

# Documentation Data Mining

---

by:

- عبدالرحمن رعدان
- مصعب الحبيشي
- هادي الجمعي
- مالك المصباحي
- عبدالعزيز عبدالغني
- ماجد النائب

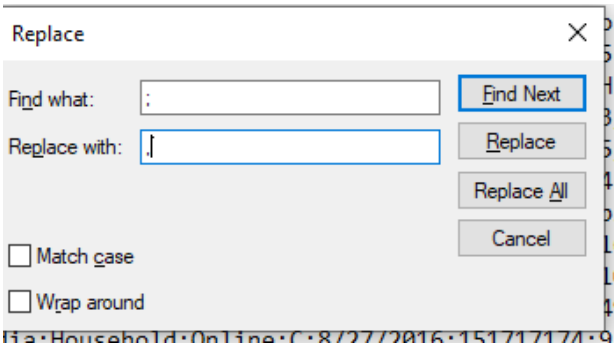
: اشراف

أ. ابراهيم الذارحي

# Data Preprocessing Steps

## Handling Delimiters

- Step 1:** Converting Commas to Semicolons in the Dataset  
**Objective:** Ensure the dataset is properly formatted by replacing all semicolons ( ; ) with commas ( , ).



- Step 2:** Clean the "Country" Column Data **Objective:** Ensure data quality by addressing issues with special characters. In our database, we found that the value "Cote d'Ivoire" in the "Country" column contains an apostrophe, which is causing an error in the Wiki.

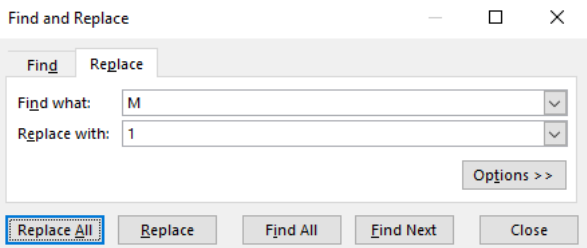
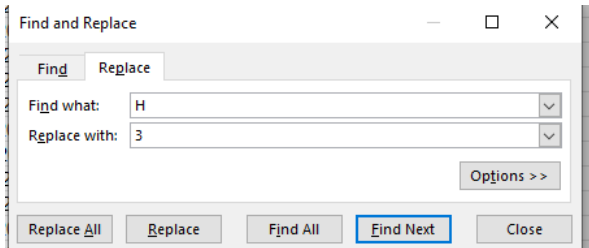
**Problem:** The value "Cote d'Ivoire" includes an apostrophe that leads to errors during data analysis or query execution.

- Step 3:** Re-encoding the "Order Priority" Column We performed re-encoding of the textual values in the "Order Priority" column into numerical values to facilitate processing by machine learning algorithms.

The values were transformed as follows:

Critical → 3  
High → 2  
Medium → 1  
Low → 0

This step was necessary because some algorithms, such as Naïve Bayes and K-Means, require numerical data for analysis and processing.



# Algorithm Implementation

## Apriori Algorithm (Association Rule Mining)

- **Objective:** Discover frequent itemsets and generate association rules.
- **Selected Columns:** Item Type, Sales Channel.
- **Parameters:**
  - Support Threshold: A reasonable value based on dataset characteristics.
  - Confidence Threshold: Set a meaningful value to filter rules.
  - Lift: Evaluated to assess rule significance.
- **Steps:**
  - Preprocessed data by removing duplicates and inconsistencies.
  - Implemented Apriori using association rule mining tools in Weka.

### Step 1: Association Rule Mining (Apriori)

#### Preprocessing for Apriori

Select columns for Apriori

Region

Item Type

Order Priority

#### Processed Data for Apriori

	Order Priority	Region_Australia and Oceania	Region_Central America and the Caribbean	Region_Europe
0	1	0	0	1
1	1	0	0	1
2	1	0	0	1
3	0	0	0	1
4	1	0	0	1

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	levenshtein	cosine	dtw	manhattan	euclidean	cityblock	hamming	editdistance	normalized
1	frozenset('Item_Type_VeggieBurger')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
2	frozenset('Item_Type_CheeseBurger')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
3	frozenset('Region_Europe')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
4	frozenset('Item_Type_Salmon')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
5	frozenset('Item_Type_Salmon')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
6	frozenset('Region_Middle East and North Africa')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
7	frozenset('Item_Type_Meat')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
8	frozenset('Region_Australia and Oceania')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
9	frozenset('Item_Type_Personal Care')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
10	frozenset('Item_Type_Personal Care')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
11	frozenset('Region_South America and the Caribbean')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
12	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
13	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
14	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
15	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
16	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
17	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
18	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
19	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
20	frozenset('Item_Type_Soft Drink')	frozenset('Order Priority')	0.0008	0.7500	0.0049	0.7500	1.0000	0.0001	1.125	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

Business Insights

Key Insight: When frozenset({'Region\_Australia and Oceania'}), they are 75.0% likely to also frozenset({'Order Priority'}) (Lift = 1.00)

## Algorithm and Reasoning:

The Apriori algorithm is used to identify frequent itemsets in a dataset and generate association rules. It helps in discovering interesting relationships between variables in large databases.

### Parameters:

- **Support Threshold:** A minimum percentage of records in the dataset that contain the itemset. In our case, with 10,000 records, a support threshold of 0.05 means any itemset must appear in at least 500 transactions to be considered significant.

- **Confidence Threshold:** The likelihood that a rule is true for the dataset. A threshold of 0.5 means we are interested in rules where the likelihood is at least 50%.
- **Lift:** Measures the importance of a rule. Lift > 1 indicates a strong association.

Naïve Bayes (Classification)

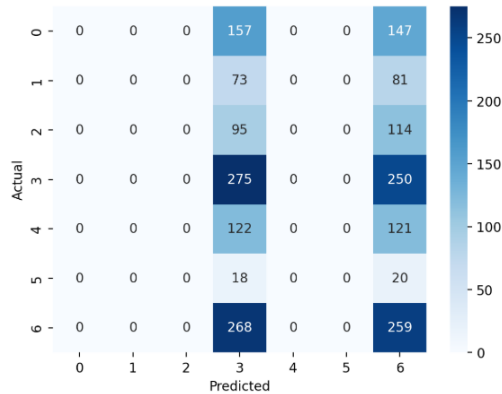
- **Objective:** Build a probabilistic model to classify data into predefined classes.
- **Selected Columns:** Region, Country, Item Type, Sales Channel, Order Priority.
- **Steps:**
  - Split dataset into training (70%) and testing (30%) sets.
  - Assumed feature independence.
  - Evaluated model using accuracy, precision, recall, and F1-score.

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
0	4	0	0	0	2	1,516	217,971,109	1,524	1,368	255.28	159.42	349,223.04	228,086.56	121,136.48
1	4	0	1	0	2	625	990,005,975	2,399	4,850	47.45	31.79	230,132.5	154,181.5	75,951
2	4	0	1	0	2	1,938	921,505,153	306	2,707	47.45	31.79	128,447.15	86,055.53	42,391.62
3	4	0	1	1	0	244	373,300,033	350	6,823	47.45	31.79	323,751.35	216,303.17	106,948.18
4	4	0	2	0	3	1,162	393,157,184	340	2,802	205.7	117.11	576,371.4	320,142.22	246,229.18

Results

Accuracy: 26.70%

Confusion Matrix

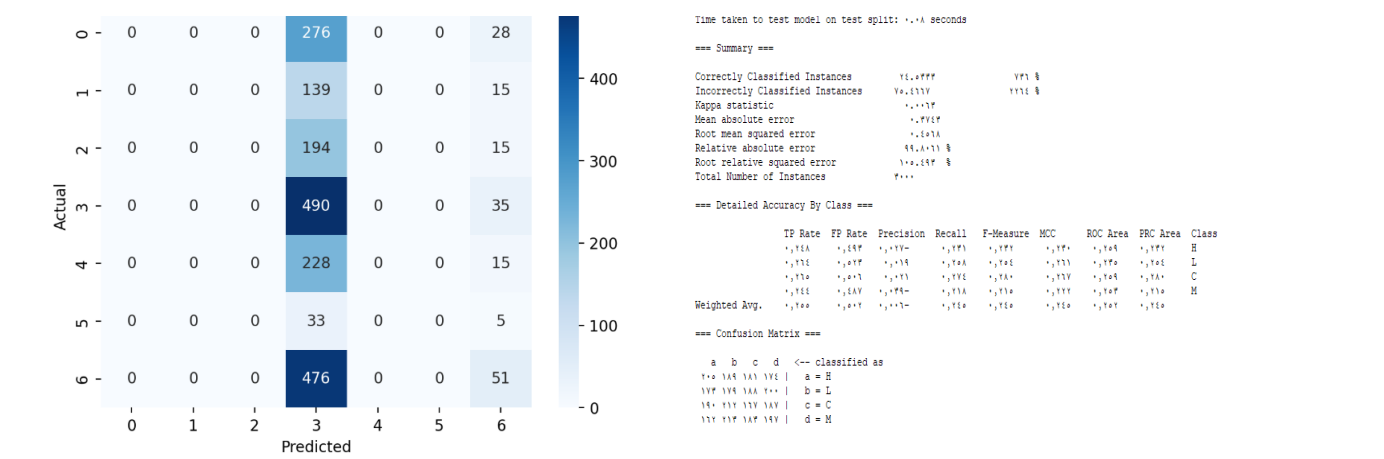


Classification Report

	precision	recall	f1-score	support
0	0	0	0	304
1	0	0	0	154
2	0	0	0	209
3	0.2762	0.5981	0.3779	525
4	0	0	0	243
5	0	0	0	38
6	0.2677	0.4383	0.3324	527
accuracy	0.2725	0.2725	0.2725	2,000
macro avg	0.0777	0.1481	0.1015	2,000
weighted av	0.143	0.2725	0.1868	2,000

☐ Show Prediction Probabilities

Total Cost				
mean	911155.491	949499.117	949899.187	949910.147
std. dev.	1110001.672	1111111.111	1111111.111	1110001.672
weight sum	1000	1000	1000	1000
precision	0.1111	0.1111	0.1111	0.1111
Total Profit				
mean	949911.111	949911.111	949911.111	949911.111
std. dev.	1111111.111	1111111.111	1111111.111	1111111.111
weight sum	1000	1000	1000	1000
precision	111.111	111.111	111.111	111.111
Time taken to build model:	0.11 seconds			
=== Evaluation on test split ===				
Time taken to test model on test split:	0.11 seconds			
=== Summary ===				
Correctly Classified Instances	11.1111	111.111		
Incorrectly Classified Instances	11.1111	111.111		
Macro statistic	11.111			
Mean absolute error	11.111			
Root mean squared error	11.111			
Relative absolute error	11.111			
Root relative squared error	11.111			
Total Number of Instances	1111			
=== Detailed Accuracy By Class ===				



Algorithm and Reasoning:

Naïve Bayes is a probabilistic classification algorithm based on Bayes' Theorem. It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Relevant Columns:

- **Region:** Useful for understanding geographic influence on purchasing behavior.
- **Country:** To identify differences between countries.
- **Item Type:** To know the types of products that are bought together.
- **Sales Channel:** To understand the differences between various sales channels (online and offline).
- **Order Priority:** To understand how order priority affects product relationships.

ID3 Algorithm (Decision Trees)

- **Objective:** Create decision trees based on information gain.
- **Selected Columns:** Region, Item Type, Sales Channel, Order Priority, Order Date.
- **Steps:**
  - Used entropy and information gain to construct the tree.
  - Visualized decision tree structure.
  - Evaluated accuracy using cross-validation.

K-Means Algorithm (Clustering)

- **Objective:** Partition the data into clusters based on similarity.
- **Selected Columns:** Units Sold, Unit Price, Total Revenue, Total Profit.
- **Parameters:**
  - Number of Clusters (K): Determined using the elbow method.
  - Initialization: Used k-means++ to enhance convergence.
- **Steps:**
  - Standardized the data.
  - Applied K-Means clustering algorithm.
  - Visualized clusters and centroids.

Processed Data for K-Means

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Un
0	4	0	0	0	2	1,516	217,971,169	1,524	
1	4	0	1	0	2	625	990,685,975	2,399	
2	4	0	1	0	2	1,938	921,505,153	386	
3	4	0	1	1	0	244	373,300,683	350	
4	4	0	2	0	3	1,662	393,157,184	348	

Select Feature Columns

Units Sold ×

Unit Price ×

Total Revenue ×

Total Profit ×

⊞

▼

Initial starting points (random):  
Cluster 0: 7712,316,17,11111111,73111111,11,11111111  
Cluster 1: 1111,111,11,11111111,11,11111111,11,11111111  
Missing values globally replaced with mean/mode  
Final cluster centroids:  
Attribute      Full Data      Cluster#  
                 (10000.0)      (2470.0)      (7530.0)  
-----  
Units Sold      1477.4000      1174.1111      8112.8000  
Unit Price      111.1111      111.1111      111.1111  
Total Revenue      164111.1111      128111.1111      899999.9999  
Total Cost      111111.1111      111111.1111      111111.1111  
Total Profit      53000.0000      17000.0000      788888.8888  
  
Time taken to build model (full training data) : 0.00 seconds  
=== Model and evaluation on training set ===  
Clustered Instances  
(0%) 1111  
(1%) 1111

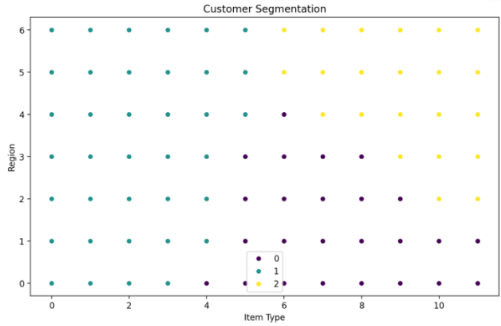
Silhouette Score: 0.60

Evaluation Metrics

Feature Importance

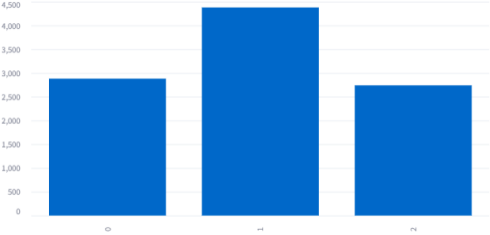


Cluster Visualization



Results and Insights

Cluster Distribution



Decision Rules

