

**TITLE: CUSTOMER SALES ANALYSIS: LEVERAGING MACHINE LEARNING AND TABLEAU**

**AUTHOR**

**Charlotte Olivia Namagembe**

**Module Name: Data Science Internship Project**

**Module Leader: Mr. Vidya Sagar**

[**TABLE OF CONTENTS** 2](#_Toc173428989)

[LIST OF FIGURES 3](#_Toc173428990)

[LIST OF TABLES 3](#_Toc173428991)

[LIST OF ABBREVIATIONS 4](#_Toc173428992)

[**CRISP-DM LIFECYCLE DOCUMENTATION FOR A RETAIL DATASET** 5](#_Toc173428993)

[**1.0 PHASE ONE: BUSINESS UNDERSTANDING** 5](#_Toc173428994)

[1.1 BACKGROUND OF THE PROJECT (CUSTOMER RETAIL ANALYSIS) 5](#_Toc173428995)

[1.2 MOTIVATION 6](#_Toc173428996)

[1.3 USE CASES: 6](#_Toc173428997)

[1.3.1 Business Context: 6](#_Toc173428998)

[1.3.2 Success Criteria: 6](#_Toc173428999)

[1.4 Business Questions 6](#_Toc173429000)

[1.4.1 Primary Question of the Project: 6](#_Toc173429001)

[1.4.2 Secondary Questions: 6](#_Toc173429002)

[1.4.3 Project Goals: 7](#_Toc173429003)

[**2.0 PHASE TWO: DATA UNDERSTANDING** 7](#_Toc173429004)

[2.1 Data Collection: 7](#_Toc173429005)

[2.1.1 Data Source: 7](#_Toc173429006)

[2.1.2 Data Acquisition: 7](#_Toc173429007)

[2.2 Data Inventory: 7](#_Toc173429008)

[2.2.1 Initial Data Exploration: 7](#_Toc173429009)

[2.2.2 Summary Statistics: 7](#_Toc173429010)

[2.2.3 Data Quality Issues: 7](#_Toc173429011)

[2.2.4 ABOUT THE DATASET 8](#_Toc173429012)

[**3.0 PHASE THREE: DATA PREPARATION** 8](#_Toc173429013)

[3.1 Importing Libraries 8](#_Toc173429014)

[3.2 Loading Data 9](#_Toc173429015)

[3.3 Checking all Necessary Insights 9](#_Toc173429016)

[3.3 Cleaning and Handling Missing Values 12](#_Toc173429017)

[3.3.1 Handling Missing Values 12](#_Toc173429018)

[3.3.2 Dropping Columns 13](#_Toc173429019)

[3.4 Feature Engineering 13](#_Toc173429020)

[3.5 Exploratory Data Analysis 13](#_Toc173429021)

[3.5.1 Plotting to check the most purchased products 13](#_Toc173429022)

[3.5.2 Plotting to check which season with what product is purchased more 13](#_Toc173429023)

[3.5.3 Checking for Outliers 13](#_Toc173429024)

[3.5.4 Use of the Shapiro Test and Normal Distribution 13](#_Toc173429025)

[3.5.5 Correlation of the Features 13](#_Toc173429026)

[3.6 Data Type Conversion 14](#_Toc173429027)

[**4.0 PHASE FOUR: MODELLING** 14](#_Toc173429028)

[4.1 Model Building 14](#_Toc173429029)

[4.2 Model Selection 14](#_Toc173429030)

[4.2.0 KNN Algorithm 14](#_Toc173429031)

[4.2.1 SVC Algorithm 15](#_Toc173429032)

[4.2.2 Naïve Bayes Algorithm 16](#_Toc173429033)

[4..2.3 Decision Tree Classifier 16](#_Toc173429034)

[4.3 Model Evaluation 17](#_Toc173429035)

[**5.0 Evaluation** 18](#_Toc173429036)

[5.1 Model Performance Metrics 18](#_Toc173429037)

[5.2Evaluation Results: 19](#_Toc173429038)

[**6.0 PHASE SIX: DEPLOYMENT** 19](#_Toc173429039)

[6.1 Deployment Plan: 19](#_Toc173429040)

[6.2 Monitoring and Maintenance: 19](#_Toc173429041)

[6.3 Documentation and Reporting: 20](#_Toc173429042)

[6.3.1 Key Performance Indicators: (KPIs) Requirements 20](#_Toc173429043)

[6.3.2 Charts 20](#_Toc173429044)

[7.0 RECOMMENDATIONS 21](#_Toc173429045)

# LIST OF FIGURES

[Figure 1: Phases of the CRISP-DM 5](#_Toc173424877)

[Figure 2: Initializing KNN Algorithm 14](#_Toc173424878)

[Figure 3: Initializing LinearSVC Algorithm 15](#_Toc173424879)

[Figure 4: Initializing Naive Bayes Algorithm 15](#_Toc173424880)

[Figure 5: Initializing Decision Tree Algorithm 16](#_Toc173424881)

[Figure 6: Outlook of the Dashboard 19](#_Toc173424882)

# LIST OF TABLES

[Table 1: Evaluation Metrics and Formulas 16](#_Toc173424883)

[Table 2: Confusion Matrix 16](#_Toc173424884)

[Table 3: Model Performance Metrics 17](#_Toc173424885)

[Table 4: Evaluation Results 17](#_Toc173424886)

# LIST OF ABBREVIATIONS

CRISP-DM Cross-Industry Standard Process for Data Mining

EDA Explanatory Data Analysis

TP True Positive

FN False Negative

TN True Negative

FP False Positive

KNN K-Nearest Neighbour

SVC Support Vector Classifier

# **CRISP-DM LIFECYCLE DOCUMENTATION FOR A RETAIL DATASET**

CRISP-DM is a process model that describes common approaches used by data mining experts. It is an open standard and the most widely-used analytics model. This document contains models that describes the lifecycle of a data science project that is intended to solve a retail business problem. This project is broken into 6 distinct phases starting from the inception of the project all the way to its final impact. The project runs on a CRISP-DM (Cross-Industry Standard Process for Data Mining) lifecycle. The key phases for the project (Figure 1): Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment. In the documentation below, you will find detailed information on each of the phases, practical information (code implementation), important considerations, ethical and moral policies to be undertaken.

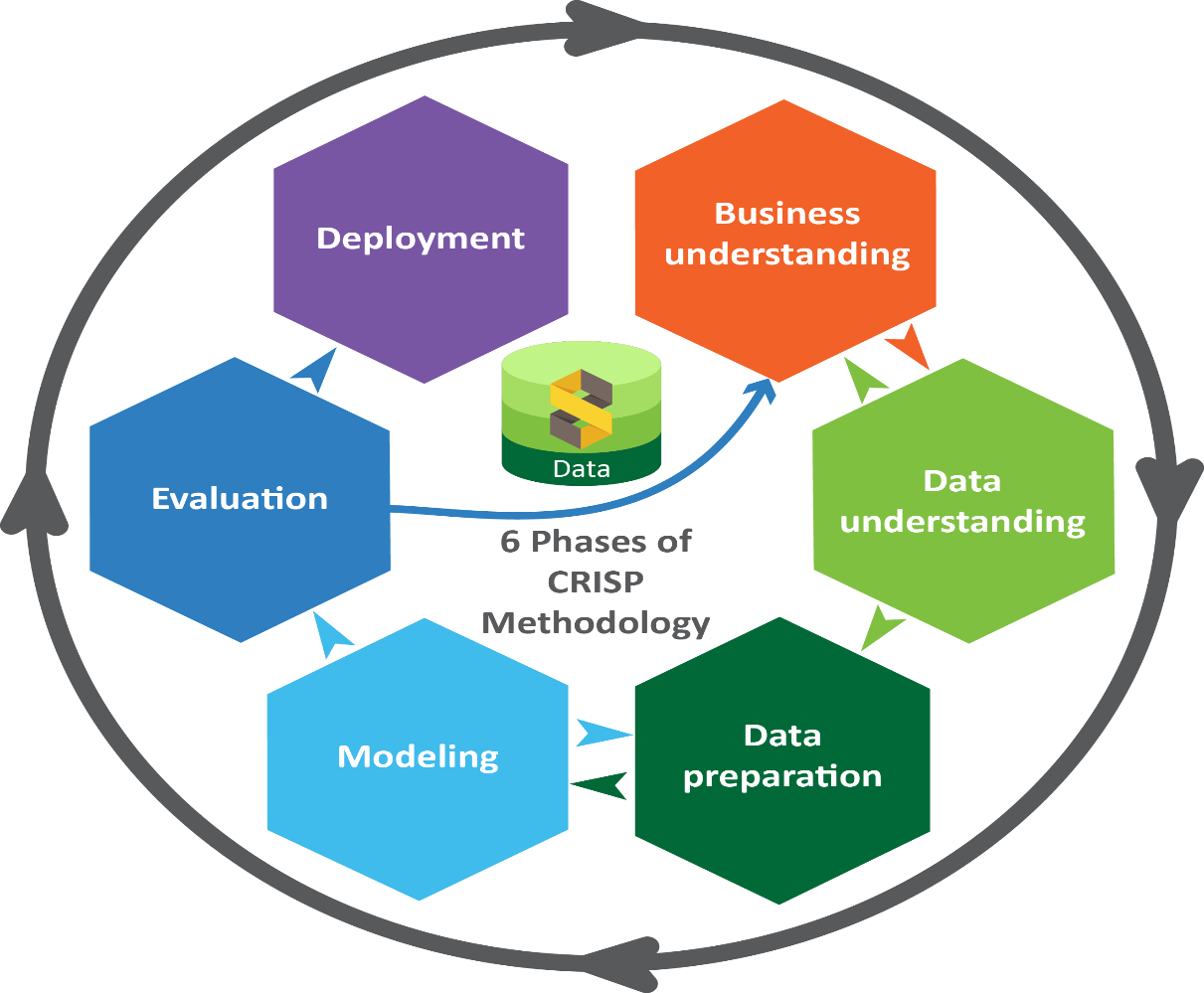


Figure 1: Phases of the CRISP-DM

# **1.0 PHASE ONE: BUSINESS UNDERSTANDING**

## 1.1 BACKGROUND OF THE PROJECT (CUSTOMER RETAIL ANALYSIS)

The niche to find the correlation on how different customers are purchasing products is very crucial. Retail businesses, from supermarkets to convenience stores, are constantly seeking ways to better understand their customers and improve their operations. The gap in recommending how customers are linked in a way that if a customer is buying a certain product, what is the affiliation with the different products is an interesting question to answer.

Market basket analysis: Is a data mining technique used in retail analytics, explores customer purchase patterns to uncover associations between products, identify trends, and optimize pricing and promotions. It works by looking for combinations of items that occur frequently in transactions. For example, if customers are buying milk, how probably are they to also buy bread (and which kind of bread) on the same trip to the supermarket? This information may lead to an increase in sales by helping retailers to do selective marketing based on predictions, cross-selling and planning their ledge space for optimal product.

Customer segmentation: Is the practice of categorizing customers into distinct groups based on characteristics such as; demographics, behaviour and preferences.

Pricing Optimization: Retail businesses are optimizing price strategies and identifying opportunities for discounts and promotions. Identification of optimal price point for any given product at any given location that will yield the highest profit.

## 1.2 MOTIVATION

The motivation for this dataset comes from the need for accessible and customizable market basket datasets. While real-world retail data is sensitive and often restricted, synthetic datasets offer a safe and versatile alternative. Researchers, data scientists, and analysts can use this dataset to develop and test algorithms, models, and analytical tools.

## 1.3 USE CASES:

Market Basket Analysis: Discover associations between products and uncover buying patterns.

Customer Segmentation: Group customers based on purchasing behaviour.

Pricing Optimization: Optimize pricing strategies and identify opportunities for discounts and promotions.

Retail Analytics: Analyse store performance and customer trends.

### 1.3.1 Business Context:

The project is important for the business because they would like to have a better insight that can revolutionize marketing strategies and ultimately enhance customer satisfaction and loyalty.

### 1.3.2 Success Criteria:

The success will be measured by increased customer loyalty and revenue growth.

## 1.4 Business Questions

### 1.4.1 Primary Question of the Project:

The primary question of this project is to predict customer Gender based on their shopping behaviours and other relevant features present in the dataset.

### 1.4.2 Secondary Questions:

* What are the most popular products?
* Which products are often purchased together?
* In which season do the customers tend to purchase most?
* How can we identify and target high-value customers?
* Which features will influence the customers the most and less when they are in the supermarket?

### 1.4.3 Project Goals:

* Develop a recommendation system to suggest products to customers.
* Segment customers to tailor marketing strategies.
* Predict sales trends to optimize inventory management.

# **2.0 PHASE TWO: DATA UNDERSTANDING**

Acquiring, storing, linking, understanding data for analysis on a project often entails an involved and iterative process, requiring working closely with the team to work on the choice of the dataset to ensure that the dataset collected and chosen answers the business objectives. Consistency and reliability in the data format was essential to ensure protection of private and sensitive information. During this phase of the work, the team was required to apply skills working with and structuring data to get it into a storage format that is appropriate for linking it with the data sources. Each of the steps require active communication with the project’s stakeholders to understand the context in which the data were collected and structured to ensure data definitions actually describe the events they are supposed to reflect.

## 2.1 Data Collection:

## 2.1.1 Data Source:

The data was collected through team work. We used Kaggle, which is a platform for finding datasets, participate in data science competitions, collaborate with other data scientists etc.

The title of our dataset is: Retail Transaction Dataset

The link to access this dataset:

https://www.kaggle.com/datasets/prasad22/retail-transactions-dataset/data

Data Files: Retail Transactions Dataset.csv

2.1.2 Data Acquisition:

The data was acquired because this was our task during our Data Science internship with London School of Informatics.

## 2.2 Data Inventory:

The Retail Transaction Dataset which we have collected was created to simulate a market basket dataset, providing insights into customer purchasing behaviour and store operations. The dataset facilitates market basket analysis, customer segmentation, and other retail analytics tasks.

### 2.2.1 Initial Data Exploration:

Retail Transactions Dataset.csv: Transaction\_ID, Customer Name, Date, Product, Total\_Items, Total\_Cost, Payment\_Method, City, Store\_Type, Discount\_Applied, Customer\_Category, Promotion

### 2.2.2 Summary Statistics:

Number of transactions/rows: 100,000, 000

Number of columns: 13

### 2.2.3 Data Quality Issues:

* In our dataset we did not have any outliers or duplicates.
* Missing values in Promotion column.
* Inconsistent customer names.

### 2.2.4 ABOUT THE DATASET

This dataset was created to simulate a market basket dataset, providing insights into customer purchasing behaviour and store operations. The dataset facilitates market basket analysis, customer segmentation, and other retail analytics tasks.

#### 2.2.4.1 Dataset Information (Attributes and Variables):

The columns provide information about the transactions, customers, products, and purchasing behaviour, making the dataset suitable for various analysis, including market basket analysis and customer segmentation. Here's a brief explanation of each column in the Dataset:

* Transaction\_ID: A unique identifier for each transaction, represented as a 10-digit number. This column is used to uniquely identify each purchase.
* Date: The date and time when the transaction occurred. It records the timestamp of each purchase.
* Customer\_Name: The name of the customer who made the purchase. It provides information about the customer's identity.
* Product: A list of products purchased in the transaction. It includes the names of the products bought.
* Total\_Items: The total number of items purchased in the transaction. It represents the quantity of products bought.
* Total\_Cost: The total cost of the purchase, in currency. It represents the financial value of the transaction.
* Payment\_Method: The method used for payment in the transaction, such as credit card, debit card, cash, or mobile payment.
* City: The city where the purchase took place. It indicates the location of the transaction.
* Store\_Type: The type of store where the purchase was made, such as a supermarket, convenience store, department store, etc.
* Discount\_Applied: A binary indicator (True/False) representing whether a discount was applied to the transaction.
* Customer\_Category: A category representing the customer's background or age group.
* Season: The season in which the purchase occurred, such as spring, summer, fall, or winter.
* Promotion: The type of promotion applied to the transaction, such as "None," "BOGO (Buy One Get One)," or "Discount on Selected Items."

NB: The dataset has 2 numerical columns and more categorical ones.

# **3.0 PHASE THREE: DATA PREPARATION**

At this phase (also be referred to as "data munging"), the data is prepared for modelling. It has the following tasks and more as the data demands.

## 3.1 Importing Libraries

We Imported all the relevant libraries that will be useful for our analysis. The dataset was then loaded into the pandas.

* Numpy: use for working with arrays.
* Pandas: use for working with data sets. (functions: Analysing, Cleaning, Exploring and Manipulating data across the Data Frame (isna() and .isnull())
* Matplotlib: for creating visualizations such as Plots, Charts, and Graphs.
* Seaborn - data visualisation tool - help us to see the graph on the output
* Statistical: built-in module that provides functions for mathematical.
* (statistical measures such as mean, median, mode, standard deviation, variance, and more)

import numpy as np

import pandas as pd

import matplotlib as mlp

import matplotlib.pyplot as plt

import seaborn as sns

import statistics

## 3.2 Loading Data

The dataset was read and loaded into Spyder environment.

df = pd.read\_csv(r'C:\Users\elyns\Desktop\Retail\_Transactions\_Dataset (1).csv')

print(df)

## 3.3 Checking all Necessary Insights

We explored the data further: by check the first few rows of the dataset to understand its structure and contents. You can always specify any number of rows you want. In our case, we wanted to get the first 5 rows of the 13 columns.

df. head(5)

Transaction\_ID Date Season Promotion

0 1000000000 21/01/2022 06:27 ... Winter NaN

1 1000000001 01/03/2023 13:01 ... Fall BOGO (Buy One Get One)

2 1000000002 21/03/2024 15:37 ... Winter NaN

3 1000000003 31/10/2020 09:59 ... Spring NaN

4 1000000004 10/12/2020 00:59 ... Winter Discount on Selected Items

[5 rows x 13 columns]

We describe the Table Information by initiating df.describe() for Data Count/Mean/STD/Min/%

print(df.describe())

Transaction\_ID Total\_Items Total\_Cost

count 1.000000e+06 1000000.000000 1000000.000000

mean 1.000500e+09 5.495941 52.455220

std 2.886753e+05 2.871654 27.416989

min 1.000000e+09 1.000000 5.000000

25% 1.000250e+09 3.000000 28.710000

50% 1.000500e+09 5.000000 52.420000

75% 1.000750e+09 8.000000 76.190000

max 1.001000e+09 10.000000 100.000000

Interpretation: The returned columns are the numerical values in our dataset and the values show the central tendencies measures.

Check any items on the column for more information - example Payment\_Method)

pd.isna(df.Payment\_Method)

pd.isna(df.Payment\_Method)

Out[186]:

0 False

1 False

2 False

3 False

4 False

999995 False

999996 False

999997 False

999998 False

999999 False

Name: Payment\_Method, Length: 1000000, dtype: bool

We went further to explore an individual parameters for indepth Information. Only few output we will display here.

df.Customer\_Category.value\_counts()

df.City.value\_counts()

df.Product.value\_counts()

df.Store\_Type.value\_counts()

df.Payment\_Method.value\_counts()

df.Season.value\_counts()

df.Discount\_Applied.value\_counts()

df.Promotion.value\_counts()

df.Total\_Cost.value\_counts()

The output below corresponds to the actual dataset or orders made by customers.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  | | df.Customer\_Category.value\_counts()  Out[188]:  Customer\_Category  Senior Citizen 125485  Homemaker 125418  Teenager 125319  Retiree 125072  Student 124842  Professional 124651  Middle-Aged 124636  Young Adult 124577  Name: count, dtype: int64 |  | |  |  | |
| df.City.value\_counts()  Out[189]:  City  Boston 100566  Dallas 100559  Seattle 100167  Chicago 100059  Houston 100050  New York 100007  Los Angeles 99879  Miami 99839  San Francisco 99808  Atlanta 99066  Name: count, dtype: int64 |
|  |
| df.Season.value\_counts()  Out[193]:  Season  Spring 250368  Fall 250248  Winter 249763  Summer 249621  Name: count, dtype: int64 |

The info() method provides a summary of the data frame, including the number of non-null entries and data types for each column.

df.info()

Checking the Dimensions in the dataset

Shape is used to retrieve the dimensions of a dataset. It provides a tuple representing the number of rows and columns in the dataset. This information is crucial for understanding the structure of your data, performing data validation, and preparing data for further analysis or modelling.

df.shape

Result: We have 1000000 columns and 13 rows

Understand Data Types and Missing Values

Data types: Check the data types of each column.

print(df.dtypes)

Result: The dataset has different types with;int, object, float and boolean

Checking for Duplicate Values

Checking for duplicate values is a critical step in data preprocessing. It helps maintain data integrity, ensures statistical accuracy, improves model performance, and enhances resource efficiency.

## 3.3 Cleaning and Handling Missing Values

Clean data: Here, you handle missing values by either imputing or removal.

The column of product was cleaned to remove square brackets and quotations within the dataset.

NB: We can see that the promotion column has missing values with a total sum of 333943 counts. They are various ways to decide how to handle missing values: drop, fill, or other imputation methods.

### 3.3.1 Handling Missing Values

* Drop/Delete the missing values is not efficient/ suitable if the dataset is big.
* The imputation method is when we find the mean/mode value for that column in the dataset and then replace it with the same value.
* Forward fill or backward fill: Use the adjacent values to fill missing entries.
* Fill the missing values with specific words like 'unknown'.
* Delete the rows with the missing the values.

NB: In this dataset, we are going to use the fourth option as the missing values are many. This is also because the Promotion column is of object data type and does not have a statistical summary of mean, mode or standard deviation.

* Promotion: This column had missing values but it had about 333,943 which is 33% of the

promotion column. Here as the team, we replaced the missing values with a specific word ‘unknown’. This is because also the Promotion column had ‘string’ with object as a data type.

### 3.3.2 Dropping Columns

* Transaction\_ID was dropped from the dataset as it did not serve any purpose on the performance of the analysis.
* Customer\_Name: This helped in feature engineering of the gender column, after use, it was dropped.
* Date: This column was dropped after extracting the month, year and date and later it was dropped.

## 3.4 Feature Engineering

This involves the creation of new features recommended by from subject matter experts and as needed

* Gender: Using the gender\_guessor library, the team dropped the last names, maintained the first name that generated the identification of the females and males in the dataset.
* First, we imported the gender\_guessor library.
* Then we wrote a function to extract the first name.
* Next, we applied the function to the 'FullName' column to create a new 'FirstName' column.
* Drop the 'FullName' column if you only want to keep the first names.
* Initialize the gender detector.
* Function to detect gender.
* Apply the gender detection function to the 'Name' column and create a new column 'Gender'.
* On printing the values, we noticed that the gender column did have unknown values, mostly female and mostly values, these were strictly categorized into female and male as gender is only female or male.
* Date: Extraction of date, month and year from the date column.

## 3.5 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical step in the data analysis process. It involves summarizing and visualizing the main characteristics of a dataset, often with the help of graphical representations. Analysis can be done in three ways; Univariate, Bi-Variate and Multi-Variate.

* Univariate Analysis: Analyse individual columns to show distribution, central tendency and dispersion. The possible plots include; Histograms, Box Plots, Value Counts.
* Bi-Variate Analysis: Analyse relationships between pairs of variables. The possible plots include; Scatter Plots, Correlation Matrix and Heat Map, Box Plot (Continuous vs Categorical Data)
* Multi-Variate Analysis: Analyse relationships between more than 2 variables. The possible plots include; Pair Plots, Facet Plots, Swarm Plots.

NB: Different Visualizations took place to check how different parameters flow through.

### 3.5.1 Plotting to check the most purchased products

Tooth paste is the most purchased product among all the products

### 3.5.2 Plotting to check which season with what product is purchased more

### 3.5.3 Checking for Outliers

### 3.5.4 Use of the Shapiro Test and Normal Distribution

### 3.5.5 Correlation of the Features

## 3.6 Data Type Conversion

Formatting and converting values in data to acceptable format (e.g. strings/categorical to numeric) by machine learning model.

* Store\_Type: Using the label encoder, this column was encoded into numeric.
* Customer\_Category: Using the label encoder, this column was encoded into numeric.
* Payment\_Method: Using the label encoder, this column was encoded into numeric.
* Promotion: Using the label encoder, this column was encoded into numeric.
* Discount\_Applied: Using the hot encoder, the column data type was converted into numeric
* Gender: Using the label encoder, this column was converted into category and later label encoded.

V. Final Dataset: Product, Total\_Items, Total\_Cost, Payment\_Method, City, Store\_Type, Discount\_Applied, Customer\_Category, Season, Promotion, Gender.

# **4.0 PHASE FOUR: MODELLING**

## 4.1 Model Building

Tools and Libraries: The library I used for modelling is scikit-learn.

Model Training and Validation: The data was divided into 2 parts, testing set (20%) and training set (80%). We used the train set to teach the model and test set for the validation of the model.

Our target variable is gender and the features are the rest of the variables in the dataset. This implies that the gender takes up the 20% and the rest of the variables are the 80%.

## 4.2 Model Selection

Modelling Techniques: We used different mused K-Nearest Neighbors, Support Vector Classifier, Naive Bayes and Decision Tree Classifier.

Justification: The used and selected algorithms were the suggested models by the stakeholders (London School Informatics).

### 4.2.0 KNN Algorithm

The K-Nearest Neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used for both classification and regression tasks. It is based on the principle that similar data points are close to each other in feature space. Here is an overview of how KNN works:

#### 4.2.1 How KNN Algorithm Works

* Choosing the Value of K:

The number of nearest neighbors (K) is chosen. This parameter determines how many neighbors will be used to make a prediction.

* Calculating Distance:

The algorithm calculates the distance between the data point (whose class or value needs to be predicted) and all other points in the training dataset. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance.

* Finding Nearest Neighbors:

The K data points in the training set that are closest to the data point in question are identified as the nearest neighbors.

* Making Predictions:
* Classification Choice:

The algorithm counts the number of data points from each class among the K nearest neighbors. The class with the highest count is assigned to the data point.

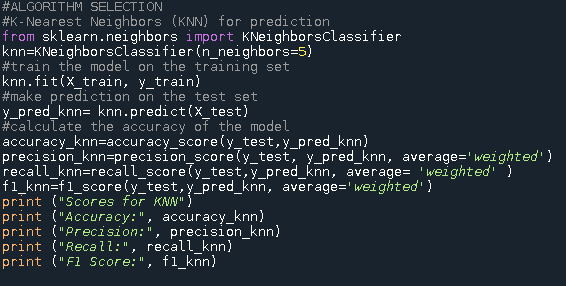


Figure 2: Initializing KNN Algorithm

### 4.2.1 SVC Algorithm

The Support Vector Classification (SVC) algorithm is a supervised machine learning algorithm used for classification tasks. It is part of the Support Vector Machine (SVM) family of algorithms and is particularly effective in high-dimensional spaces. Here are the key concepts and details about the SVC algorithm.

#### 4.2.1.1 Flow of the Algorithm

* Hyperplane:

SVC aims to find the optimal hyperplane that best separates the data points of different classes. In a two-dimensional space, this hyperplane is a line, but in higher dimensions, it becomes a plane or a hyperplane.

* Support Vectors:

Support vectors are the data points that are closest to the hyperplane. These points are crucial for defining the position and orientation of the hyperplane. The algorithm optimizes the hyperplane based on these support vectors.

* Margin:

The margin is the distance between the hyperplane and the nearest data points from each class. SVC seeks to maximize this margin, making the classification more robust.

* Kernel Trick:

When the data is not linearly separable, SVC uses the kernel trick to transform the data into a higher-dimensional space where it becomes linearly separable. Common kernels include linear, polynomial, radial basis function (RBF), and sigmoid.

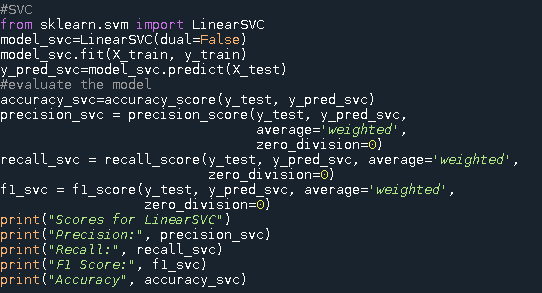


Figure 3: Initializing LinearSVC Algorithm

### 4.2.2 Naïve Bayes Algorithm

Naive Bayes is a probabilistic machine learning algorithm based on Bayes' Theorem with an assumption of independence among features. It's particularly known for its simplicity, efficiency, and effectiveness for a variety of tasks such as text classification and spam detection.

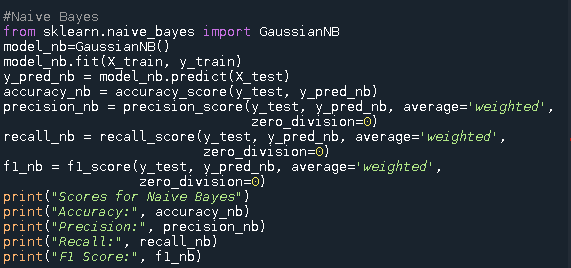


Figure 4: Initializing Naive Bayes Algorithm

### 4..2.3 Decision Tree Classifier

A Decision Tree Classifier is a machine learning algorithm used for classification tasks. It works by splitting the data into subsets based on the value of input features, creating a tree-like model of decisions. Here is a detailed breakdown of how a Decision Tree Classifier works:

#### 4.2.1. Flow of the Algorithm

* Selection of its needs. A Decision Tree consists of:
* Nodes: Where the data is split based on a feature.
* Edges: The outcome of a split, connecting one node to another.
* Leaves: Terminal nodes that represent a class label or a decision.
* Splitting the Data

The core idea is to split the dataset into subsets that are more homogeneous (i.e., contain a higher proportion of a single class). The algorithm starts at the root node and splits the data based on feature values to maximize the homogeneity of the resulting subsets.

* Choosing the Best Split
* Stopping Criteria
* The splitting continues until a stopping criterion is met, such as:
* All instances in a node belong to the same class.
* The maximum depth of the tree is reached.
* The number of instances in a node is less than a specified minimum.
* Prediction

To classify a new instance, the tree is traversed from the root node to a leaf node, following the splits based on the feature values of the instance. The class label of the leaf node is assigned to the instance.

* Pruning

To prevent overfitting, pruning techniques can be applied to reduce the size of the tree:

* Pre-pruning (Early Stopping): Stop the tree growth early based on certain criteria (e.g., maximum depth, minimum samples per node).
* Post-pruning: Remove nodes that provide little power to classify instances.

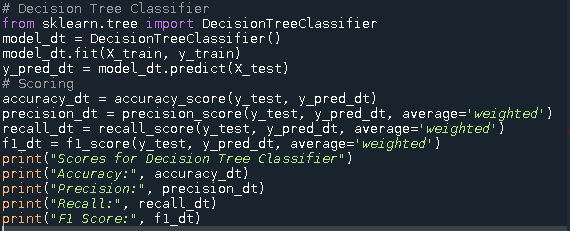


Figure 5: Initializing Decision Tree Algorithm

## 4.3 Model Evaluation

Evaluation Metrics: The metrics were used to evaluate model performance are: accuracy, F1 score, precision, recall. One of the crucial tasks in the application of machine learning is evaluating the execution of procedures. A precise model that can forecast the previously unobserved data must be developed. The model might perform well by some metrics, but by others, its strength might perform poorly. Therefore, it is crucial to use a variety of evaluation indicators to rate the model. The evaluation metrics are displayed in Table 1.

Table 1: Evaluation Metrics and Formulas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Evaluation Metrics | Accuracy | Precision | Recall | F1 Score |
| Formulae |  |  |  |  |

A table with four distinct sections that call for actual and anticipated standards is known as a confusion matrix. When a classification model is applied to a collection of test data, it is used to explain how the true values for those test data came to be. Whether the data is accurately categorised or not may be quickly determined thanks to this. Comparison of expected and actual values must be done. By determining how many data records are correctly and incorrectly classified by the model, it assigns a performance vision. The totals of true positives, true negatives, and false positives are listed in two rows and two columns, respectively.

Table 2: Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Gender Detected/Found | No Gender Found |
|  | Gender Detected/Found | TP (True Positive) | FP (False Negative) |
| Ground Truth | No Gender Found | FN (False Negative) | TN (True Negative) |

# **5.0 Evaluation**

## 5.1 Model Performance Metrics

Comparison: The performance is not very good for any of the models, the Linear SVC and Naive Bayes are very poor in precision and F1 Score. KNN and Decision Tree Classifier did a little bit better, but still got very low scores, see the result below.

Table 3: Model Performance Metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **KNN** | **LinearSVC** | **Naives Bayes Classifier** | **Decision Tree Classifier** |
| Accuracy | 0.5 | 0.5 | 0.5 | 0.5 |
| Precision | 0.5 | 0.25 | 0.25 | 0.5 |
| Recall | 0.5 | 0.5 | 0.5 | 0.5 |
| F1-Score | 0.5 | 0.33 | 0.33 | 0.5 |

During the analysis, before using the models I used feature scoring. It helps us to understand the importance of different features in our model, it tells us which features have more influence on the predictions. In our project I highlighted with yellow the column names which got the best scores.

Accuracy: High accuracy indicates that the model correctly classifies a high proportion of instances. However, it is not always the best metric for imbalanced datasets.

Precision and Recall: High precision and recall are indicators of a good balance between false positives and false negatives. Retail customer analysis, recall might be more important if missing a positive case (e.g., identifying a loyal customer) is costly.

F1-Score: A balanced metric that is useful when you need to take both precision and recall into account.

**NB: KNN Algorithm performed better than other algorithms.**

## 5.2Evaluation Results:

Promotion as a feature has the best performing score as seen in Table 4 with the use of the algorithms. Total\_Items has the lowest performing score among all the features.

This gives an insight that there is a correlation on how customers tend to purchase products if given a promotion.

The city feature depends on the location for the sale of the products and the purchasing power of customers.

Table 4: Evaluation Results

|  |  |
| --- | --- |
| **Feature** | **Score** |
| Promotion | 1.785875 |
| City | 1.630015 |
| Product | 1.334177 |
| Total\_Cost | 1.206675 |
| Payment\_Method | 1.040041 |
| Store\_Type | 0.230836 |
| Discount\_Applied | 0.198195 |
| Season | 0.072142 |
| Customer\_Category | 0.012379 |
| Total\_Items | 0.006976 |

# **6.0 PHASE SIX: DEPLOYMENT**

## 6.1 Deployment Plan:

* Integrate the recommendation system into the retail website/app.
* Use customer segments to tailor marketing campaigns.
* Implement sales prediction model to optimize inventory management.

## 6.2 Monitoring and Maintenance:

* Continuously monitor model performance and update models with new data.
* Gather feedback from users to improve recommendation system.
* Regularly retrain models to adapt to changing customer behaviour and market trends.

## 6.3 Documentation and Reporting:

We used Tableau Software to design a dashboard for reporting track the progress of our business.

### 6.3.1 Key Performance Indicators: (KPIs) Requirements

**1.Total Items:**

* Understand the overall landscape of items sold to assess the market's size and growth.

**2.Total Cost:**

* Determine the cost of the products within per transaction giving a snippet how price of items is encompassed.

**3.Total Sales:**

* Identify and analyse the total sales of in the dataset.
* Calculate the percentage of total sales relative to the total number of items and the total cost.

**4. Average Sales:**

* Identify and analyse the total number of sales in the dataset with the whole totality in the dataset.

### 6.3.2 Charts

1. **Total Cost by Year:**

* Visualization: Line/ Area Chart
* Description: This chart will illustrate the distribution of cost over the years, starting from 2020 to 2024, providing insights into the growth pattern and adoption trends.

NB: We had to extract the year from the date column in the dataset

**2. Total Sales by City:**

* Visualization: Map Chart
* Description: This chart will showcase the geographical distribution of total sales across different states, allowing for the identification of regions with higher adoption rates.

**3. Top 10 Total Sales by Product:**

* Visualization: Bar Chart
* Description: Highlight the top 10 – 15 products based on the total number of sales, providing insights into the market dominance of specific brands.

**4. Total Sales by Gender:**

* Visualization: Pie Chart or Donut Chart
* Description: Illustrate the proportion of gender to provide incentives, aiding in understanding the impact of incentives on sales adoption.

**5. Top Sales by Payment Method:**

* Visualization: Tree map
* Description: Highlight the payment method based on the total number of sales, offering insights into consumer preferences and popular promotion in the market.

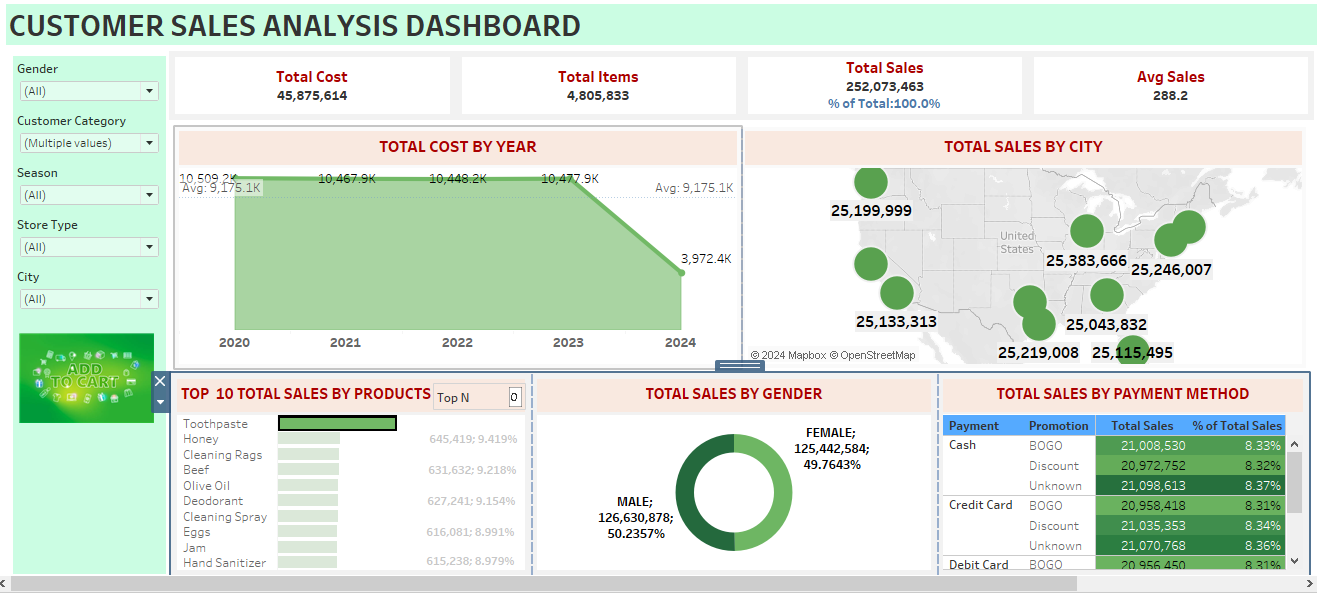


Figure 6: Outlook of the Dashboard

# 7.0 RECOMMENDATIONS