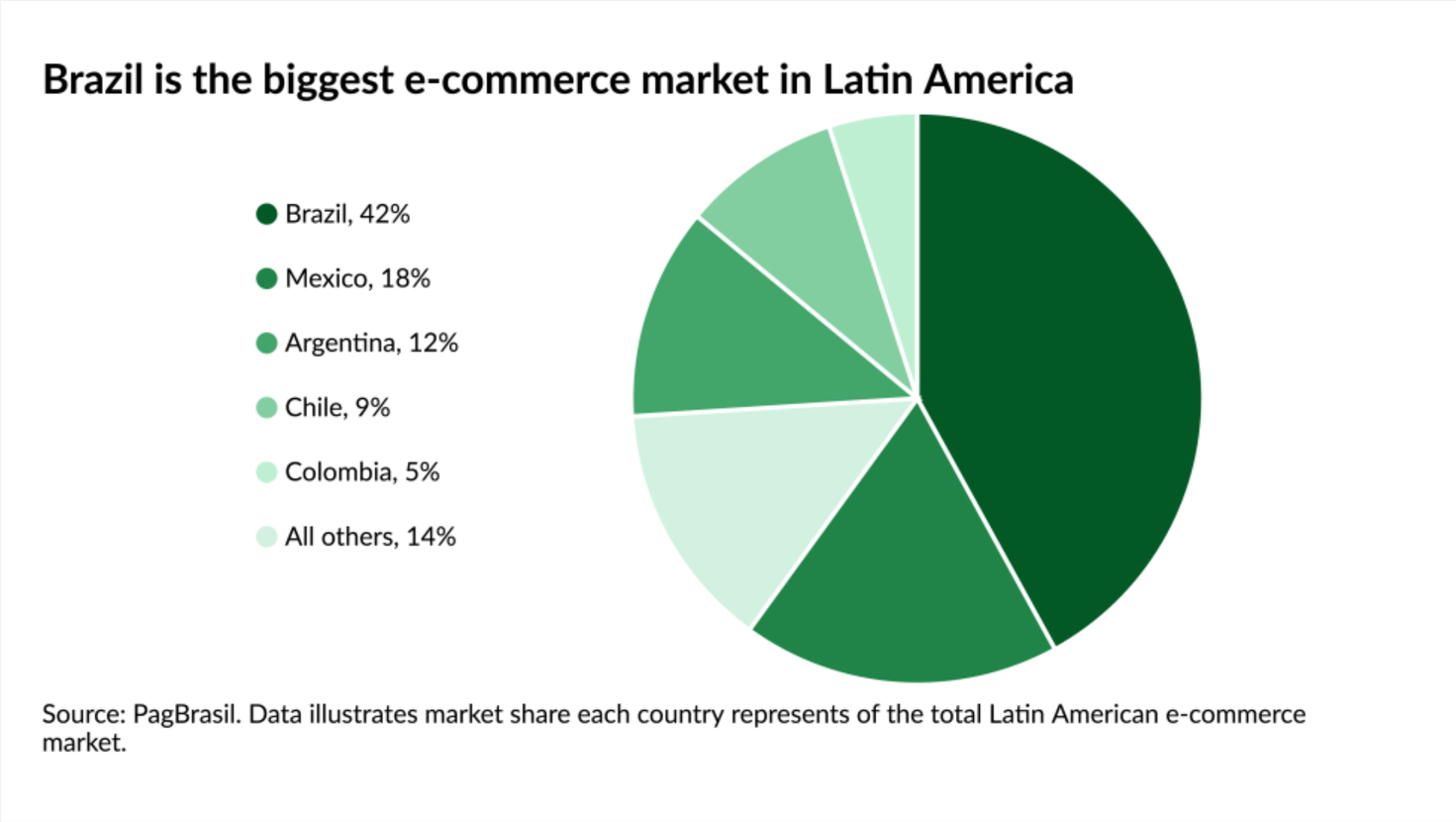
Chen Yanjin
Crystal Lee Chau
Eva Faith Wong

Content

NEXT

INTRODUCTION

About E-Commerce in Brazil



NEXT

INTRODUCTION

About E-Commerce in Brazil

NEXT

INTRODUCTION

E-commerce: Olist

NEXT

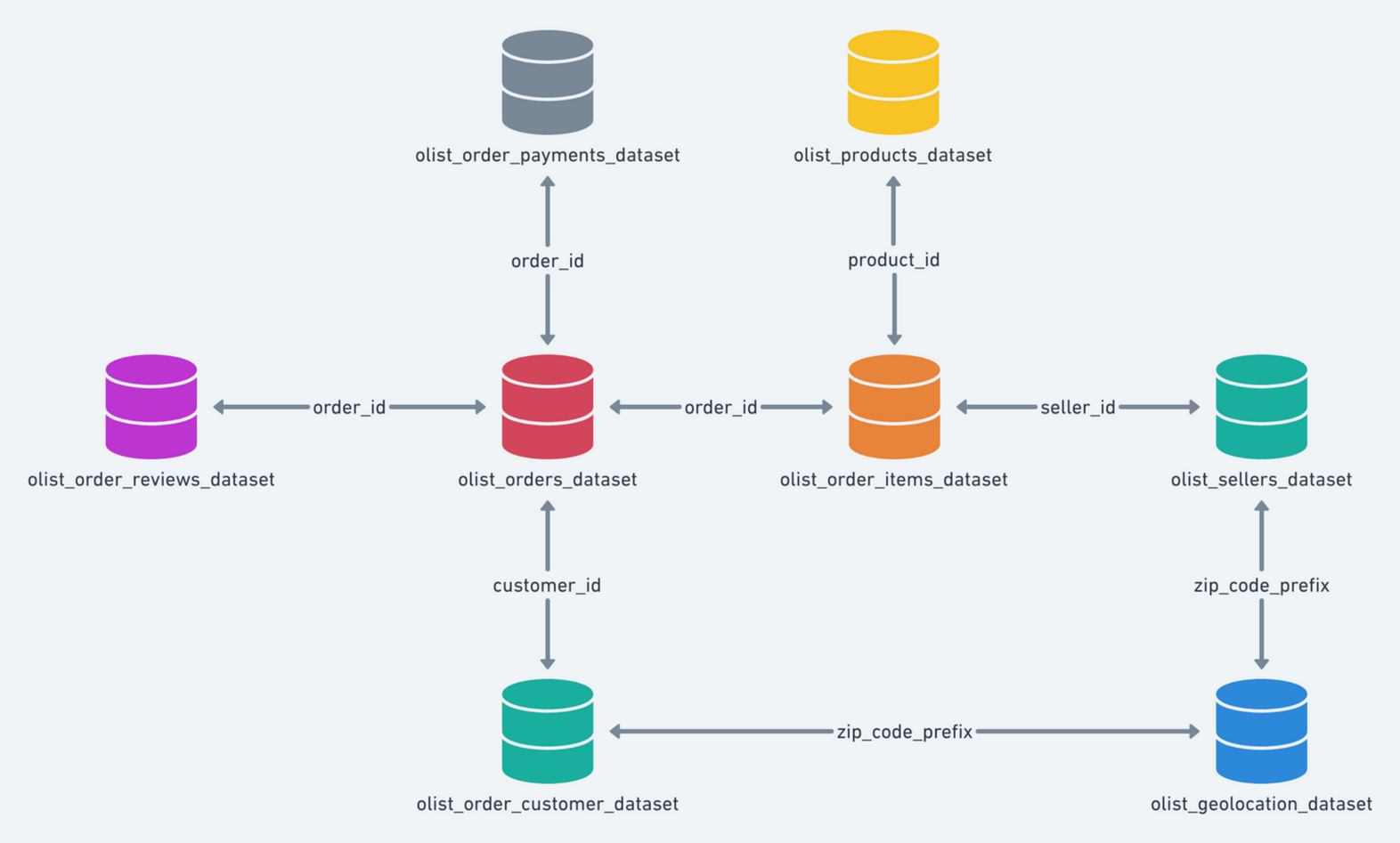
CONTENT

1. Motivation

NEXT

MOTIVATION

Data Model



NEXT

PROBLEM STATEMENT

Analyze the customer behaviour and purchase patterns on the Brazilian E-commerce website to provide insights to improve the sales and marketing strategy.

NEXT

CONTENT

2. Data Preparation and Cleaning

NEXT

DATA PREPARATION & CLEANING

Data Preparation

- Datatset is structured based on the **order ID**
- Handling missing and duplicated values
- Transform the column into appropriate data type
- Merging all datasets
- Cleaning final dataset

```
print(cleaned_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 110012 entries, 0 to 110838
Data columns (total 18 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   customer_unique_id             110012 non-null  object
 1   customer_state                 110012 non-null  object
 2   customer_zip_code_prefix       110012 non-null  object
 3   order_id                       110012 non-null  object
 4   order_item_id                  110012 non-null  object
 5   price                          110012 non-null  float64
 6   freight_value                  110012 non-null  float64
 7   order_purchase_timestamp       110012 non-null  datetime64[ns]
 8   order_estimated_delivery_date  110012 non-null  datetime64[ns]
 9   order_delivered_customer_date  110012 non-null  datetime64[ns]
 10  delivery_days                  110012 non-null  int64
 11  delay                          110012 non-null  int64
 12  product_id                     110012 non-null  object
 13  product_category_name          110012 non-null  category
 14  review_id                      110012 non-null  object
 15  review_score                   110012 non-null  int64
 16  seller_id                      110012 non-null  object
 17  seller_state                   110012 non-null  category
dtypes: category(2), datetime64[ns](3), float64(2), int64(3), object(8)
memory usage: 14.5+ MB
```

```
translate = pd.read_csv('product_category_name_translation.csv')
translate.head()
```

	product_category_name	product_category_name_english
0	beleza_saude	health_beauty
1	informatica_acessorios	computers_accessories
2	automotivo	auto
3	cama_mesa_banho	bed_bath_table
4	moveis_decoracao	furniture_decor

```
cleaned_data = orders.merge(order_items, on = 'order_id', how = 'left')
cleaned_data = cleaned_data.T.drop_duplicates().T
cleaned_data.head()
```

NEXT

CONTENT

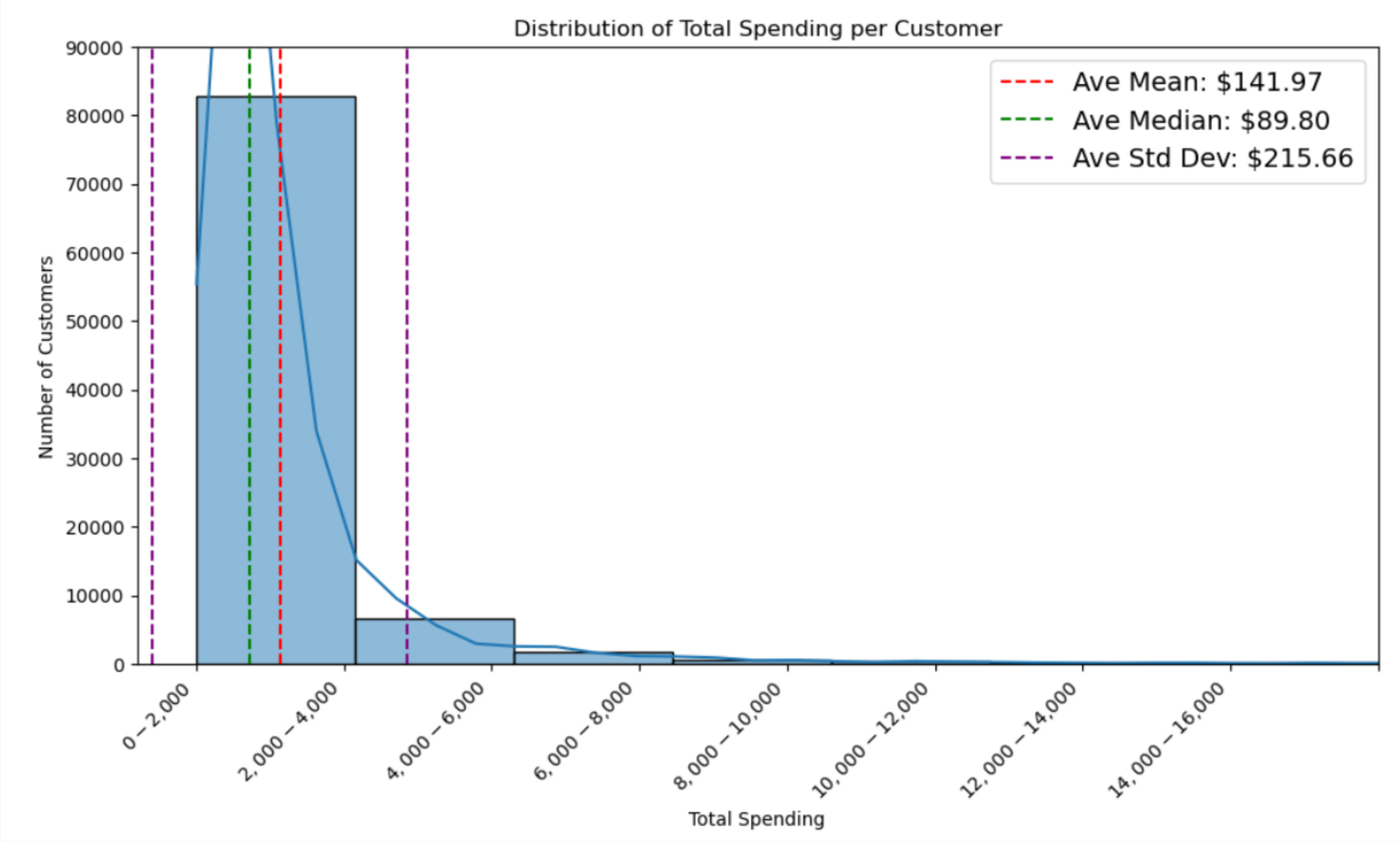
3. Exploratory Data Analysis

NEXT

EXPLORATORY DATA

Total Spending Per Customer

MAJORITY OF CUSTOMERS SPENT RELATIVELY SMALL AMOUNTS

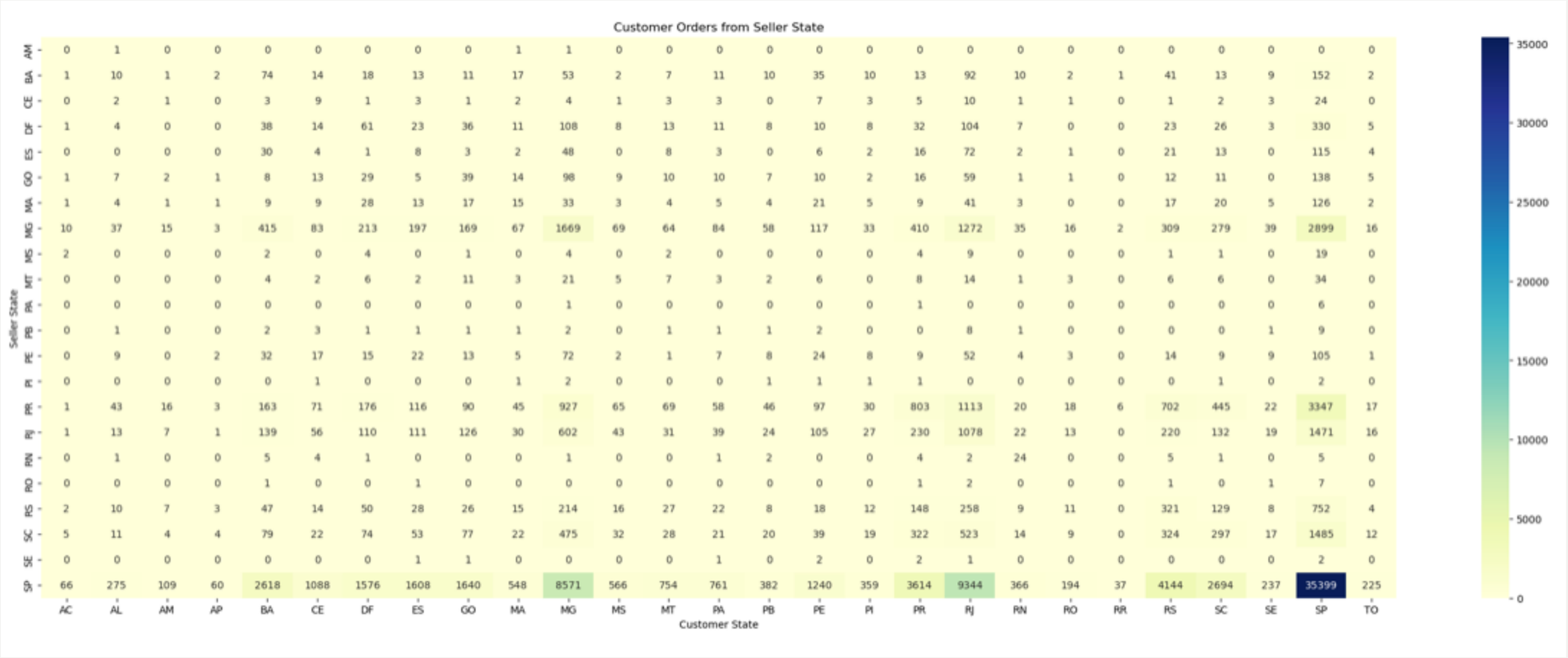


NEXT

EXPLORATORY DATA

Customer Orders from Seller State

RELATIONSHIP BETWEEN THE SELLER STATE AND CUSTOMER STATE

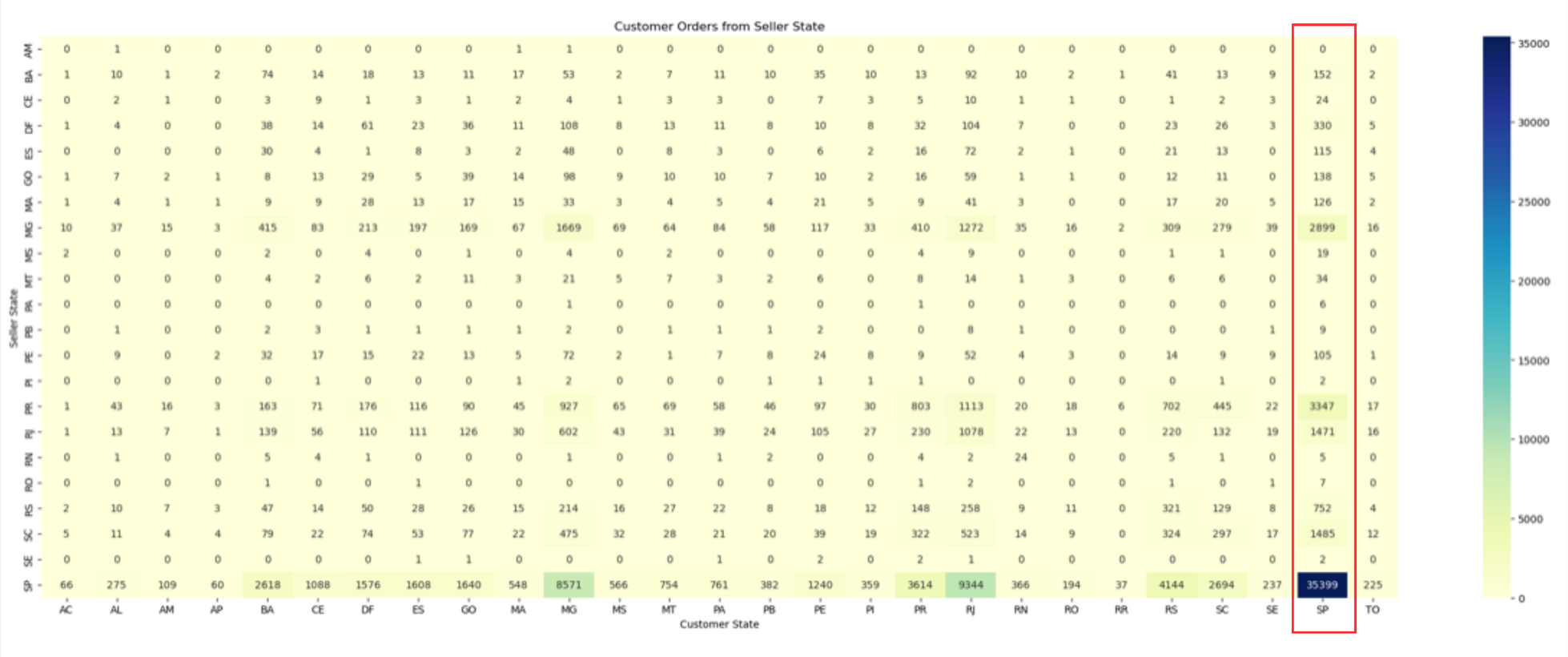


NEXT

EXPLORATORY DATA

Customer Orders from Seller State

RELATIONSHIP BETWEEN THE SELLER STATE AND CUSTOMER STATE



SÃO PAULO

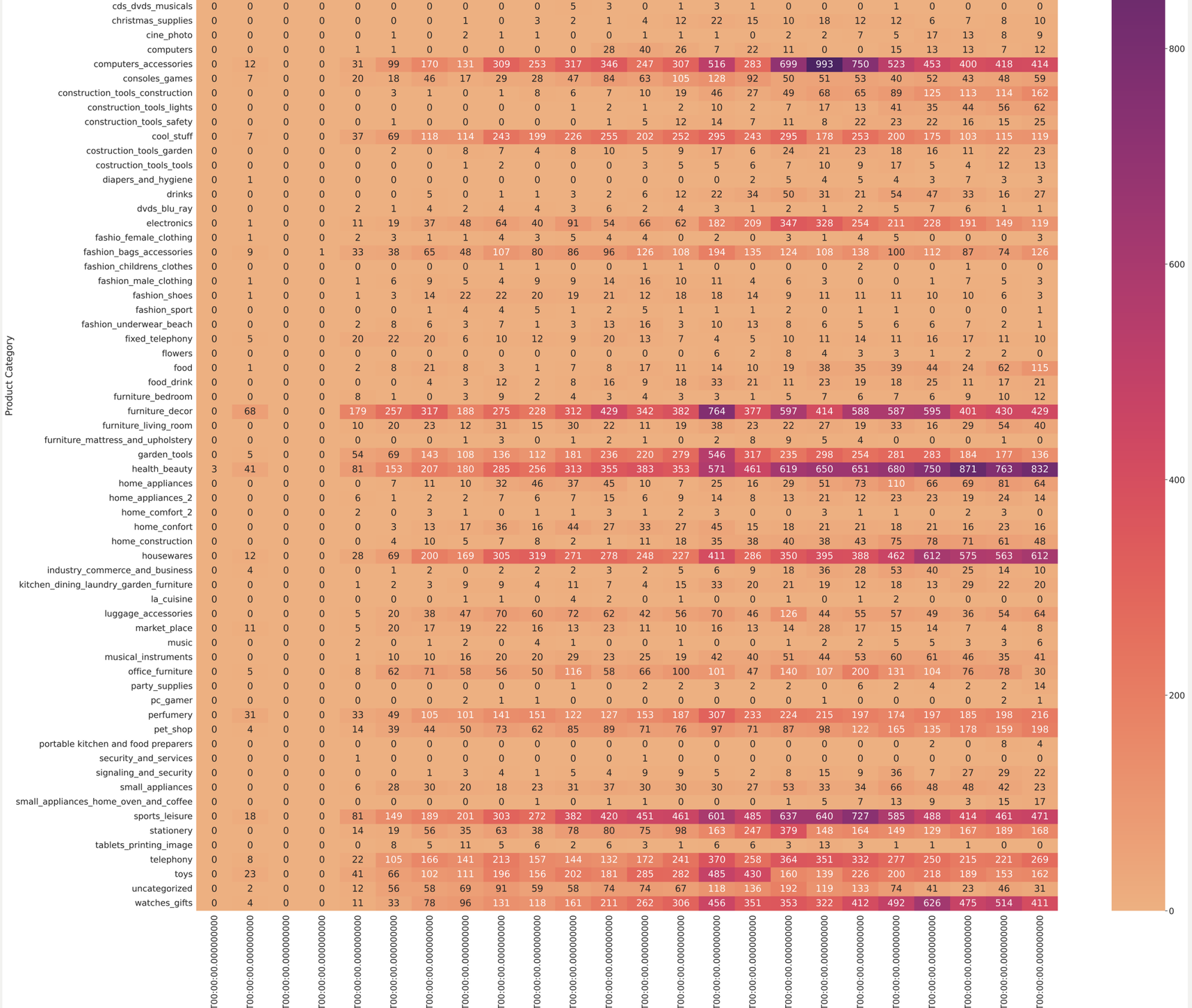
NEXT

EXPLORATORY

Products Category Trade Per Month

SEASONAL PATTERN WITH HIGHER SALES ON CERTAIN PRODUCTS

NEXT

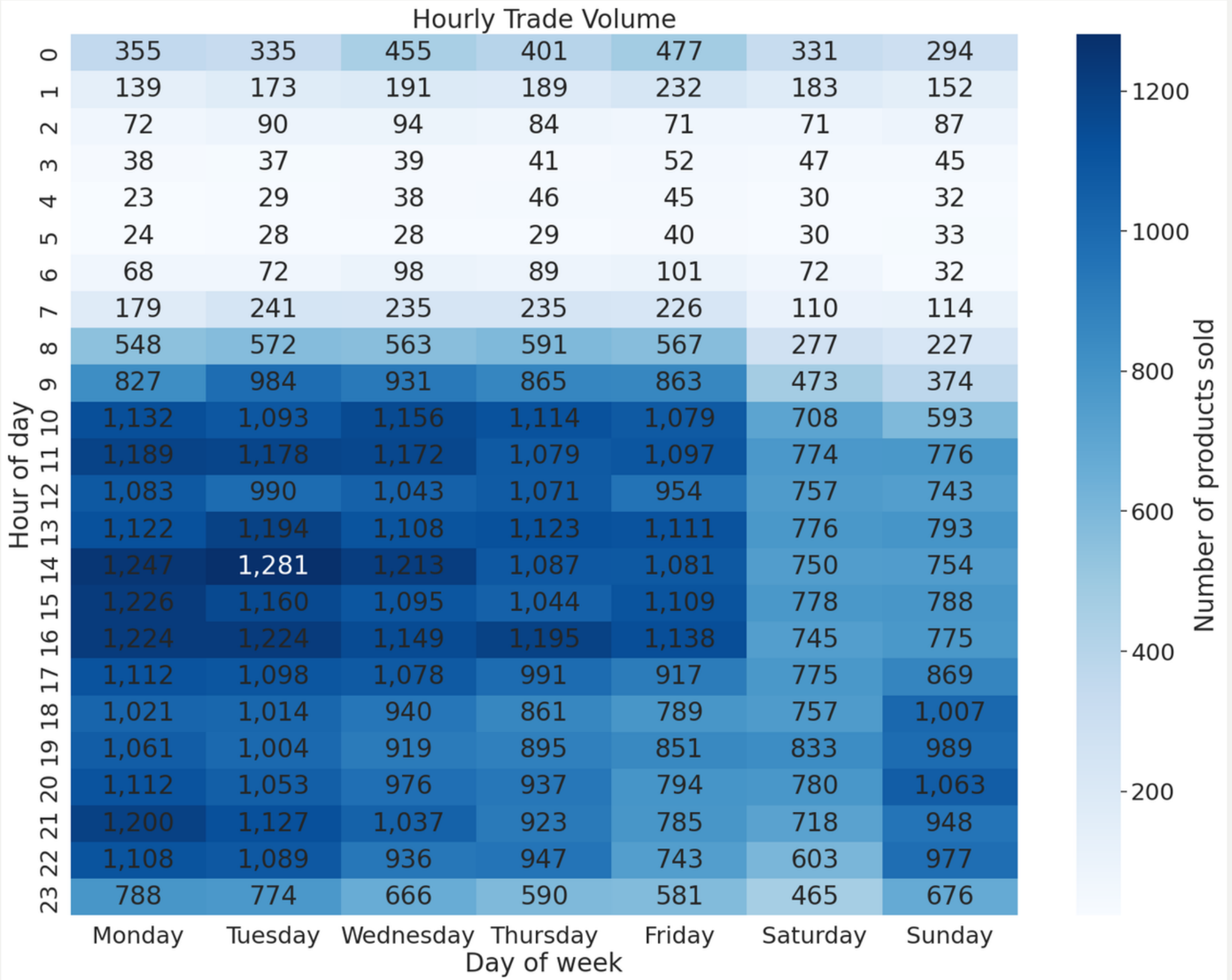

Product Category
cds_dvds_musicals 0 0 0 0 0 0 0 0 0 5 3 0 1 3 1 0 0 0 1 0 0 0 0
christmas_supplies 0 0 0 0 0 0 0 1 0 3 2 1 4 12 22 15 10 18 12 12 6 7 8 10
cine_photo 0 0 0 0 0 1 0 2 1 1 0 0 1 1 1 0 2 2 7 5 17 13 8 9
computers 0 0 0 0 1 1 0 0 0 0 0 28 40 26 7 22 11 0 0 15 13 13 7 12
computers_accessories 0 12 0 0 31 99 170 131 309 253 317 346 247 307 516 283 699 993 750 523 453 400 418 414
consoles_games 0 7 0 0 20 18 46 17 29 28 47 84 63 105 128 92 50 51 53 40 52 43 48 59
construction_tools_construction 0 0 0 0 0 3 1 0 1 8 6 7 10 19 46 27 49 68 65 89 125 113 114 162
construction_tools_lights 0 0 0 0 0 0 0 0 0 0 1 2 1 2 10 2 7 17 13 41 35 44 56 62
construction_tools_safety 0 0 0 0 0 1 0 0 0 0 0 1 5 12 14 7 11 8 22 23 22 16 15 25
cool_stuff 0 7 0 0 37 69 118 114 243 199 226 255 202 252 295 243 295 178 253 200 175 103 115 119
costruction_tools_garden 0 0 0 0 0 2 0 8 7 4 8 10 5 9 17 6 24 21 23 18 16 11 22 23
costruction_tools_tools 0 0 0 0 0 0 0 1 2 0 0 0 3 5 5 6 7 10 9 17 5 4 12 13
diapers_and_hygiene 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 2 5 4 5 4 3 7 3 3
drinks 0 0 0 0 0 0 5 0 1 1 3 2 6 12 22 34 50 31 21 54 47 33 16 27
dvds_blu_ray 0 0 0 0 2 1 4 2 4 4 3 6 2 4 3 1 2 1 2 5 7 6 1 1
electronics 0 1 0 0 11 19 37 48 64 40 91 54 66 62 182 209 347 328 254 211 228 191 149 119
fashio_female_clothing 0 1 0 0 2 3 1 1 4 3 5 4 4 0 2 0 3 1 4 5 0 0 0 3
fashion_bags_accessories 0 9 0 1 33 38 65 48 107 80 86 96 126 108 194 135 124 108 138 100 112 87 74 126
fashion_childrens_clothes 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 2 0 0 1 0 0
fashion_male_clothing 0 1 0 0 1 6 9 5 4 9 9 14 16 10 11 4 6 3 0 0 1 7 5 3
fashion_shoes 0 1 0 0 1 3 14 22 22 20 19 21 12 18 18 14 9 11 11 11 10 10 6 3
fashion_sport 0 0 0 0 0 0 1 4 4 5 1 2 5 1 1 1 2 0 1 1 0 0 0 1
fashion_underwear_beach 0 0 0 0 2 8 6 3 7 1 3 13 16 3 10 13 8 6 5 6 6 7 2 1
fixed_telephony 0 5 0 0 20 22 20 6 10 12 9 20 13 7 4 5 10 11 14 11 16 17 11 10
flowers 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 2 8 4 3 3 1 2 2 0
food 0 1 0 0 2 8 21 8 3 1 7 8 17 11 14 10 19 38 35 39 44 24 62 115
food_drink 0 0 0 0 0 0 4 3 12 2 8 16 9 18 33 21 11 23 19 18 25 11 17 21
furniture_bedroom 0 0 0 0 8 1 0 3 9 2 4 3 4 3 3 1 5 7 6 7 6 9 10 12
furniture_decor 0 68 0 0 179 257 317 188 275 228 312 429 342 382 764 377 597 414 588 587 595 401 430 429
furniture_living_room 0 0 0 0 10 20 23 12 31 15 30 22 11 19 38 23 22 27 19 33 16 29 54 40
furniture_mattress_and_upholstery 0 0 0 0 0 0 0 1 3 0 1 2 1 0 2 8 9 5 4 0 0 0 1 0
garden_tools 0 5 0 0 54 69 143 108 136 112 181 236 220 279 546 317 235 298 254 281 283 184 177 136
health_beauty 3 41 0 0 81 153 207 180 285 256 313 355 383 353 571 461 619 650 651 680 750 871 763 832
home_appliances 0 0 0 0 0 7 11 10 32 46 37 45 10 7 25 16 29 51 73 110 66 69 81 64
home_appliances_2 0 0 0 0 6 1 2 2 7 6 7 15 6 9 14 8 13 21 12 23 23 19 24 14
home_comfort_2 0 0 0 0 2 0 3 1 0 1 1 3 1 2 3 0 0 3 1 1 0 2 3 0
home_confort 0 0 0 0 0 3 13 17 36 16 44 27 33 27 45 15 18 21 21 18 21 16 23 16
home_construction 0 0 0 0 0 4 10 5 7 8 2 1 11 18 35 38 40 38 43 75 78 71 61 48
housewares 0 12 0 0 28 69 200 169 305 319 271 278 248 227 411 286 350 395 388 462 612 575 563 612
industry_commerce_and_business 0 4 0 0 0 1 2 0 2 2 2 3 2 5 6 9 18 36 28 53 40 25 14 10
kitchen_dining_laundry_garden_furniture 0 0 0 0 1 2 3 9 9 4 11 7 4 15 33 20 21 19 12 18 13 29 22 20
la_cuisine 0 0 0 0 0 0 0 1 1 0 4 2 0 1 0 0 1 0 1 2 0 0 0 0
luggage_accessories 0 0 0 0 5 20 38 47 70 60 72 62 42 56 70 46 126 44 55 57 49 36 54 64
market_place 0 11 0 0 5 20 17 19 22 16 13 23 11 10 16 13 14 28 17 15 14 7 4 8
music 0 0 0 0 2 0 1 2 0 4 1 0 0 1 0 0 1 2 2 5 5 3 3 6
musical_instruments 0 0 0 0 1 10 10 16 20 20 29 23 25 19 42 40 51 44 53 60 61 46 35 41
office_furniture 0 5 0 0 8 62 71 58 56 50 116 58 66 100 101 47 140 107 200 131 104 76 78 30
party_supplies 0 0 0 0 0 0 0 0 0 0 1 0 2 2 3 2 2 0 6 2 4 2 2 14
pc_gamer 0 0 0 0 0 0 0 2 1 1 0 0 0 0 0 0 0 1 0 0 0 0 2 1
perfumery 0 31 0 0 33 49 105 101 141 151 122 127 153 187 307 233 224 215 197 174 197 185 198 216
pet_shop 0 4 0 0 14 39 44 50 73 62 85 89 71 76 97 71 87 98 122 165 135 178 159 198
portable kitchen and food preparers 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 8 4
security_and_services 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
signaling_and_security 0 0 0 0 0 0 1 3 4 1 5 4 9 9 5 2 8 15 9 36 7 27 29 22
small_appliances 0 0 0 0 6 28 30 20 18 23 31 37 30 30 30 27 53 33 34 66 48 48 42 23
small_appliances_home_oven_and_coffee 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 5 7 13 9 3 15 17
sports_leisure 0 18 0 0 81 149 189 201 303 272 382 420 451 461 601 485 637 640 727 585 488 414 461 471
stationery 0 0 0 0 14 19 56 35 63 38 78 80 75 98 163 247 379 148 164 149 129 167 189 168
tablets_printing_image 0 0 0 0 0 8 5 11 5 6 2 6 3 1 6 6 3 13 3 1 1 1 0 0
telephony 0 8 0 0 22 105 166 141 213 157 144 132 172 241 370 258 364 351 332 277 250 215 221 269
toys 0 23 0 0 41 66 102 111 196 156 202 181 285 282 485 430 160 139 226 200 218 189 153 162
uncategorized 0 2 0 0 12 56 58 69 91 59 58 74 74 67 118 136 192 119 133 74 41 23 46 31
watches_gifts 0 4 0 0 11 33 78 96 131 118 161 211 262 306 456 351 353 322 412 492 626 475 514 411
T00:00:00.000000000
0
200
400
600
800

NEXT

EXPLORATORY

Customer Purchase Timeframe

COMSUMER MORE ACTIVE FOR PURCHAISNG IS BETWEEN 10:00 AND 16:00 ON WEEKDAYS

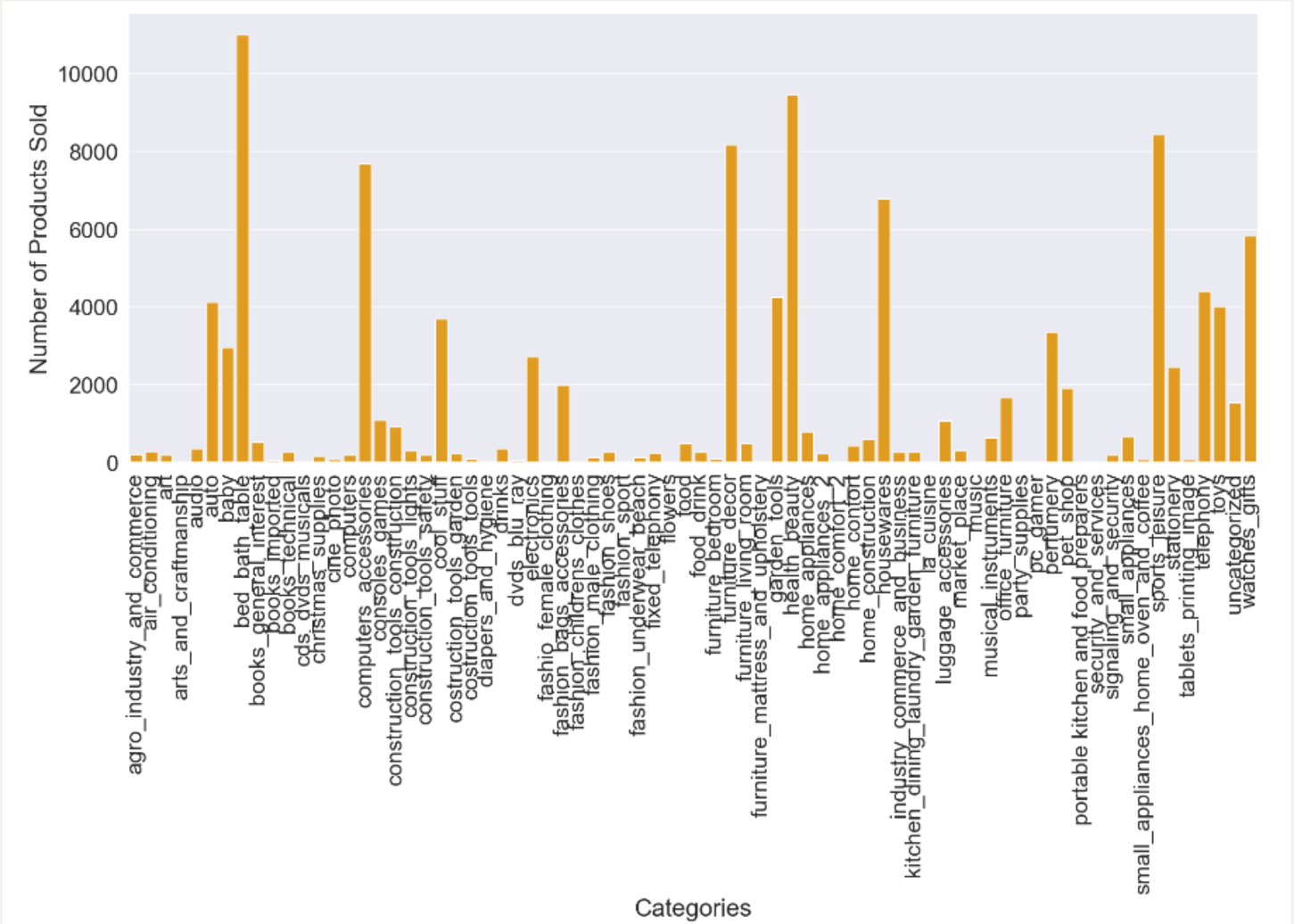


NEXT

EXPLORATORY

Products Sold Per Category

"BED_BATH_TABLE" AND "HEALTH_BEAUTY" CATEGORIES ARE IN HIGH DEMANDS

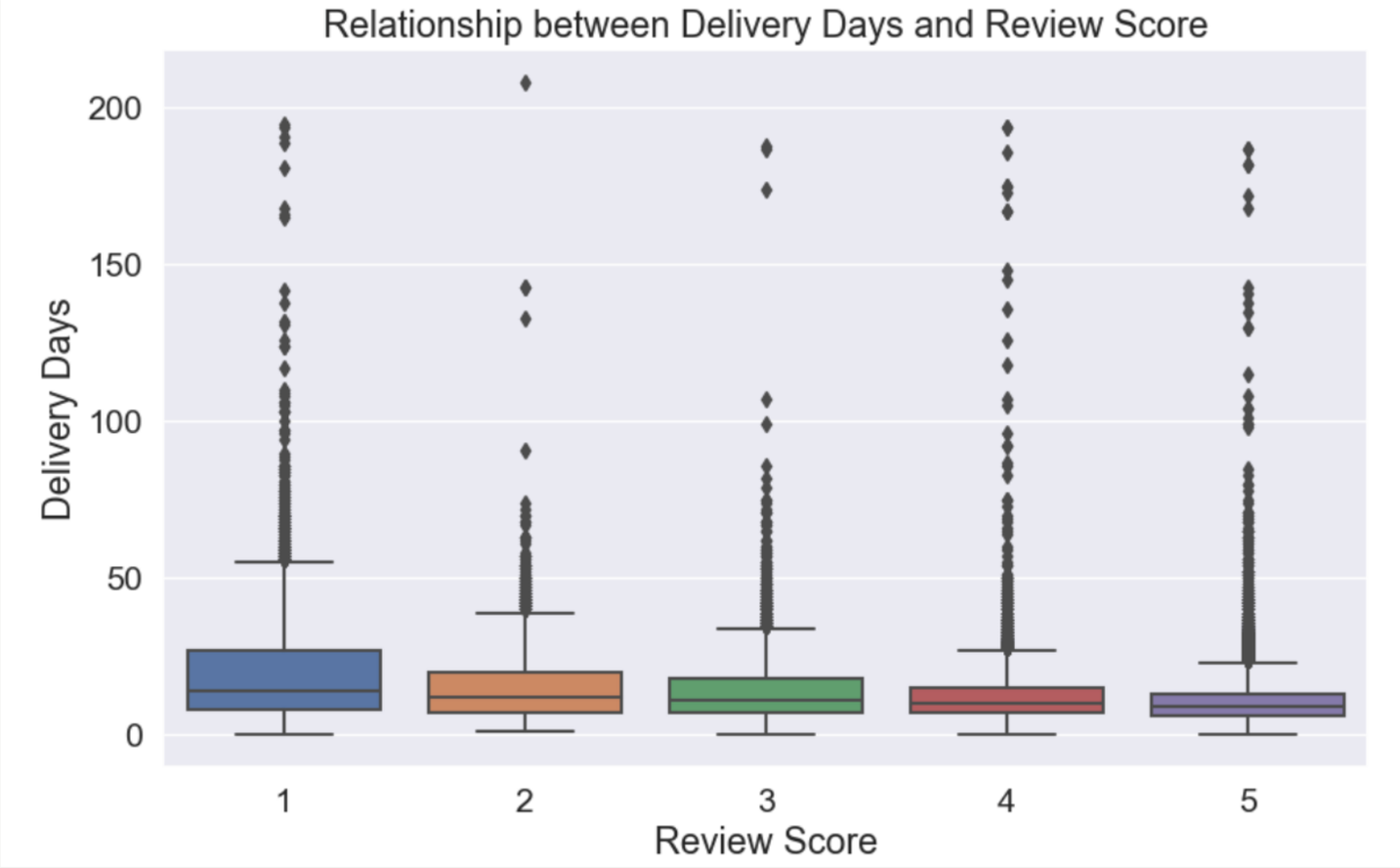


NEXT

EXPLORATORY

Review Score & Delivery Days

DELIVERY DURATIONS AFFECT REVIEW SCORES SIGNIFICANTLY



NEXT

CONTENT

4. Machine Learning Techniques

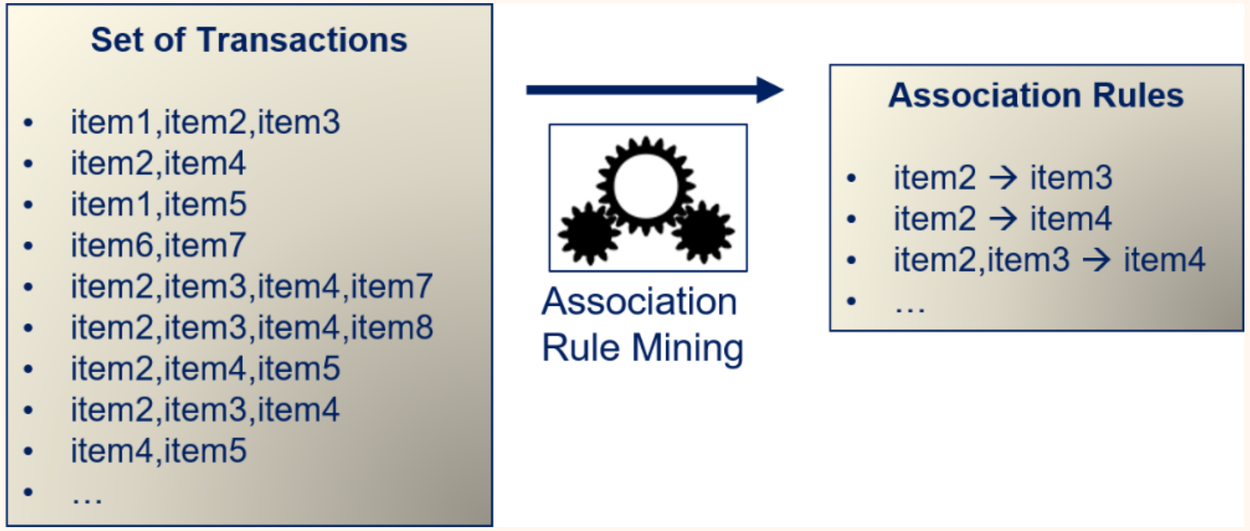
NEXT

Machine Learning

ASSOCIATION RULE MINING

LINEAR REGRESSION

PREDICTIVE MODELLING



MACHINE LEARNING

ASSOCIATION RULE MINING

ARM uses algorithms to analyse the data and identify the relationships between variables in large dataset. The resulting rules are often expressed in the form of "if-then"

- "if" - antecedent (the condition being tested)
- "then" - consequent (the outcome that occurs if the condition is met)

NEXT

ASSOCIATION RULE MINING

It can be measured in 3 common ways:

SUPPORT

For Movie Recommendation, we calculate it as:

$$support(M) = \frac{\text{number of user watchlists containing M}}{\text{total number of user watchlists}}$$

CONFIDENCE

For Movie Recommendation, we calculate it as:

$$confidence(M_1 \rightarrow M_2) = \frac{\text{number of user watchlists containing } M_1 \text{ and } M_2}{\text{number of user watchlists containing } M_1}$$

wheras for Market Basket Optimization, we calculate it as:

$$confidence(I_1 \rightarrow I_2) = \frac{\text{number of transactions containing } I_1 \text{ and } I_2}{\text{number of transactions containing } I_1}$$

LIFT

For Movie Recommendation, we calculate it as:

$$lift(M_1 \rightarrow M_2) = \frac{Confidence(M_1 \rightarrow M_2)}{Support(M_2)}$$

wheras for Market Basket Optimization, we calculate it as:

$$lift(I_1 \rightarrow I_2) = \frac{Confidence(I_1 \rightarrow I_2)}{Support(I_2)}$$

ASSOCIATION RULE MINING

Apriori Algorithm consists of:

Step 1: Set a minimum support and confidence.

Step 2: Take all the subsets in transactions having higher support than minimum support.

Step 3: Take all the rules of these subsets having higher confidence than minimum confidence.

Step 4: Sort the rules by decreasing lift.

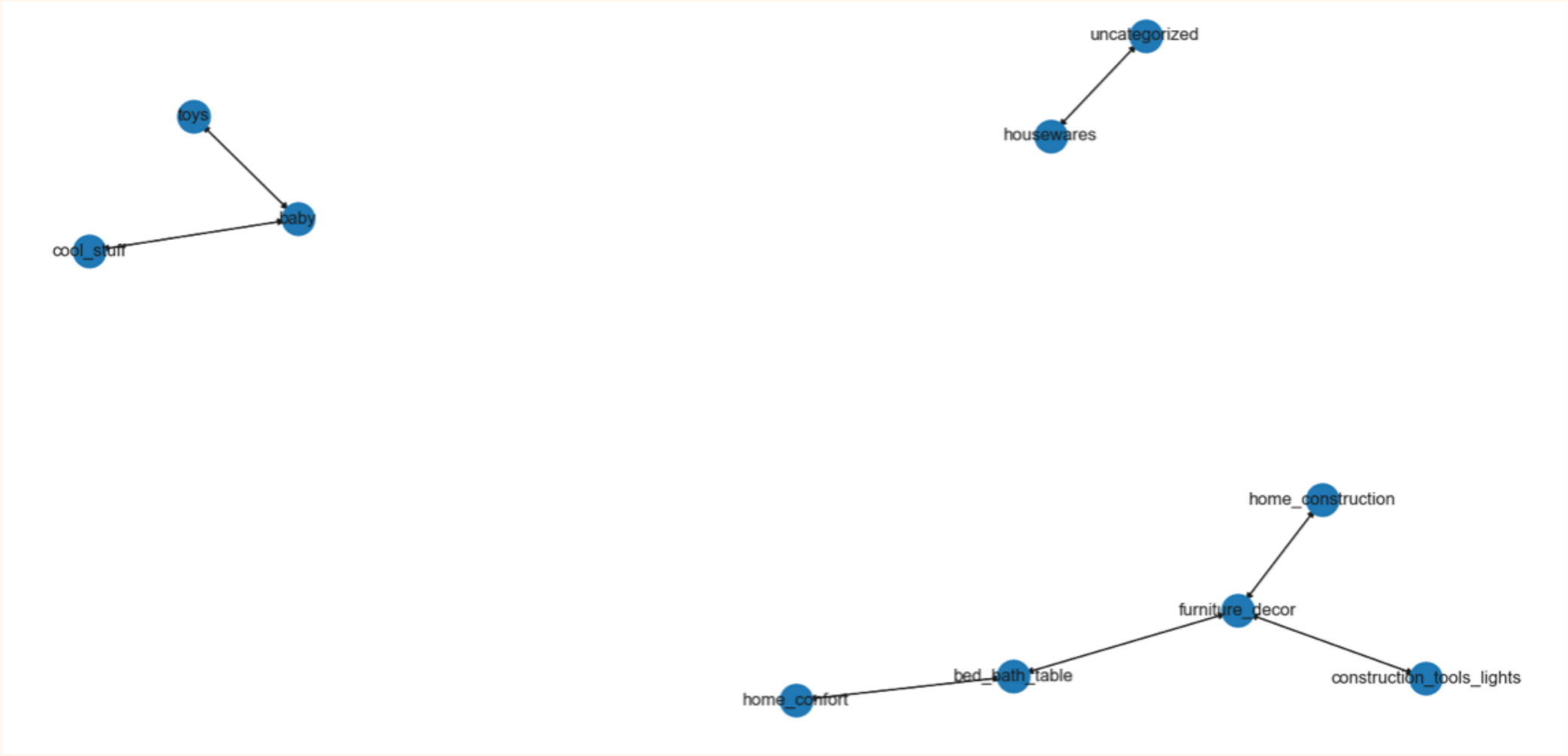
```
frequent_itemsets = apriori(df_encoded, min_support=0.0001,
                            use_colnames=True)
frequent_itemsets = frequent_itemsets.sort_values('support',
                                                  ascending=False)

frequent_itemsets
```

Use association_rules function

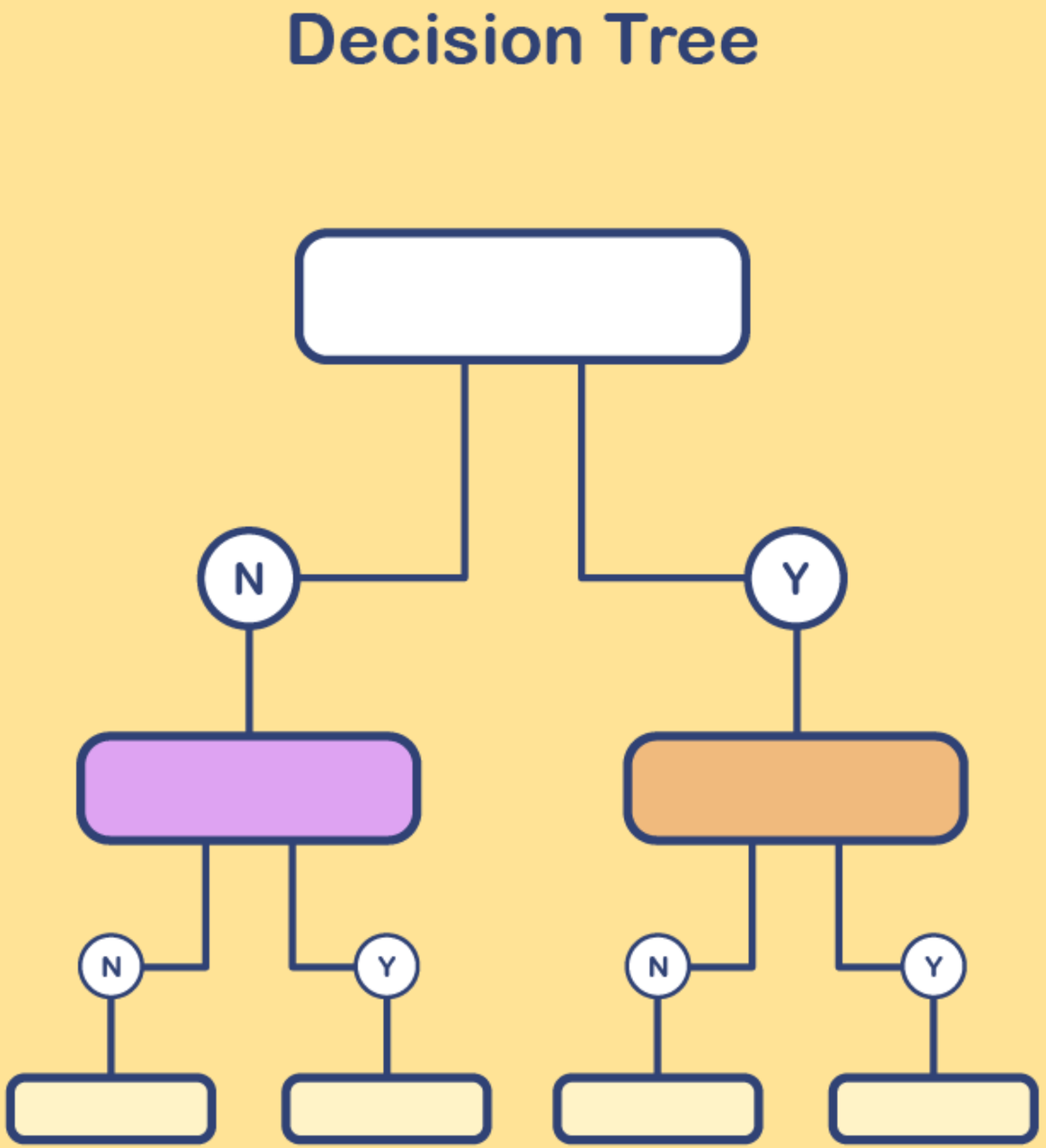
```
rules = association_rules(frequent_itemsets,
                          metric="lift",
                          min_threshold=0.1)
rules = rules.sort_values('lift', ascending=False)

rules
```

MACHINE LEARNING

ASSOCIATION RULE MINING



NEXT



MACHINE LEARNING

Decision Tree

A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered.

NEXT

MACHINE LEARNING

Decision Tree

```
# Import required libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Prepare the data
X = cleaned_data[['customer_state', 'customer_zip_code_prefix', 'price', 'freight_value', 'delivery_days', 'review_score', 'selle
y = cleaned_data['product_category_name']
X = pd.get_dummies(X, columns=['customer_state', 'seller_state'])

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

# Train the decision tree model
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)

# Calculate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

MACHINE LEARNING

Decision Tree

Example 1:
This array represents a customer from the state of São Paulo (SP) with a zip code prefix of 10000, who purchased an electronics product for a price of 50.0.

```
new_data_df = pd.DataFrame({
    'customer_state': ['SP'],
    'customer_zip_code_prefix': [10000],
    'price': [50.0],
    'product_category_name': ['electronics']})

new_data_df = pd.get_dummies(new_data_df,
                             columns=['customer_state',
                                      'product_category_name'])

new_data = new_data_df.reindex(columns=X.columns, fill_value=0)

prediction = clf.predict(new_data)

print(prediction)
```

Example 2:
This array represents a customer from the state of Rio de Janeiro (RJ) with a zip code prefix of 20000, who purchased a home appliance product for a price of 80.0.

```
new_data_df2 = pd.DataFrame({
    'customer_state': ['RJ'],
    'customer_zip_code_prefix': [20000],
    'price': [80.0],
    'product_category_name': ['home appliances']
})

new_data_df2 = pd.get_dummies(new_data_df2,
                              columns=['customer_state',
                                       'product_category_name'])

new_data = new_data_df2.reindex(columns=X.columns, fill_value=0)

prediction = clf.predict(new_data)

print(prediction)
```

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

data = cleaned_data[['customer_unique_id', 'customer_state',
                     'customer_zip_code_prefix', 'order_id',
                     'order_item_id', 'price', 'freight_value',
                     'order_purchase_timestamp', 'order_estimated_delivery_date',
                     'order_delivered_customer_date', 'delivery_days', 'delay',
                     'product_id', 'review_id', 'review_score', 'seller_id',
                     'seller_state']]

X = data[['delivery_days', 'review_score']]
y = data['delay']
X = X.dropna()
y = y[X.index]

split = int(len(X) * 0.8)
X_train = X[:split]
y_train = y[:split]
X_test = X[split:]
y_test = y[split:]

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)
print('R2 score:', r2)
```

MACHINE LEARNING

Linear Regression

To predict customer behavior and preferences based on review scores and delivery time, we can use a supervised machine learning technique, Linear Regression.

NEXT

MACHINE LEARNING

Linear Regression

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

# Select the relevant columns
data = cleaned_data[['customer_unique_id', 'customer_state', 'customer_zip_code_prefix', 'order_id', 'order_item_id', 'price', 't

# Prepare the data
X = data[['delivery_days', 'review_score']]
y = data['delay']
X = X.dropna()
y = y[X.index]

# Split the data into training and testing sets
split = int(len(X) * 0.8)
X_train = X[:split]
y_train = y[:split]
X_test = X[split:]
y_test = y[split:]

# Train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
r2 = r2_score(y_test, y_pred)
print('R2 score:', r2)
```

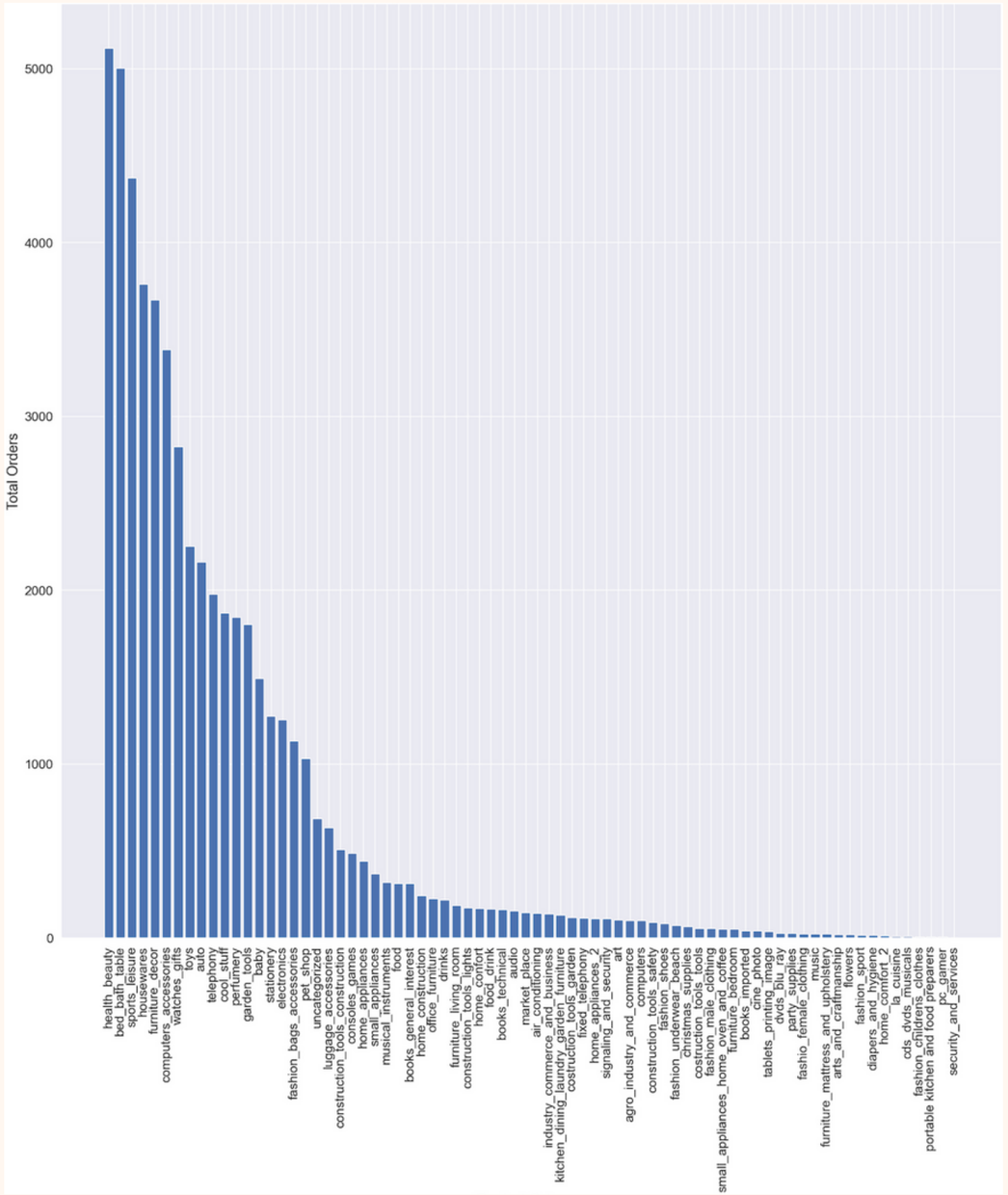
```
R2 score: 0.34342029791113604
```

NEXT

MACHINE LEARNING

Linear Regression

product_category_name	total_orders	total_revenue
health_beauty	5119	605740.62
bed_bath_table	5002	470755.82
sports_leisure	4370	465243.13
housewares	3759	322008.03
furniture_decor	3671	305607.66
...	...	...
cds_dvds_musicals	9	485.00
fashion_childrens_clothes	6	429.96
portable kitchen and food preparers	5	1641.31
pc_gamer	4	786.99
security_and_services	0	0.00

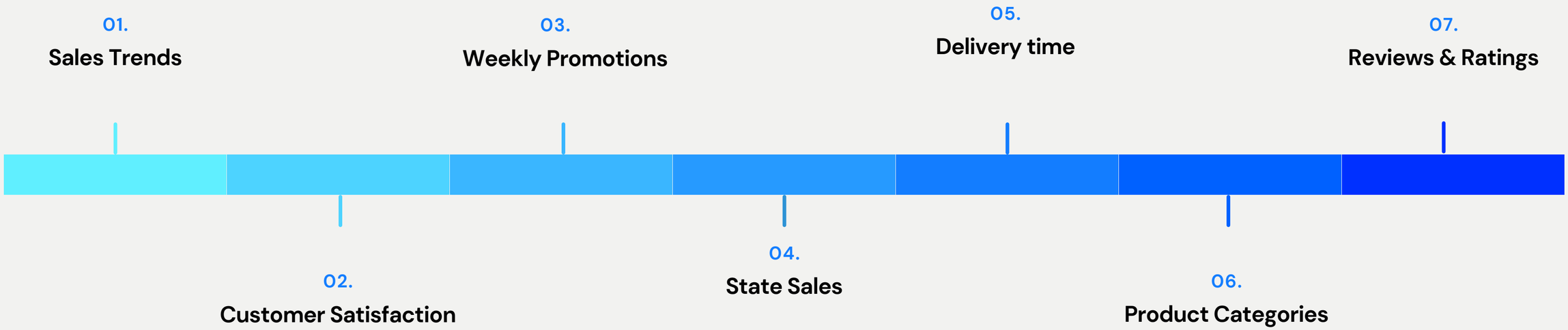


CONTENT

5. Data-Driven Insights & Recommendations

NEXT

Data-Driven Insights & Recommendations



NEXT

Thank you!