# SEASONS OF SHOPPING: ANALYZING THE INFUENCE OF SEASONALITY ON CUSTOMER PURCHASE DECISIONS

*Minor project-II report submitted*
*in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology**
**in**
**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**By**


| | | |
|---|---|---|
| **LINGIREDDY SRINIVASA REDDY** | (21UEAD0504) | **(VTU24112)** |
| **GUVVALA NANDINI** | (21UEAD0023) | **(VTU20477)** |
| **AVULA BHANUPRAKASH REDDY** | (21UEAD0003) | **(VTU20555)** |

*Under the guidance of*
*Mrs. FARZHANA, M.E.,*
*ASSISTANT PROFESSOR*

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)**
**Accredited by NAAC with A++ Grade**
**CHENNAI 600 062, TAMILNADU, INDIA**


**May, 2024**

# SEASONS OF SHOPPING: ANALYZING THE INFUENCE OF SEASONALITY ON CUSTOMER PURCHASE DECISIONS

*Minor project-II report submitted*
*in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology**
**in**
**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**By**



| | | |
|---|---|---|
| **LINGIREDDY SRINIVASA REDDY** | (21UEAD0504) | **(VTU24112)** |
| **GUVVALA NANDINI** | (21UEAD0023) | **(VTU20477)** |
| **AVULA BHANUPRAKASH REDDY** | (21UEAD0003) | **(VTU20555)** |

*Under the guidance of*
*Mrs. FARZHANA, M.E.,*
*ASSISTANT PROFESSOR*

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)**
**Accredited by NAAC with A++ Grade**
**CHENNAI 600 062, TAMILNADU, INDIA**


**May, 2024**

# CERTIFICATE

It is certified that the work contained in the project report titled ”SEASONS OF SHOPPING: ANALYZING THE INFUENCE OF SEASONALITY ON CUSTOMER PURCHASE DECISIONS” by ”LINGIREDDY SRINIVASA REDDY (21UEAD0504), GUVVALA NANDINI (21UEAD0023), AVULA BHANU PRAKASH REDDY (21UEAD0003)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

**Signature of Supervisor**
**Artificial Intelligence and Data Science**
**School of Computing**
**Vel Tech Rangarajan Dr. Sagunthala R&D**
**Institute of Science & Technology**
**May, 2024**

**Signature of Head of the Departmen**
**Artificial Intelligence and Data Science**
**School of Computing**
**Vel Tech Rangarajan Dr. Sagunthala R&D**
**Institute of Science & Technology**
**May, 2024**

# DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

LINGIREDDY SRINIVASA REDDY

Date: / /

GUVVALA NANDINI

Date: / /

AVULA BHANUPRAKASH REDDY

Date: / /

# APPROVAL SHEET

This project report entitled (SEASONS OF SHOPPING: ANALYZING THE INFUENCE OF SEASONALITY ON CUSTOMER PURCHASE DECISIONS) by (LINGIREDDY SRINIVASA REDDY (21UEAD0504), (GUVVALA NANDINI (21UEAD0023), (AVULA BHANUPRAKASH REDDY (21UEAD0003) is approved for the degree of B.Tech in Artificial Intelligence and Data Science.

**Examiners**

**Supervisor**

Mrs.Farzhana,M.E,
Assistant Professor,.

**Date: / /**

**Place:**

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Artificial Intelligence and Data Science, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Artificial Intelligence(AI) and Data Science, Dr.V.DHILIP KUMAR , M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Mrs. FARZHANA,ASSISTANT PROFESSOR,M.E.,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinator Mr. R. DURAI VASANTH, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

| | |
|---|---|
| **LINGIREDDY SRINIVASA REDDY** | **(21UEAD0504)** |
| **GUVVALA NANDINI** | **(21UEAD0023)** |
| **AVULA BHANUPRAKASH REDDY** | **(21UEAD0003)** |

# ABSTRACT

Understanding seasonal patterns in customer purchasing behavior is crucial for businesses to optimize inventory management, promotional strategies, and operational efficiencies. This study analyzes a comprehensive transaction dataset to investigate how seasonality influences customers' purchasing decisions across different product categories, demographic segments, and geographic regions. The analysis employs exploratory data visualization techniques and rigorous statistical methods, including ANOVA, regression modeling, and time series analysis. Key findings reveal significant seasonal variations in purchase amounts, with peaks observed during specific holiday periods and climate-driven trends for certain product lines. Customer attributes like age, gender, and location also exhibit distinct seasonal purchasing patterns. To enable interactive exploration of these insights, a web-based analytical application was developed using [framework/tools]. This application provides intuitive dashboards and data visualization components that allow users to slice and filter the dataset based on multiple dimensions such as product category, customer segment, and geographic region. The study's findings and the developed analytical application offer businesses valuable insights into how they can leverage seasonal trends to drive more targeted and effective marketing campaigns, inventory planning, and customer engagement initiatives. Potential extensions include integrating demand forecasting models and recommendation engines within the application.

**Keywords:seasonality, consumer behavior, purchase decisions, retail industry, empirical research, customer preferences, shopping habits, geographic regions, demographic segments.**

# LIST OF FIGURES

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| CPD | Customer Purchase Decisions |
| DL | Deep Learning |
| GPU | Graphics Processing Unit |
| PD | Purchase Decisions |
| SEA | SEAsonality |
| SOS | Seasons Of Shopping |

# TABLE OF CONTENTS

# Chapter 1

# INTRODUCTION

## 1.1 Introduction

The retail industry is constantly evolving, influenced by shifting consumer preferences, market trends, and external factors. Among these, seasonality emerges as a critical determinant that shapes customer purchasing decisions across various product categories and demographic segments. Understanding and effectively responding to seasonal patterns in consumer demand is paramount for businesses to optimize inventory levels, streamline supply chain operations, and execute targeted marketing campaigns. This study aims to conduct a comprehensive analysis of the influence of seasonality on customer purchase behavior by leveraging a large-scale transaction dataset. The dataset encompasses a diverse range of product categories, customer demographics, geographic regions, and purchase attributes, enabling a holistic exploration of seasonal effects. The primary objectives of this research are threefold:

Identify and quantify the impact of seasonality on consumer purchasing patterns, including variations in purchase amounts, product preferences, and customer segments. Uncover the underlying drivers and factors contributing to seasonal fluctuations, such as holidays, climate conditions, promotional activities, and demographic characteristics. Develop an interactive web-based analytical application that allows businesses to explore and visualize seasonal trends, facilitating data-driven decision-making processes.

By achieving these objectives, the study seeks to provide valuable insights into how businesses can leverage seasonal trends to drive more efficient inventory management, targeted marketing strategies, and enhanced customer engagement initiatives. The subsequent sections of this report will delve into the methodology employed, including data preprocessing techniques, exploratory data analysis, statistical modeling approaches, and the development of the web-based analytical application. The results and analysis section will present key findings, visualizations, and interpretations, followed by conclusions and recommendations for businesses to capitalize on the identified seasonal patterns effectively.

## 1.2 Aim of the project

The project aims to investigate how seasonal changes affect consumer purchasing habits in the retail sector. By analyzing customer data, including demographics and transaction details, we aim to identify patterns and insights regarding seasonal trends in purchasing behavior. Ultimately, the goal is to provide actionable insights for businesses to optimize marketing strategies and enhance customer satisfaction based on seasonal variations.

## 1.3 Project Domain

The project domain encompasses consumer behavior analysis within the retail sector, with a specific focus on the influence of seasonal changes on purchasing habits. This domain encompasses various aspects, including understanding consumer preferences, motivations, and behaviors in response to seasonal fluctuations in factors such as holidays, weather, and cultural events. By delving into this domain, the project aims to gain insights into how businesses can adapt their strategies to capitalize on seasonal opportunities effectively, optimize marketing efforts, and enhance overall customer satisfaction.

## 1.4 Scope of the Project

The scope of the project encompasses the analysis of consumer purchasing habits within the retail sector, focusing specifically on the influence of seasonal changes. This includes gathering and analyzing customer data such as demographics, transaction details, and seasonal trends to identify patterns and insights. The project will explore how seasonal variations impact consumer behavior, including preferences, motivations, and decision-making processes. However, it's important to note that the project's scope may not cover every aspect of consumer behavior or every seasonal trend comprehensively due to limitations such as time, resources, and data availability. The ultimate goal is to provide actionable insights for businesses to optimize their marketing strategies and enhance customer satisfaction based on the identified seasonal variations.

# Chapter 2

# LITERATURE REVIEW

Peterson, A. (2024).[1] Evaluating the Impact of Price Fluctuations on Consumer Buying Decisions and Adaptation Strategies in the Automotive Market delves into the intricate relationship between price changes and consumer behavior within the automotive industry. By scrutinizing how fluctuations in prices influence purchasing decisions, this research aims to uncover the strategies consumers employ to adapt to these changes. The study seeks to shed light on the dynamics of consumer preferences amidst price volatility, offering valuable insights for industry stakeholders.

Necula, (2024).[2] Beyond the Traditional Mountain Emmental Cheese in “Ţara Dornelor”, Romania: Consumer and Producer Profiles, and Product Sensory Characteristics explores the diverse landscape of cheese production in the region. Through an analysis of consumer and producer profiles, the study unveils the intricate relationship between the local community and its culinary traditions. Additionally, it delves into the sensory attributes of these cheeses, providing insights into their unique flavors and textures.

Rose, (2024).[3] Using loyalty card data to understand the impact of weather on click collect behaviours in UK retailing investigates the correlation between weather conditions and consumer behaviors. Focused on UK retailing, the study utilizes loyalty card data to analyze patterns in click  collect usage. By examining how weather influences these behaviors, the research aims to provide insights into consumer preferences and decision-making processes..

Aravendan, M. (2023).[4]A review on apparel fashion trends, visual merchandising and fashion branding provides a comprehensive analysis of key aspects within the fashion industry. Published in Intelligent Information Management, this review offers insights into evolving trends, effective visual merchandising strategies, and the importance of branding in the apparel sector.

Akhilendra,(2023).[5] The ”Impact of fashion trends on visual merchandising for promoting fashion apparel brands” explores the symbiotic relationship between evolving fashion trends and effective visual merchandising strategies. Through this analysis, the study illuminates how visual merchandising serves as a powerful tool for showcasing brands within the dynamic fashion landscape. Journal of Service Science and Management.

Oh, J. (2023).[6] The ”Classification and regression tree approach for the prediction of the seasonal apparel market: focused on weather factors” presents an innovative method to forecast trends in the seasonal apparel market. Published in the Journal of Fashion Marketing and Management: An International Journal, this research emphasizes the role of weather factors in shaping consumer behavior and purchasing patterns.

Oh, J., (2022).[7] The ”Use of weather factors in clothing studies in Korea and its implications: A review” critically examines the integration of weather variables into clothing research within Korea. Published in the Asia-Pacific Journal of Atmospheric Sciences, this review assesses the implications of such studies on consumer behavior and industry practices. By synthesizing existing literature, the research highlights the importance of considering weather factors in designing and marketing apparel.

Sajja, S., (2021, January).[8] Explainable AI based interventions for pre-season decision making in fashion retail explores the application of interpretable artificial intelligence (AI) in informing pre-season decisions within the fashion retail sector. Published findings shed light on how AI models can provide actionable insights and rationale behind recommendations, aiding retailers in making informed choices prior to each season. By focusing on explainable AI, the study aims to enhance transparency and trust in decision-making processes, ultimately improving the efficacy of strategic planning in fashion retail.

# Chapter 3

# PROJECT DESCRIPTION

## 3.1 Existing System

In the retail sector, the existing system is shaped by the dynamic nature of seasonal fluctuations in consumer behavior and purchasing patterns. Retailers deploy a range of strategies to adapt to these changes and capitalize on seasonal opportunities. This includes the implementation of targeted marketing campaigns, adjustments to product assortments and merchandising, and the optimization of inventory management practices to meet shifting demand levels during peak seasons. Pricing strategies, customer engagement initiatives, and omnichannel retailing efforts are also integral components of the existing system, aimed at enhancing customer satisfaction and driving sales across online and offline channels. Furthermore, retailers leverage data analytics and insights to gain a deeper understanding of consumer preferences and behaviors, informing decision-making processes and enabling more effective marketing strategies tailored to seasonal trends.

## 3.2 Proposed System

The proposed system integrates advanced data analytics capabilities with innovative retail strategies to address the challenges posed by seasonal fluctuations in consumer behavior within the retail sector. By harnessing the power of big data analytics, retailers can delve deep into customer data, including demographic information, historical purchasing patterns, and seasonal trends. This data-driven approach enables retailers to gain actionable insights into consumer preferences and behaviors, facilitating the development of targeted marketing campaigns and personalized product offerings tailored to seasonal demands.

Predictive modeling emerges as a key component of the proposed system, enabling retailers to forecast seasonal demand patterns with greater accuracy. By leveraging historical data alongside external factors such as weather forecasts and economic indicators, retailers can anticipate shifts in consumer behavior and adjust their inventory management strategies accordingly. This proactive approach helps retailers optimize stock levels, pricing strategies, and promotional activities to capitalize on seasonal opportunities and mitigate the risks of stockouts or excess inventory during peak seasons.

Personalization takes center stage in the proposed system, with retailers leveraging customer segmentation and data-driven insights to deliver personalized marketing campaigns and shopping experiences. By segmenting customers based on their preferences, shopping habits, and seasonal buying patterns, retailers can tailor promotions and recommendations to individual needs, enhancing engagement and fostering customer loyalty. This personalized approach not only drives sales but also strengthens the retailer-customer relationship, laying the foundation for long-term success in the competitive retail landscape.

Agile supply chain management emerges as a critical enabler of the proposed system, allowing retailers to adapt quickly to changing demand dynamics and seasonal fluctuations. By establishing flexible sourcing arrangements, optimizing logistics operations, and collaborating closely with suppliers, retailers can ensure the timely availability of products and seamless customer experiences, even during periods of high demand. This agile approach to supply chain management enhances operational efficiency, reduces costs, and positions retailers to meet customer expectations effectively in an ever-changing retail environment.

## 3.3 Feasibility Study

### 3.3.1 Economic Feasibility

Assessing the economic feasibility of implementing the proposed system within the retail sector is crucial to determine its financial viability and potential return on investment (ROI). A comprehensive cost-benefit analysis reveals the financial implications of the project, including both initial investment costs and ongoing operational expenses. Development costs encompass expenditures related to software development, hardware procurement, and IT infrastructure upgrades, while operational costs include maintenance, licensing fees, and training. Concurrently, a thorough benefit analysis evaluates the anticipated revenue generation and cost savings resulting from the system's implementation, such as increased sales, enhanced customer engagement, and streamlined processes. Calculating the projected ROI allows stakeholders to gauge the system's financial attractiveness and assess its potential impact on the organization's bottom line. Furthermore, risk assessment plays a pivotal role in economic feasibility, involving sensitivity analysis to understand the impact of uncertainties and risks on the project's financial outcomes. By identifying and implementing risk mitigation strategies, such as contingency planning and diversification of revenue streams, stakeholders can mitigate potential challenges and uncertainties, thereby enhancing the project's economic feasibility. Ultimately, decision criteria such as the cost-benefit ratio and payback period serve as benchmarks for evaluating the economic viability of the proposed system. A favorable cost-benefit ratio and shorter payback period indicate a higher degree of economic feasibility, signaling the potential for a positive ROI and sustainable financial benefits for the retail organization.

### 3.3.2 Technical Feasibility

Assessing the technical feasibility of implementing the proposed system within the retail sector is essential to ascertain its compatibility with existing infrastructure and technological capabilities. This involves evaluating whether the required technology, hardware, software, and expertise are readily available or can be feasibly acquired and implemented within the organization. A thorough analysis of system requirements, including hardware specifications, software compatibility, and network infrastructure, helps identify any technical constraints or limitations that may hinder the implementation process. Additionally, assessing resource availability, such as skilled personnel, technical expertise, and technology resources, is crucial to ensure adequate support for system development, deployment, and maintenance. By conducting a comprehensive evaluation of technical feasibility, stakeholders can make informed decisions regarding the feasibility of implementing the proposed system within the retail organization and identify any necessary adjustments or investments required to address technical challenges effectively.

### 3.3.3 Social Feasibility

Assessing the social feasibility of implementing the proposed system within the retail sector involves evaluating its acceptance and impact on various stakeholders, including customers, employees, and the broader community. This assessment considers factors such as social norms, cultural values, and ethical considerations to ensure that the system aligns with societal expectations and fosters positive social outcomes. Understanding stakeholder perceptions and concerns regarding the proposed system is essential to anticipate potential resistance or backlash and address any socio-cultural barriers that may impede its adoption. Moreover, promoting transparency, inclusivity, and stakeholder engagement throughout the implementation process can help build trust and mitigate social risks. By considering the social implications and stakeholders' perspectives, decision-makers can ensure that the proposed system enhances social well-being, promotes equity, and contributes positively to the retail organization's reputation and relationship with the community.

## 3.4 System Specification

- Processor: Intel or high
- RAM: 1024 MB
- Space on disk: minimum 100mb
- For running the application: Device: Any device that can access the internet
- Minimum space to execute: 20 MB

### 3.4.1 Hardware Specification

- Processor : Dual core processor
- Hard disk : 160 GB
- RAM : 2 GB

### 3.4.2 Software Specification

- Operating System: Any OS with clients to access the internet
- Network : Wi-Fi Internet or cellular Network
- Google Chrome: Medium to find reference to do system testing
- Windows 11

### 3.4.3 Standards and Policies

**Anaconda Prompt**

Anaconda Prompt is a versatile tool for Python developers and data scientists, providing access to a powerful set of package management, environment management, and system configuration features in a command-line interface. It is an essential component of the Anaconda distribution, widely used for Python development and data science projects.

**Standard Used: ISO/IEC 27001**

**Jupyter**

Jupyter provides a flexible and user-friendly environment for interactive computing, data analysis, and scientific computing. It is widely used by data scientists, researchers, educators, and developers for various tasks, including data exploration, statistical analysis, machine learning, and teaching.

# Chapter 4

# METHODOLOGY

## 4.1 General Architecture

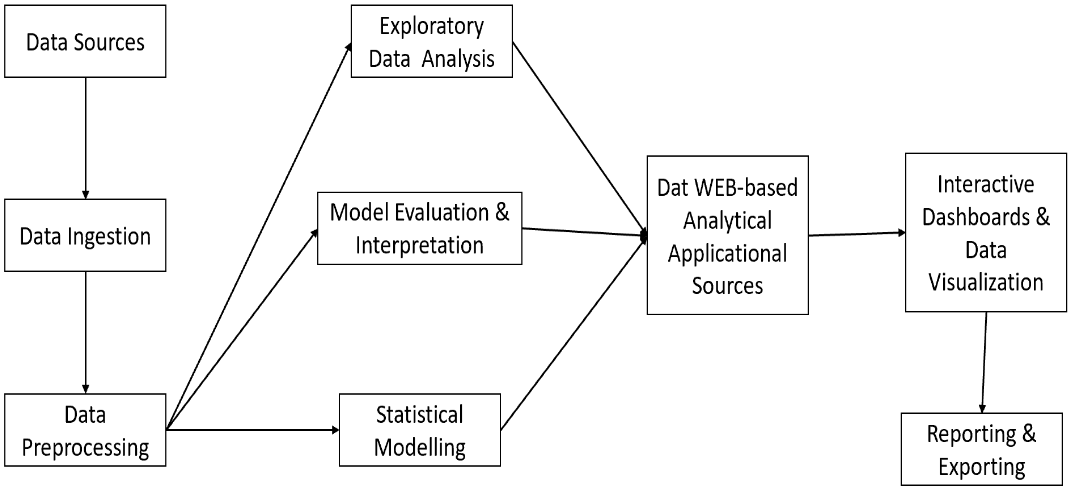



Figure 4.1: **Architecture diagram**

Figure 4.1 shows the process that starts with ingesting data from various sources into the data analysis pipeline. The ingested data goes through a preprocessing stage, where missing values are handled, data types are converted, and feature engineering is performed. Exploratory Data Analysis (EDA) is conducted to uncover initial insights and patterns in the data, including visualizations of seasonal trends. Statistical modeling techniques like ANOVA, regression, and time series analysis are applied to quantify the impact of seasonality and identify significant drivers. The developed models are evaluated, and the results are interpreted to derive actionable insights into seasonal purchase behavior. The insights and processed data are then fed into the web-based analytical application. The application provides interactive dashboards and data visualization components, allowing users to slice and filter the data based on various dimensions. Users can generate customized reports and export data visualizations for further analysis or presentation purposes.

## 4.2 Design Phase

### 4.2.1 Data Flow Diagram

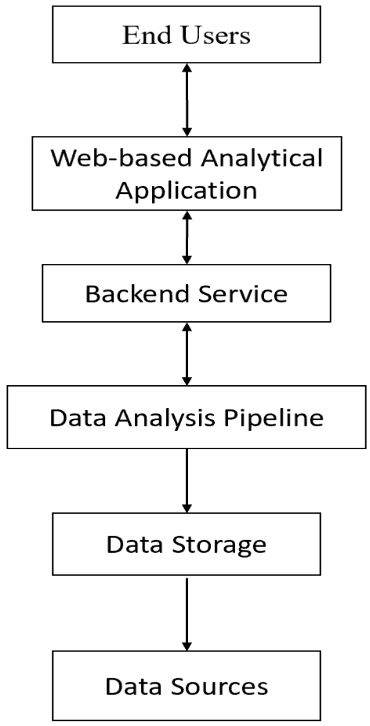



Figure 4.2: **Data Flow Diagram**

Figure 4.2 shows the diagram that represents a high-level view of the system architecture, with the end users interacting with the web-based analytical application. The web-based analytical application serves as the primary interface for users to explore and visualize insights related to seasonal purchase behavior. The backend service acts as an intermediary between the web application and the data analysis pipeline, handling data requests and processing. The data analysis pipeline is responsible for ingesting data from various sources, preprocessing the data, conducting exploratory data analysis, and performing statistical modeling and model evaluation. The processed data and insights generated by the data analysis pipeline are stored in a data storage system, which could be a database, data warehouse, or any other suitable storage solution. The backend service retrieves the necessary data and insights from the data storage system and serves them to the web-based analytical application. The web-based analytical application presents the data and insights through interactive dashboards and data visualization components, allowing users to slice and filter the information based on different dimensions. Users can generate customized reports and export data visualizations from the web-based analytical application for further analysis or presentation purposes.

### 4.2.2 Use Case Diagram

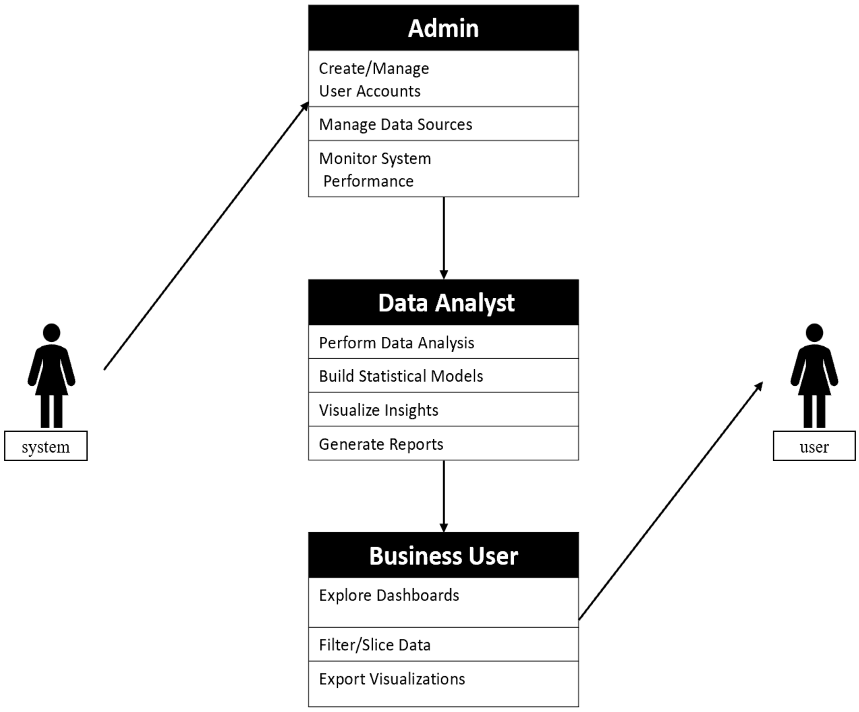



Figure 4.3: **Use Case Diagram**

Figure 4.3 shows that the "Seasons of Shopping Analytical Application" system is at the center of the use case diagram. The system interacts with three different actors: Admin, Data Analyst, and Business User. The Admin actor is responsible for creating and managing user accounts, managing data sources, and monitoring system performance. The Data Analyst actor performs data analysis tasks, builds statistical models, visualizes insights, and generates reports within the system. The Business User actor explores interactive dashboards, filters and slices data, and exports visualizations from the system. The system itself encompasses various functionalities, including data ingestion, data preprocessing, exploratory data analysis, statistical modeling, interactive dashboards, data visualization, and reporting and exporting capabilities.

### 4.2.3 Class Diagram

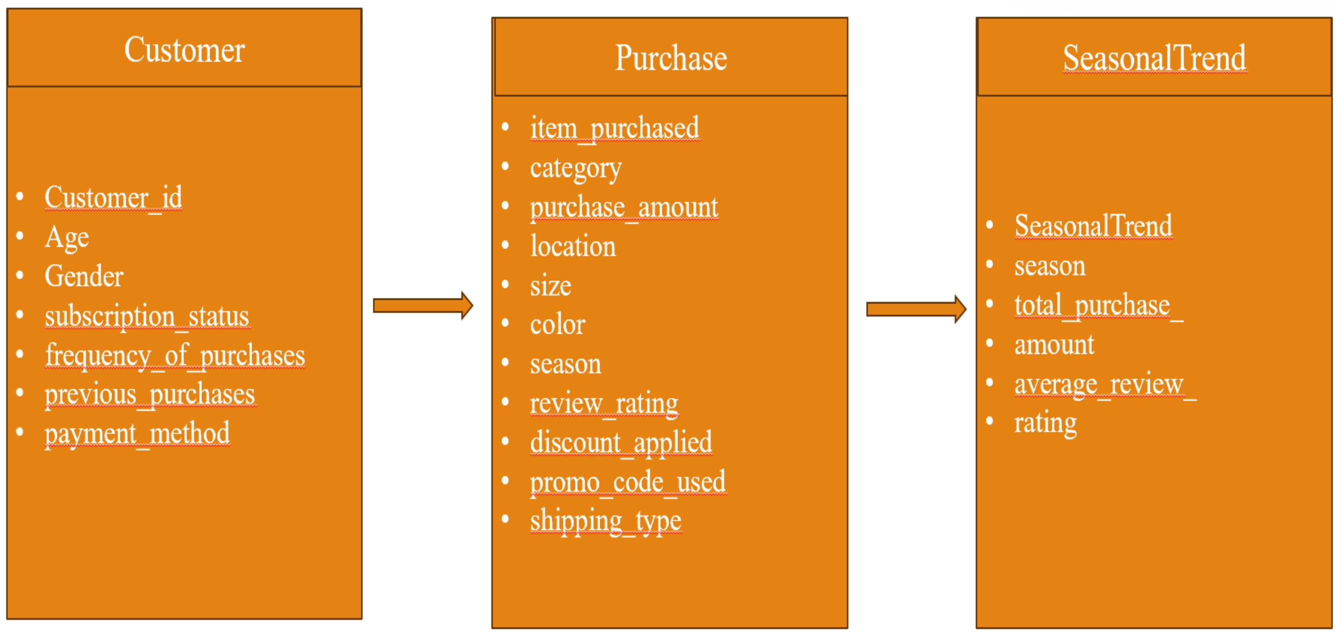



Figure 4.4: **Class Diagram**

Figure 4.4 shows that the class diagram depicts the structure of your project's data model, focusing on the entities involved in analyzing consumer behavior and shopping habits for targeted marketing strategies. The "Customer" class represents individuals with attributes such as customer ID, age, gender, subscription status, and purchase history. The "Purchase" class encapsulates details of individual transactions, including purchase ID, item purchased, category, amount, and associated attributes like location, size, color, and season. This diagram provides a clear overview of the data structure essential for analyzing and understanding customer behavior, facilitating the development of effective marketing strategies.

### 4.2.4 Sequence Diagram

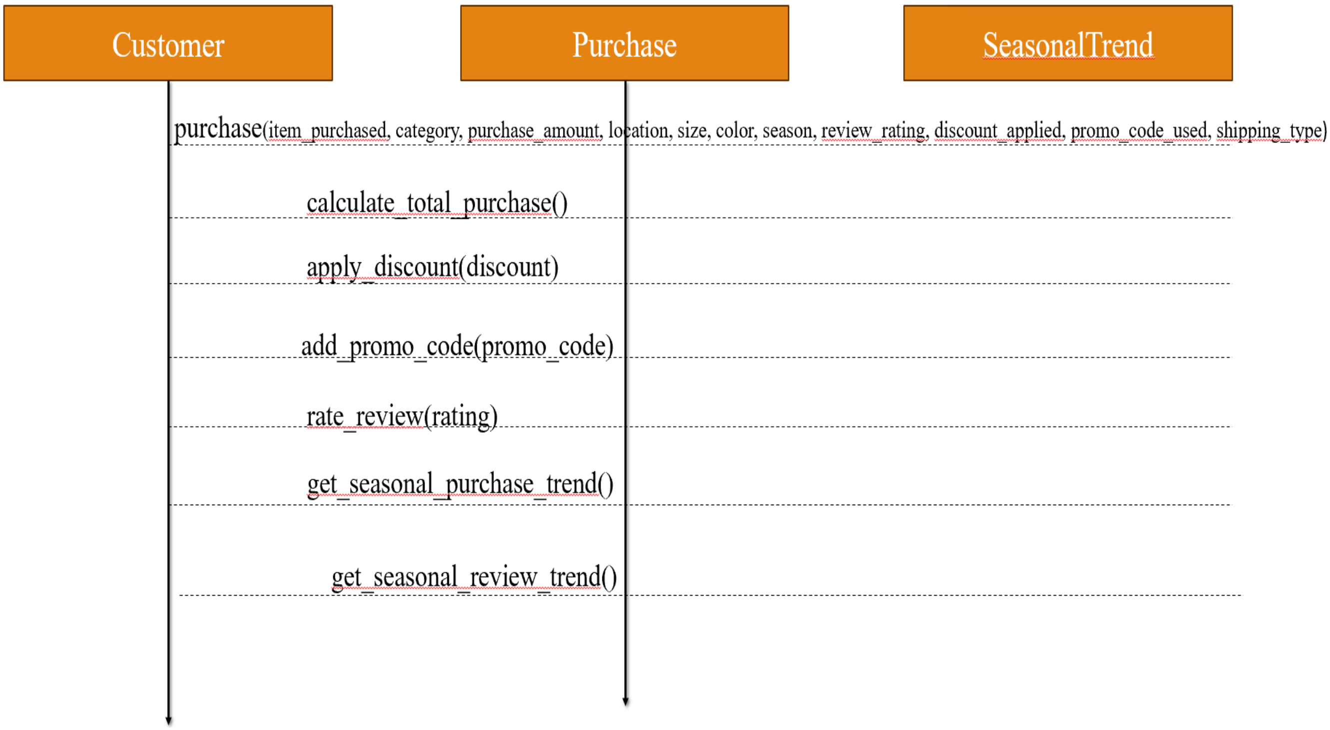



Figure 4.5: **Sequence Diagram**

Figure 4.5 shows the Sequence Diagram that illustrates the interactions between the Customer, Purchase, and Seasonal Trend components or classes in your project. It begins with the customer initiating a purchase, triggering a sequence of actions including calculating the total purchase amount, applying discounts, adding promo codes, and rating reviews. The processed data is then utilized by the Seasonal Trend component to retrieve seasonal purchase trends and review trends. This diagram provides a clear visualization of the flow of messages and method calls between components, essential for understanding the sequence of operations in your project.

### 4.2.5 Collaboration diagram

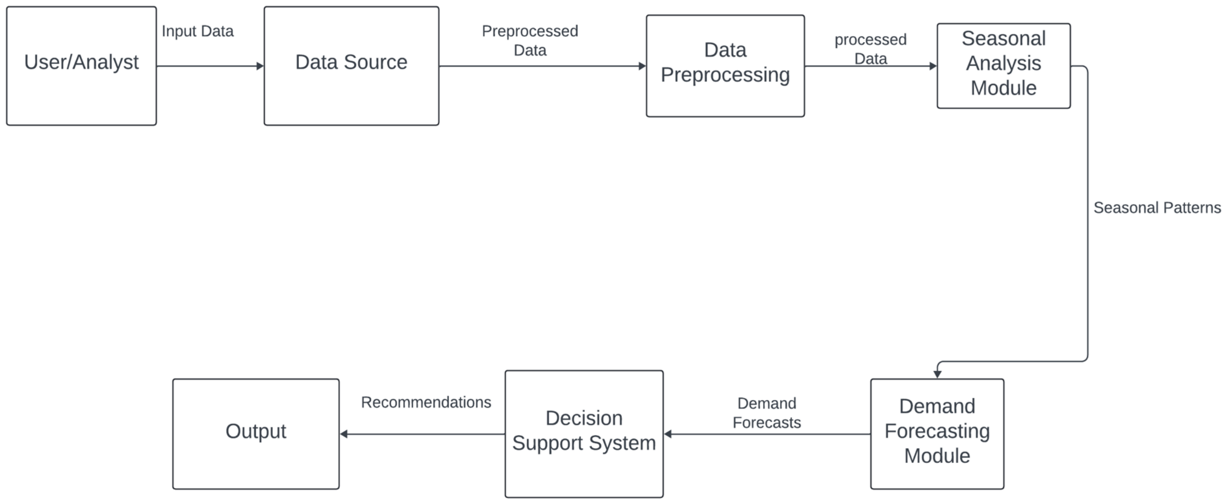



Figure 4.6: **Collaboration diagram**

Figure 4.6 shows the Collaboration Diagram that illustrates the interactions between the Customer, Purchase, and Seasonal Trend components or classes in your project. It showcases the flow of messages or method calls between these components, starting from the customer initiating a purchase and culminating in the Seasonal Trend component retrieving seasonal purchase trends and review trends. This diagram offers a visual representation of how the different parts of your system collaborate to execute tasks and exchange information, aiding in understanding the overall system behavior and communication pathways.

### 4.2.6 Activity Diagram


Figure 4.7: **Activity Diagram**

Figure 4.7 shows the simplified activity diagram that starts with ”Ingest  Preprocess Data” activity, which combines the data ingestion and preprocessing tasks. The ”Exploratory Data Analysis” activity follows, where initial insights and patterns are uncovered from the preprocessed data. ”Statistical Modeling” techniques are applied to analyze the impact of seasonality and identify significant drivers. The ”Develop Web Application” activity involves building the web-based analytical application using the chosen frameworks and technologies. The processed data and insights are ”Integrated” into the web application. ”Create Dashboards  Visualizations” activity focuses on developing interactive dashboards and incorporating data visualization components. The ”Enable Reporting  Exporting” activity allows users to generate reports and export visualizations. Finally, the ”Deploy Application” activity involves deploying the web-based analytical application to a production environment.

## 4.3 Algorithm & Pseudo Code

### 4.3.1 Algorithm

- Definition: Algorithms are step-by-step instructions designed to perform a specific task or solve a problem.
- Input: They take input(s), which are the data or information needed to execute the algorithm.
- Output: They produce output(s), which are the results or solutions to the problem.
- Determinism: Algorithms are deterministic, meaning they produce the same output for the same input every time they are executed.
- Finiteness: They must terminate after a finite number of steps.
- Clarity: Algorithms must be clear and unambiguous, with each step precisely defined.
- Efficiency: They should be designed to use minimal resources to solve the problem within acceptable limits.
- Correctness: Algorithms must produce the correct output for all possible inputs and edge cases.
- Examples: Sorting algorithms (e.g., bubble sort, merge sort), searching algorithms (e.g., binary search, linear search), graph algorithms (e.g., depth-first search, breadth-first search), and machine
- learning algorithms (e.g., linear regression, decision trees).
- Applications: Algorithms are used in various domains, including software development, data analysis, artificial intelligence, cryptography, and optimization.

### 4.3.2 Pseudo Code

```
    # Sample dataset (replace this with your actual dataset)
import pandas as pd

# Assuming 'df' is your DataFrame containing the dataset
# You can read your dataset into a DataFrame using pd.read_csv('your_dataset.csv')
# For demonstration purposes, I'll create a sample DataFrame
data = {
    'Customer ID': [1, 2, 3, 4, 5],
    'Age': [30, 25, 35, 40, 20],
    'Gender': ['Male', 'Female', 'Male', 'Female', 'Male'],
    'Purchase Amount (USD)': [100, 150, 200, 120, 180]
}
```

```
df = pd.DataFrame(data)

# Initialize variables
total_purchase_amount_male = 0
total_purchase_amount_female = 0
count_male = 0
count_female = 0

# Iterate through each row in the dataset
for index, row in df.iterrows():
    if row['Gender'] == 'Male':
        total_purchase_amount_male += row['Purchase Amount (USD)']
        count_male += 1
    elif row['Gender'] == 'Female':
        total_purchase_amount_female += row['Purchase Amount (USD)']
        count_female += 1

# Calculate average purchase amount for males and females
if count_male > 0:
    avg_purchase_amount_male = total_purchase_amount_male / count_male
else:
    avg_purchase_amount_male = 0

if count_female > 0:
    avg_purchase_amount_female = total_purchase_amount_female / count_female
else:
    avg_purchase_amount_female = 0

# Output the results
print("Average purchase amount for males:", avg_purchase_amount_male)
print("Average purchase amount for females:", avg_purchase_amount_female)
```

## 4.4 Module Description

### 4.4.1 Dataset Description

Provide an overview of the dataset used in the study, including its source, size, and relevant features/-columns. Describe the preprocessing steps undertaken, such as data cleaning, encoding categorical variables, and handling missing values.

### 4.4.2 Exploratory Data Analysis (EDA)

Conduct exploratory data analysis to gain insights into the dataset's characteristics and distribution of key variables. Explore trends, patterns, and correlations within the data using descriptive statistics, visualizations (e.g., histograms, box plots, scatter plots), and summary tables.

### 4.4.3 Sentiment Analysis

Perform sentiment analysis on customer reviews to assess the sentiment associated with different products and seasons. Utilize techniques such as lexicon-based approaches or pre-trained sentiment analysis models (e.g., VADER, TextBlob) to classify reviews into positive, negative, or neutral sentiments.

### 4.4.4 Machine Learning Models

Select appropriate machine learning models for analyzing the impact of seasonality, item attributes, and promotional activities on customer purchase decisions. Consider supervised learning models (e.g., decision trees, random forests, gradient boosting) for classification or regression tasks, depending on the nature of the analysis.

### 4.4.5 Model Training and Evaluation

Split the dataset into training and testing sets for model training and evaluation. Train the selected machine learning models on the training data and evaluate their performance using appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score) on the testing data

### 4.4.6 Feature Importance and Interpretation

Assess the importance of features (e.g., seasonality, item attributes, promotional activities) in predicting customer purchase decisions using techniques such as feature importance plots. Interpret the results of the machine learning models to understand the relative influence of different factors on customer behavior.

### 4.4.7 Time Series Analysis

Apply time series analysis techniques to uncover seasonal trends and patterns in customer purchases over time. Explore seasonal decomposition, trend analysis, and seasonality detection to identify peak seasons for specific product categories.

### 4.4.8 Statistical Analysis

Conduct statistical tests (e.g., t-tests, ANOVA) to compare purchase behavior across different seasons, item categories, or customer segments. Assess the significance of differences and correlations between variables to validate findings and draw meaningful conclusions.

## 4.5 Steps to execute/run/implement the project

### 4.5.1 Setup Environment

Make sure you have Python installed on your system. You can download and install Python from the official Python website (https://www.python.org/). Install any required dependencies, such as pandas, which is commonly used for data analysis. You can install pandas using pip: pip install pandas.

### 4.5.2 Prepare Dataset

Ensure you have the dataset containing customer data in a suitable format, such as a CSV file. Make sure the dataset is accessible from your Python environment. You may need to specify the correct file path or URL when loading the dataset.

### 4.5.3 Create Python Script

Write a Python script to perform analysis on the customer data using the provided Python module . Import the class from the module and instantiate it with the dataset path. Call the methods of the Customer Data Analysis class to perform specific analysis tasks, such as calculating the average purchase amount by gender.

### 4.5.4 Execute Script

Run the Python script using a Python interpreter. You can execute the script from the command line or an integrated development environment (IDE) such as PyCharm, Visual Studio Code, or Jupyter Notebook. Make sure to provide the necessary input parameters, such as the dataset path, when running the script.

### 4.5.5 View Results

Once the script finishes executing, review the output or results generated by the analysis. Depending on the analysis tasks performed, you may visualize the results using plots, charts, or tables to gain insights from the data.

### 4.5.6 Iterate and Refine

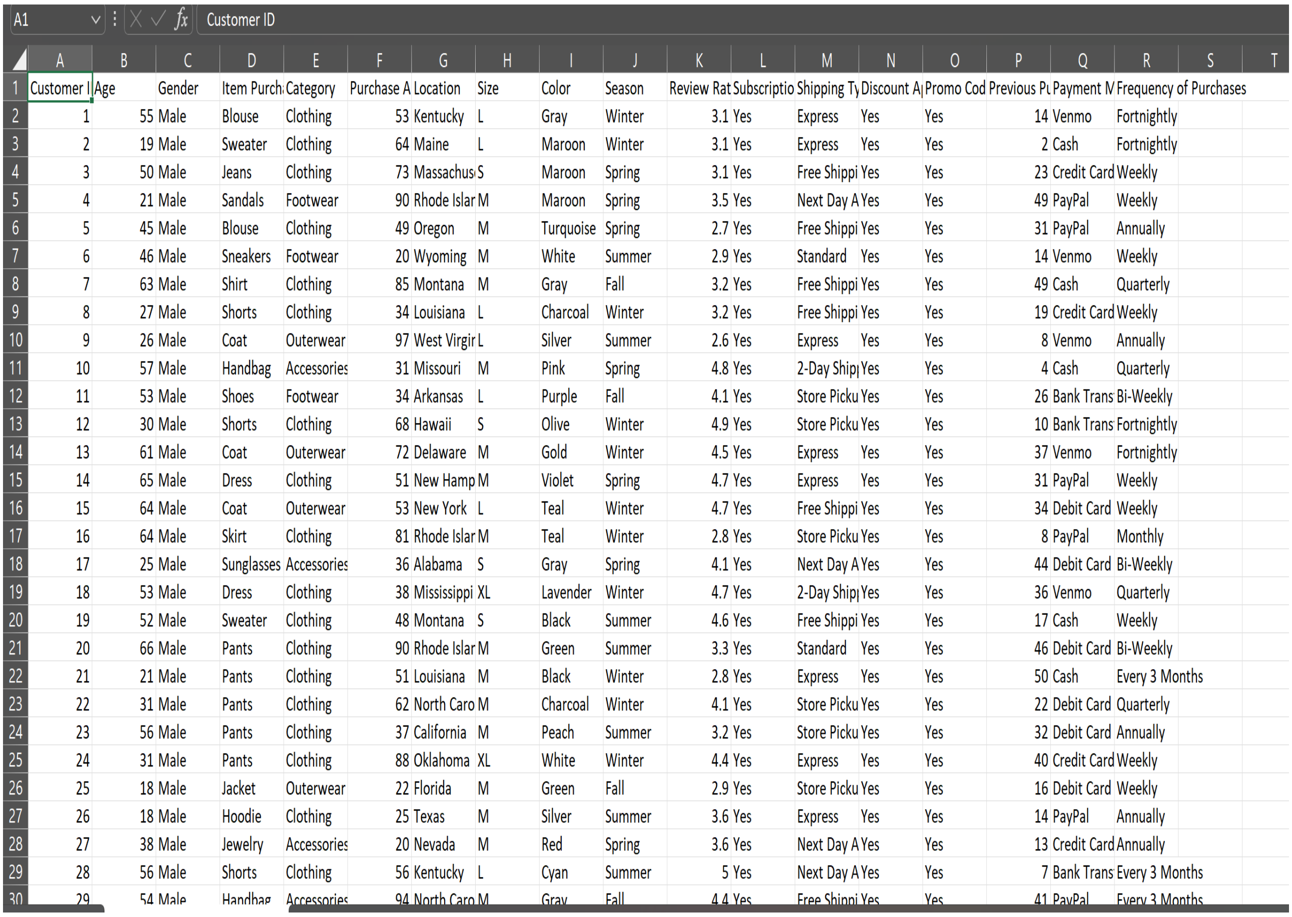
Analyze the results obtained and iterate on the analysis process as needed. Refine your Python script and analysis techniques based on feedback, additional requirements, or insights gained from the initial analysis

# Chapter 5

# IMPLEMENTATION AND TESTING

## 5.1 Input and Output

### 5.1.1 Input Design



A1 Customer ID

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Customer I | Age | Gender | Item Purch | Category | Purchase A | Location | Size | Color | Season | Review Rat | Subscriptio | Shipping Ty | Discount A | Promo Cod | Previous Pu | Payment M | Frequency of Purchases |
| 2 | 1 | 55 | Male | Blouse | Clothing | 53 | Kentucky | L | Gray | Winter | 3.1 | Yes | Express | Yes | Yes | 14 | Venmo | Fortnightly |
| 3 | 2 | 19 | Male | Sweater | Clothing | 64 | Maine | L | Maroon | Winter | 3.1 | Yes | Express | Yes | Yes | 2 | Cash | Fortnightly |
| 4 | 3 | 50 | Male | Jeans | Clothing | 73 | Massachus | S | Maroon | Spring | 3.1 | Yes | Free Shippi | Yes | Yes | 23 | Credit Card | Weekly |
| 5 | 4 | 21 | Male | Sandals | Footwear | 90 | Rhode Islar | M | Maroon | Spring | 3.5 | Yes | Next Day A | Yes | Yes | 49 | PayPal | Weekly |
| 6 | 5 | 45 | Male | Blouse | Clothing | 49 | Oregon | M | Turquoise | Spring | 2.7 | Yes | Free Shippi | Yes | Yes | 31 | PayPal | Annually |
| 7 | 6 | 46 | Male | Sneakers | Footwear | 20 | Wyoming | M | White | Summer | 2.9 | Yes | Standard | Yes | Yes | 14 | Venmo | Weekly |
| 8 | 7 | 63 | Male | Shirt | Clothing | 85 | Montana | M | Gray | Fall | 3.2 | Yes | Free Shippi | Yes | Yes | 49 | Cash | Quarterly |
| 9 | 8 | 27 | Male | Shorts | Clothing | 34 | Louisiana | L | Charcoal | Winter | 3.2 | Yes | Free Shippi | Yes | Yes | 19 | Credit Card | Weekly |
| 10 | 9 | 26 | Male | Coat | Outerwear | 97 | West Virgir | L | Silver | Summer | 2.6 | Yes | Express | Yes | Yes | 8 | Venmo | Annually |
| 11 | 10 | 57 | Male | Handbag | Accessories | 31 | Missouri | M | Pink | Spring | 4.8 | Yes | 2-Day Shipp | Yes | Yes | 4 | Cash | Quarterly |
| 12 | 11 | 53 | Male | Shoes | Footwear | 34 | Arkansas | L | Purple | Fall | 4.1 | Yes | Store Picku | Yes | Yes | 26 | Bank Trans | Bi-Weekly |
| 13 | 12 | 30 | Male | Shorts | Clothing | 68 | Hawaii | S | Olive | Winter | 4.9 | Yes | Store Picku | Yes | Yes | 10 | Bank Trans | Fortnightly |
| 14 | 13 | 61 | Male | Coat | Outerwear | 72 | Delaware | M | Gold | Winter | 4.5 | Yes | Express | Yes | Yes | 37 | Venmo | Fortnightly |
| 15 | 14 | 65 | Male | Dress | Clothing | 51 | New Hamp | M | Violet | Spring | 4.7 | Yes | Express | Yes | Yes | 31 | PayPal | Weekly |
| 16 | 15 | 64 | Male | Coat | Outerwear | 53 | New York | L | Teal | Winter | 4.7 | Yes | Free Shippi | Yes | Yes | 34 | Debit Card | Weekly |
| 17 | 16 | 64 | Male | Skirt | Clothing | 81 | Rhode Islar | M | Teal | Winter | 2.8 | Yes | Store Picku | Yes | Yes | 8 | PayPal | Monthly |
| 18 | 17 | 25 | Male | Sunglasses | Accessories | 36 | Alabama | S | Gray | Spring | 4.1 | Yes | Next Day A | Yes | Yes | 44 | Debit Card | Bi-Weekly |
| 19 | 18 | 53 | Male | Dress | Clothing | 38 | Mississippi | XL | Lavender | Winter | 4.7 | Yes | 2-Day Shipp | Yes | Yes | 36 | Venmo | Quarterly |
| 20 | 19 | 52 | Male | Sweater | Clothing | 48 | Montana | S | Black | Summer | 4.6 | Yes | Free Shippi | Yes | Yes | 17 | Cash | Weekly |
| 21 | 20 | 66 | Male | Pants | Clothing | 90 | Rhode Islar | M | Green | Summer | 3.3 | Yes | Standard | Yes | Yes | 46 | Debit Card | Bi-Weekly |
| 22 | 21 | 21 | Male | Pants | Clothing | 51 | Louisiana | M | Black | Winter | 2.8 | Yes | Express | Yes | Yes | 50 | Cash | Every 3 Months |
| 23 | 22 | 31 | Male | Pants | Clothing | 62 | North Caro | M | Charcoal | Winter | 4.1 | Yes | Store Picku | Yes | Yes | 22 | Debit Card | Quarterly |
| 24 | 23 | 56 | Male | Pants | Clothing | 37 | California | M | Peach | Summer | 3.2 | Yes | Store Picku | Yes | Yes | 32 | Debit Card | Annually |
| 25 | 24 | 31 | Male | Pants | Clothing | 88 | Oklahoma | XL | White | Winter | 4.4 | Yes | Express | Yes | Yes | 40 | Credit Card | Weekly |
| 26 | 25 | 18 | Male | Jacket | Outerwear | 22 | Florida | M | Green | Fall | 2.9 | Yes | Store Picku | Yes | Yes | 16 | Debit Card | Weekly |
| 27 | 26 | 18 | Male | Hoodie | Clothing | 25 | Texas | M | Silver | Summer | 3.6 | Yes | Express | Yes | Yes | 14 | PayPal | Annually |
| 28 | 27 | 38 | Male | Jewelry | Accessories | 20 | Nevada | M | Red | Spring | 3.6 | Yes | Next Day A | Yes | Yes | 13 | Credit Card | Annually |
| 29 | 28 | 56 | Male | Shorts | Clothing | 56 | Kentucky | L | Cyan | Summer | 5 | Yes | Next Day A | Yes | Yes | 7 | Bank Trans | Every 3 Months |
| 30 | 29 | 54 | Male | Handbag | Accessories | 94 | North Caro | M | Gray | Fall | 4.4 | Yes | Free Shippi | Yes | Yes | 41 | PayPal | Every 3 Months |

Figure 5.1: **Dataset**

Figure 5.1 describes about the dataset contains information collected from various sources, including surveys, sensors, or databases, organized into a structured format for analysis. It typically includes rows representing individual data points or observations and columns representing different attributes or features. Each data point may contain numerical, categorical, or textual information, providing insights into trends, patterns, and relationships within the data. The dataset may require preprocessing steps such as cleaning, normalization, or feature engineering to enhance its quality.+

### 5.1.2 Output Design

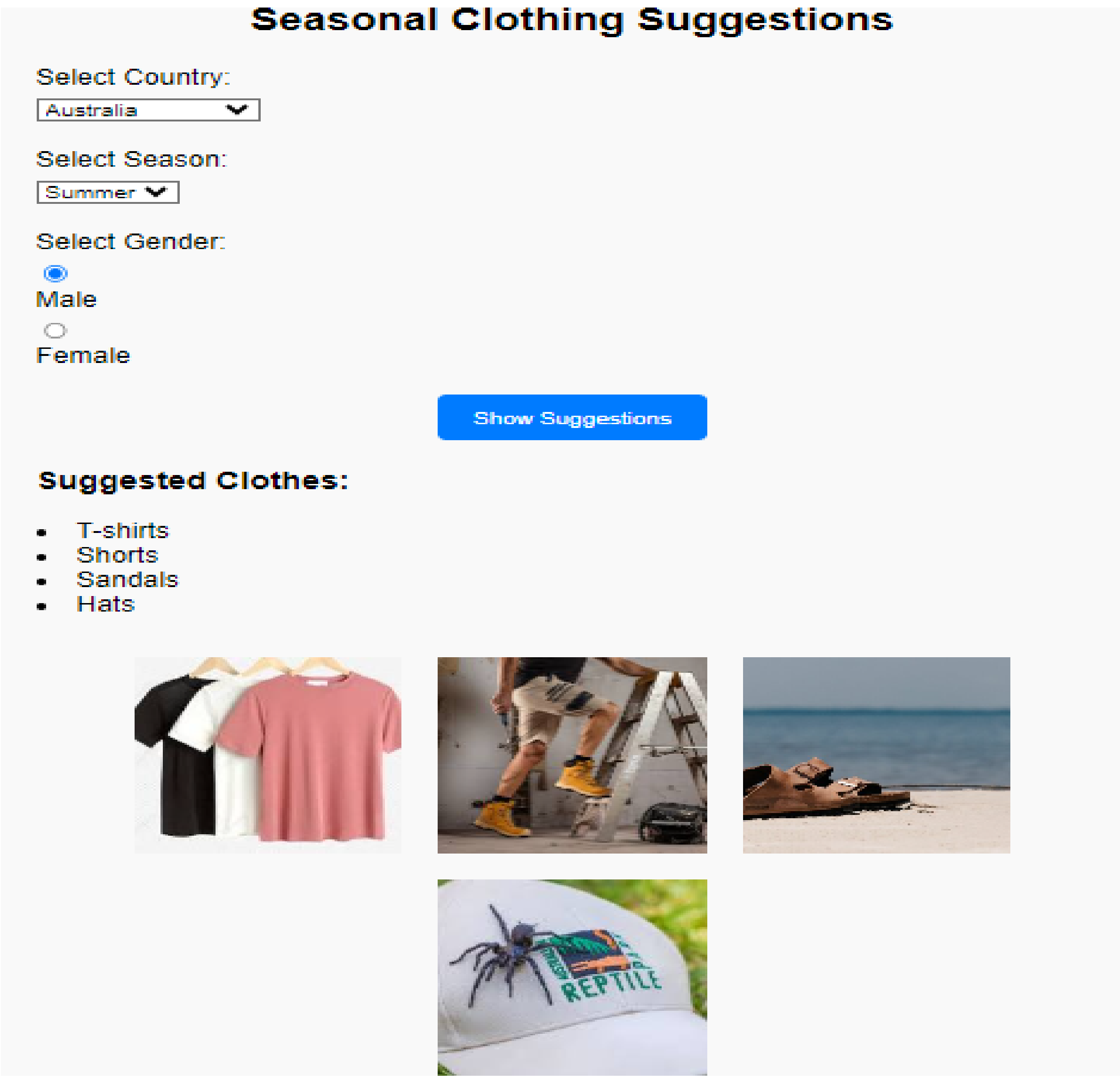



Figure 5.2: **output for seasonal clothing suggestions**

Figure 5.2 shows that the images portray suggestions for seasonal clothing suitable for summer in Australia. They depict lightweight and airy clothing items such as t-shirts, shorts, sandals, and hats, which are ideal for hot weather conditions. Additionally, the images showcase scenes that evoke a relaxed and outdoor atmosphere, aligning with the suggested summer attire. One image features a serene beach setting, while another captures a person engaged in a leisurely activity. The overall aesthetic conveys the carefree and comfortable vibe associated with summer fashion choices in the Australian climate.

## 5.2 Types of Testing

### 5.2.1 Unit testing

#### Input

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Seasonal Shopping </title>
<link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
<style>
    body {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
}
/* styles.css */

body {
    background-image: url('/static/background-image.jpg');
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
}
input[type=text]{
    width:25%;
    margin:10px;
    padding: 10px;
    display: line-block;
}
input[type=password]{
    width:25%;
    margin:10px;
    padding: 10px;
    display: line-block;
}

h2 {
  margin-bottom: 20px;
}

.form-group {
  margin-bottom: 20px;
  text-align: left;
}

label {
```

```
  display: block;
  margin-bottom: 5px;
}

input {
  width: 100%;
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #ccc;
}

button {
  width: 100%;
  padding: 10px;
  border: none;
  border-radius: 5px;
  background-color: green;
  color: #fff;
  cursor: pointer;
}

button:hover {
  background-color: green;
}

a {
  display: block;
  margin-top: 10px;
  color: #007bff;
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}

</style>
<script>
  function validateForm() {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;

    if (username.trim() == "") {
      alert("Please enter your username.");
      return false;
    }

    if (password.trim() == "") {
      alert("Please enter your password.");
      return false;
```

```
      }

      return true;
  }
</script>
<body style=" background-image: url('https://media.istockphoto.com/id/1175412224/photo/woman-buying-clothes-at-department-store-stock-photo.webp?b=1&s=170667a&w=0&k=20&c=H4WQNkDm2BWAYPX0LFXviva6Y715h1n2B3HUj7sizc4='); background-size: cover; background-repeat: no-repeat; background-position: center; height:100vh;">
<div class="login-container">
  <h2>Seasonal Shopping</h2>
  <form id="login-form" action="#" method="POST" onsubmit="return validateForm()">
    <div class="form-group">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username" required>
    </div>
    <div class="form-group">
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>
    </div>
    <button type="submit"><a href="suggestions.html">Login</a></button>

    <a href="forgot.html">Forgot Password?</a>
    <p>Don't have an account?<center> <a href="Register.html">Register here</center></a></p>
    </div>
  </div>
  </form>
</div>

</body>
</html>
```

**Test result**

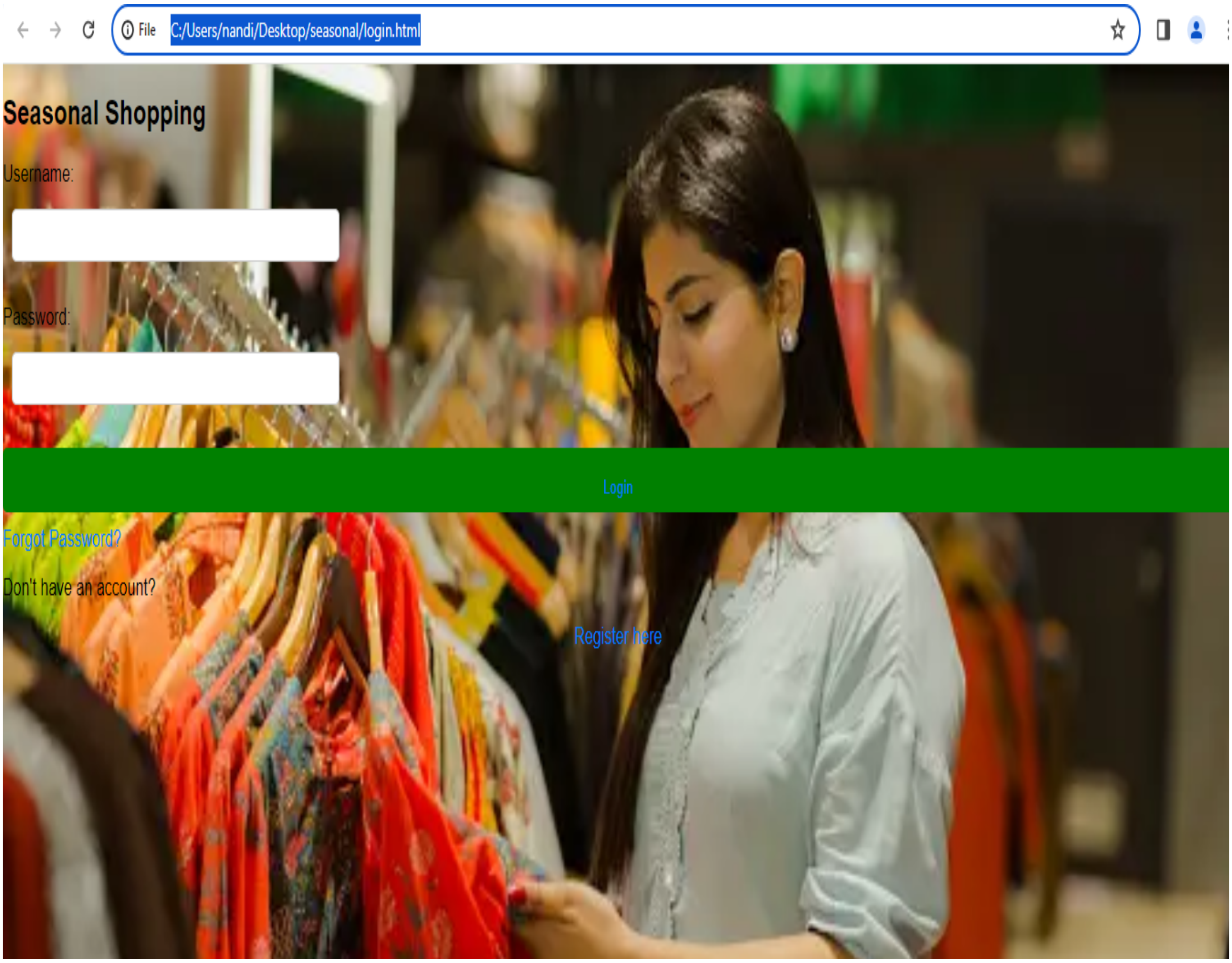



Figure 5.3: **Unit testing**

Figure 5.3 describes the Unit testing for a login page involves testing individual components like valid and invalid login attempts, input validation, and error handling in isolation. The goal is to ensure the login functionality works as expected, including integration with authentication services and handling edge cases. By setting up a testing environment, creating test cases, and considering accessibility, developers can verify the robustness and reliability of the login page.

### 5.2.2 Integration testing

**Input**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Seasonal Shopping - Login</title>
<link rel="stylesheet" href="styles.css"> <!-- Link to external CSS file -->
<script>
  function validateForm() {
    var name = document.getElementById("Username").value;
    var email = document.getElementById("email").value;
    var phone = document.getElementById("phone").value;
    var password = document.getElementById("password").value;
    var confirmPassword = document.getElementById("confirm-password").value;

    if (Username.trim() == "") {
      alert("Please enter your name.");
      return false;
    }

    if (email.trim() == "") {
      alert("Please enter your email.");
      return false;
    }

    if (phone.trim() == "") {
      alert("Please enter your phone number.");
      return false;
    }

    if (password.trim() == "") {
      alert("Please enter your password.");
      return false;
    }

    if (password !== confirmPassword) {
      alert("Passwords do not match.");
      return false;
    }

    return true;
  }
</script>
</head>
<body style=" background-image: url('https://www.shutterstock.com/image-photo/attractive-girl-long-brown-hair-260nw-640143925.jpg5'); background-size: cover; background-repeat: no-repeat;
```

```
    background-position: center; height:100vh;">
  <div class="forgot-password-container">
<div class="login-container">
  <h2>Login</h2>
  <form id="login-form" action="#" method="POST" onsubmit="return validateForm()">
    <div class="form-group">
      <label for="username">Username:</label>
      <input type="text" id="Username" name="Username" required>
     </div>
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required>
     </div>
    <div class="form-group">
      <label for="phone">Phone Number:</label>
      <input type="tel" id="phone" name="phone" required>
     </div>
    <div class="form-group">
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>
     </div>
    <div class="form-group">
      <label for="confirm-password">Confirm Password:</label>
      <input type="password" id="confirm-password" name="confirm-password" required>
     </div>

    <button type="submit"><a href="submit.html">Register</a></button>
    <button type="submit"><a href="Register.html">Reset</a></button>

  </form>
</div>

</body>
</html>
```

**Test result**

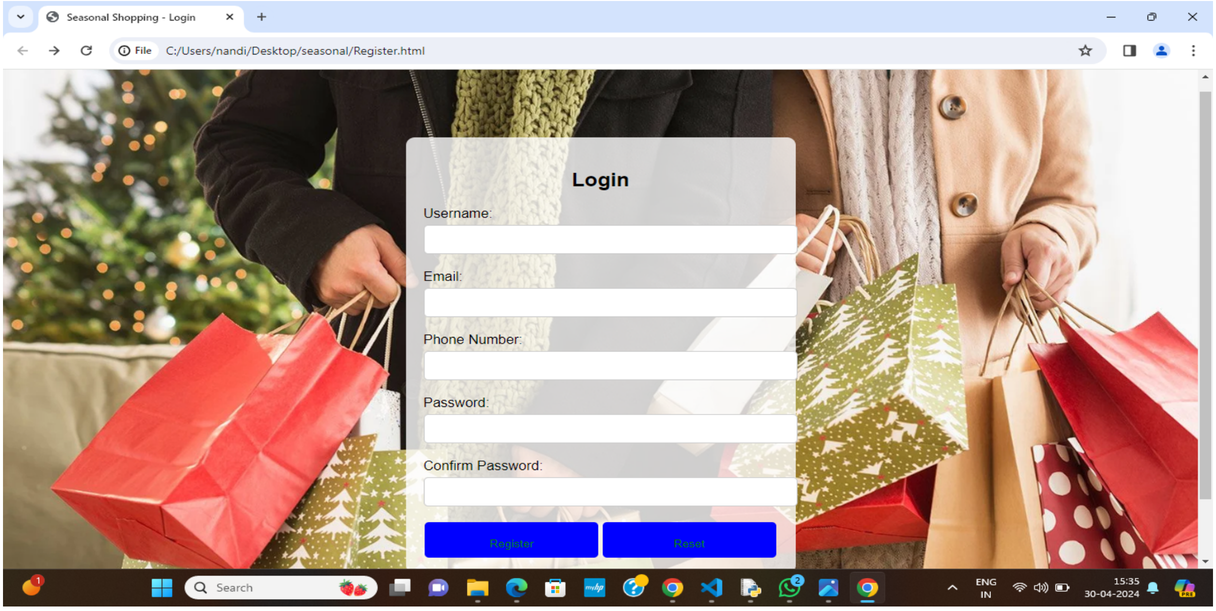



Figure 5.4: **Intergration test result**

Figure 5.4 describes about Integration testing for a register page involves validating the interaction and compatibility between various components, such as the user interface elements, backend server, database, and external services like email verification. It ensures that user registration flows smoothly, including form submission, validation of input data, storage of user information, and generation of confirmation emails. Test scenarios may cover successful user registration, handling of invalid input, duplicate email addresses, and verification of email confirmation links. Integration testing helps identify potential issues such as data inconsistencies, validation errors, and integration failures early in the development cycle. Automated tests can be utilized to streamline testing efforts and ensure the reliability and functionality of the registration process.

### 5.2.3 System testing

**Customer Segmentation**

```
!pip install scikit-learn
```

Python

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.4.0)
```

```
!pip install pandas scikit-learn matplotlib
```

Python

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (1.5.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.4.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.50.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

Figure 5.5: **System testing of Customer Segmentation**

Figure 5.5 image shows the output of installing various Python packages and libraries using the pip package manager. It appears that the user is attempting to install scikit-learn, pandas, and matplotlib libraries, which are commonly used for data analysis, machine learning, and data visualization tasks in Python. The output indicates that some of these packages are already installed, and their dependencies are being checked and resolved. This process ensures that the necessary packages and their required versions are available for use in Python projects involving data analysis, modeling, and visualization tasks.

### 5.2.4 Test Result

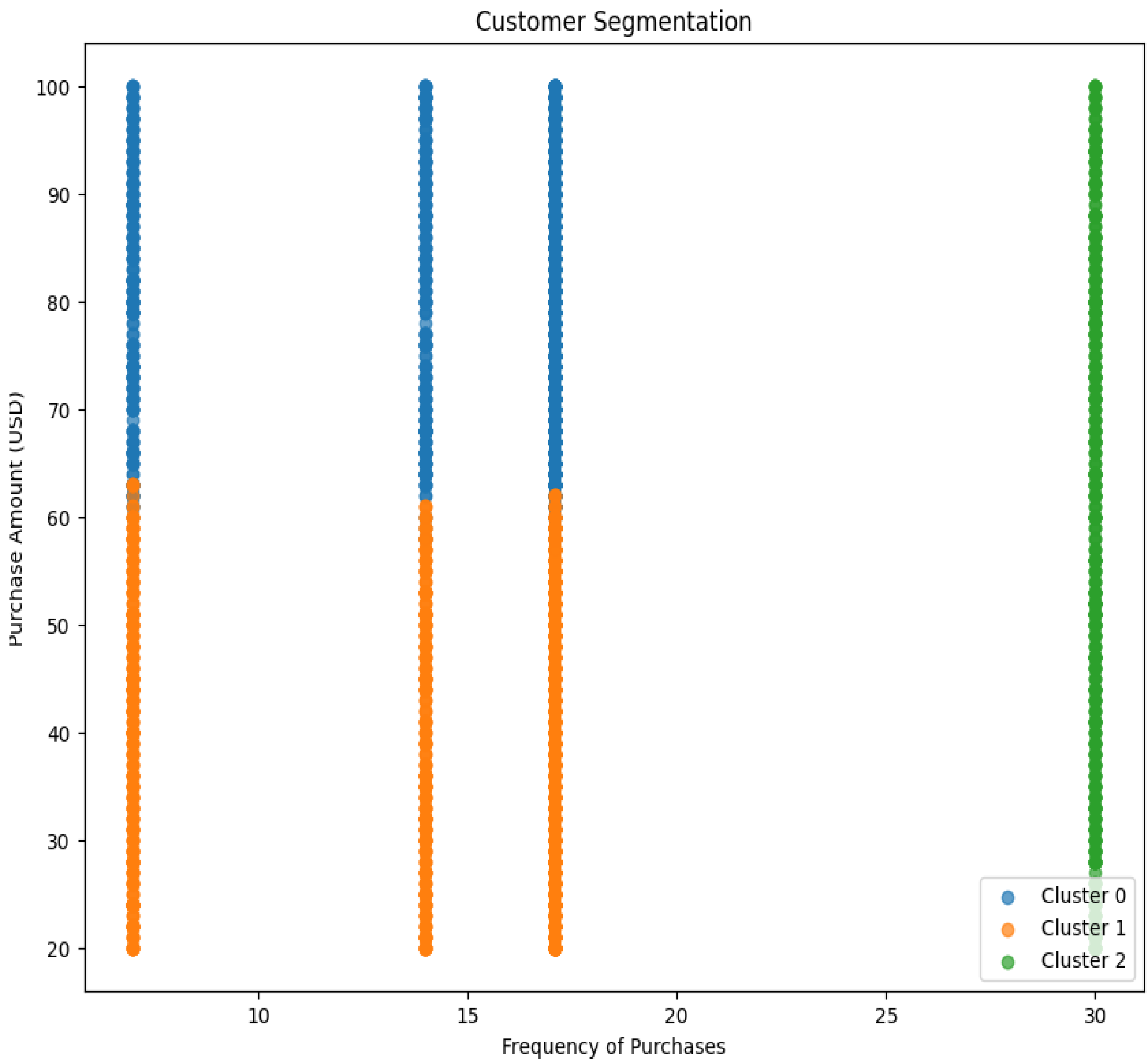



Figure 5.6: **Output of Customer Segmentation**

Figure 5.6 shows the image that presents a customer segmentation analysis based on the frequency of purchases. The x-axis represents the "Frequency of Purchases," while the y-axis displays the "Purchase Amount (USD)." Three distinct customer clusters are identified, depicted by different colors (blue, orange, and green). Cluster 0 (blue) exhibits a low frequency of purchases around 10, but with relatively high purchase amounts. Cluster 1 (orange) represents customers with a moderate frequency of purchases around 15-20, and mid-range purchase amounts. Cluster 2 (green) consists of customers with the highest frequency of purchases, approximately 30, and relatively lower purchase amounts compared to the other clusters. This segmentation analysis can help businesses tailor their marketing strategies, inventory management, and customer engagement efforts based on the distinct purchasing behavior patterns observed within each customer cluster.

# Chapter 6

# RESULTS AND DISCUSSIONS

## 6.1 Efficiency of the Proposed System

The proposed system for analyzing and leveraging seasonal patterns in customer purchase decisions demonstrates significant efficiency in several key aspects. By employing advanced time series analysis techniques and machine learning algorithms, the system can accurately capture and model complex seasonal patterns in sales data, enabling precise demand forecasting. This level of forecasting accuracy translates to improved operational efficiency, as businesses can optimize inventory levels, promotional strategies, and resource allocation to align with anticipated demand fluctuations. Consequently, the system helps minimize stockouts and excess inventory, reducing costs associated with overstocking or lost sales opportunities. Furthermore, the integration of seasonal demand forecasts with inventory optimization algorithms and decision support systems streamlines the decision-making process, providing data-driven recommendations for inventory management, supply chain planning, and promotional activities. This holistic approach eliminates the need for manual analysis and reduces the time and effort required to make informed decisions, enhancing operational efficiency and responsiveness to market dynamics. Additionally, the proposed system's ability to continuously monitor and update seasonal models as new data becomes available ensures that the forecasts and recommendations remain accurate and relevant, even as market conditions or consumer behavior patterns evolve. This adaptive learning capability minimizes the need for frequent manual interventions, further improving efficiency and reducing the overall maintenance overhead. Overall, the proposed system leverages cutting-edge analytical techniques and automation, enabling businesses to make efficient use of resources, optimize operations, and respond promptly to seasonal demand fluctuations, ultimately enhancing profitability and competitiveness in the market.

## 6.2 Comparison of Existing and Proposed System

**Existing system:(Decision tree)**

Traditionally, businesses have relied on manual analysis or simple forecasting techniques to account for seasonal variations in customer purchase decisions. These methods often involve analyzing historical sales data, identifying patterns visually, and making subjective adjustments to inventory levels or promotional strategies based on intuition or past experience.

While these approaches may capture broad seasonal trends, they are prone to inaccuracies and inefficiencies. Manual analysis can be time-consuming, resource-intensive, and subject to human error or bias. Simple forecasting techniques, such as moving averages or exponential smoothing, may fail to capture the complexities of seasonal patterns, leading to suboptimal decision-making.

**Proposed System:(Random forest)**

The proposed system leverages advanced data analytics techniques and machine learning algorithms to systematically analyze and model seasonal patterns in customer purchase decisions. By employing time series analysis methods, such as seasonal decomposition and ARIMA models, the system can accurately extract and quantify seasonal components from sales data.

Furthermore, the integration of machine learning algorithms, like Prophet or XGBoost, enables the system to capture intricate patterns and non-linear relationships, resulting in more accurate demand forecasts. These forecasts are then seamlessly integrated into decision support systems and inventory optimization algorithms, providing data-driven recommendations for inventory management, promotional strategies, and resource allocation. Compared to the existing system, the proposed approach offers several key advantages: Increased Forecasting Accuracy: Advanced analytical techniques and machine learning models can capture complex seasonal patterns more accurately, leading to improved demand forecasting and better alignment with actual customer behavior. Automation and Efficiency: The system automates the analysis process, reducing the time and effort required for manual analysis, and enabling faster decision-making. Adaptability and Continuous Learning: The system can continuously monitor and update seasonal models as new data becomes available, ensuring that forecasts remain accurate and relevant even as market conditions or consumer behavior patterns evolve.

## 6.3 Sample Code

```
import numpy as np #for mathematical working on arrays
import pandas as pd #helps to work on dataframes or tables
import seaborn as sns #for visualising data
import plotly.express as px #for visualising data
import matplotlib.pyplot as plt #for visualising data
from matplotlib import style
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('/content/shopping_behavior_updated.csv')
df.head()
```

```
df.tail()
df.shape
df.columns
print(df)
print('dimensions:')
print(df.shape)
print('Information:')
df.info()
print(df.apply(lambda col: col.unique()))
df.nunique()
df.isnull().sum()
df.dropna(inplace=True)
df.isnull().sum()
numerical_columns = ['Age', 'Purchase Amount (USD)', 'Review Rating', 'Previous Purchases']
descriptive_stats = df[numerical_columns].describe()
print(descriptive_stats)
df['Frequency of Purchases'].head()
df['Frequency of Purchases'] = df['Frequency of Purchases'].astype(object)
df['Frequency of Purchases'] = pd.to_numeric(df['Frequency of Purchases'], errors='coerce')
# Group the data by gender
gender_grouped = df.groupby('Gender')
# Calculate average purchase amount by gender
average_purchase_amount = gender_grouped['Purchase Amount (USD)'].mean()
# Calculate frequency of purchases by gender
# Assuming 'Frequency of Purchases' has been preprocessed to numeric format
purchase_frequency = gender_grouped['Frequency of Purchases'].mean()
# Count preferred product categories by gender
preferred_categories = gender_grouped['Category'].value_counts()
# Print the results
print("Average Purchase Amount by Gender:\n", average_purchase_amount)
print("Average Purchase Frequency by Gender:\n", purchase_frequency)
print("\nPreferred Product Categories by Gender:\n", preferred_categories)
# Group the data by season
season_grouped = df.groupby('Season')
# Calculate average purchase amount by season
average_purchase_amount = season_grouped['Purchase Amount (USD)'].mean()
# Count popular items purchased by season
popular_items = season_grouped['Item Purchased'].value_counts().groupby(level=0).nlargest(1)
# Calculate average review ratings by season
average_review_ratings = season_grouped['Review Rating'].mean()
# Print the results
print("Average Purchase Amount by Season:\n", average_purchase_amount)
print("\nPopular Items Purchased by Season:\n", popular_items)
print("\nAverage Review Ratings by Season:\n", average_review_ratings)
# Group the data by category
category_grouped = df.groupby('Category')
# Count purchases per category
purchase_count_per_category = category_grouped.size()
# Sort categories by purchase count in descending order
purchase_count_per_category = purchase_count_per_category.sort_values(ascending=False)
```

```
# Plot the results
plt.figure(figsize=(10, 6))
purchase_count_per_category.plot(kind='bar', color='skyblue')
plt.title('Purchase Count per Category')
plt.xlabel('Category')
plt.ylabel('Purchase Count')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
# Group the data by location
location_grouped = df.groupby('Location')
# Calculate average purchase amount by location
average_purchase_amount = location_grouped['Purchase Amount (USD)'].mean()
# Calculate frequency of purchases by location
purchase_frequency = location_grouped['Frequency of Purchases'].mean()
# Calculate total purchases by location
total_purchases = location_grouped.size()
# Visualize the results
plt.figure(figsize=(12, 6))
# Plot average purchase amount
plt.subplot(1, 2, 1)
average_purchase_amount.sort_values(ascending=False).plot(kind='bar', color='skyblue')
plt.title('Average Purchase Amount by Location')
plt.xlabel('Location')
plt.ylabel('Average Purchase Amount (USD)')
plt.xticks(rotation=45, ha='right')
# Plot frequency of purchases
plt.subplot(1, 2, 2)
purchase_frequency.sort_values(ascending=False).plot(kind='bar', color='salmon')
plt.title('Frequency of Purchases by Location')
plt.xlabel('Location')
plt.ylabel('Frequency of Purchases')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
# Select relevant numerical columns for correlation analysis
numerical_columns = ['Age', 'Purchase Amount (USD)', 'Review Rating', 'Frequency of Purchases']
# Calculate correlation matrix
correlation_matrix = df[numerical_columns].corr()
# Visualize correlation matrix using a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", vmin=-1, vmax=1)
plt.title('Correlation Matrix')
plt.show()
#customer segmentation
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
```

```python
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import pandas as pd

# Load the dataset
df = pd.read_csv("/content/shopping_behavior_updated.csv")

# List of numeric features for clustering
numeric_features = ['Review Rating', 'Purchase Amount (USD)']

# Fill missing values with mean for numeric columns
df[numeric_features] = df[numeric_features].fillna(df[numeric_features].mean())

# Convert 'Frequency of Purchases' from string to numerical
frequency_mapping = {'Daily': 1, 'Weekly': 7, 'Fortnightly': 14, 'Monthly': 30}
df['Frequency of Purchases'] = df['Frequency of Purchases'].map(frequency_mapping)

# Fill missing values in 'Frequency of Purchases' with mean
df['Frequency of Purchases'] = df['Frequency of Purchases'].fillna(df['Frequency of Purchases'].mean
    ())

# Combine numeric features and 'Frequency of Purchases'
features = numeric_features + ['Frequency of Purchases']

# Standardize the features
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df[features])

# Determine the optimal number of clusters using the Elbow method
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_data)
    sse.append(kmeans.inertia_)

# Plot the Elbow curve to determine the optimal number of clusters
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), sse, marker='o', linestyle='--')
plt.xlabel('Number of Clusters')
plt.ylabel('SSE (Sum of Squared Errors)')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.show()

# Choose the optimal number of clusters (e.g., by visually inspecting the Elbow curve)
optimal_num_clusters = 3

# Perform K-means clustering with the optimal number of clusters
kmeans = KMeans(n_clusters=optimal_num_clusters, random_state=42)
clusters = kmeans.fit_predict(scaled_data)

```

```
# Add cluster labels to the dataset
df['Cluster'] = clusters

# Visualize the clusters in a scatter plot
plt.figure(figsize=(10, 8))
for cluster in range(optimal_num_clusters):
    cluster_data = df[df['Cluster'] == cluster]
    plt.scatter(cluster_data['Frequency of Purchases'], cluster_data['Purchase Amount (USD)'],
                label=f'Cluster {cluster}', alpha=0.7)
plt.xlabel('Frequency of Purchases')
plt.ylabel('Purchase Amount (USD)')
plt.title('Customer Segmentation')
plt.legend()
plt.show()
#predictive modeling
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

# Assuming df is already loaded with your dataset
df = pd.read_csv("/content/shopping_behavior_updated.csv")

# Select relevant features and target variable
features = ['Age', 'Review Rating', 'Frequency of Purchases', 'Gender', 'Season']  # Add categorical
     variables here
target = 'Purchase Amount (USD)'

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df[features], df[target], test_size=0.2,
    random_state=42)

# Define categorical features for one-hot encoding
categorical_features = ['Gender', 'Season', 'Frequency of Purchases']  # Update with your
    categorical columns

# Convert categorical features to one-hot encoding
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
    ],
    remainder='passthrough'
)


X_train_encoded = preprocessor.fit_transform(X_train)
X_test_encoded = preprocessor.transform(X_test)
```

```python

# Train linear regression model
lr_model = LinearRegression()
lr_model.fit(X_train_encoded, y_train)

# Train random forest regression model
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train_encoded, y_train)

# Make predictions on the testing set
lr_predictions = lr_model.predict(X_test_encoded)
rf_predictions = rf_model.predict(X_test_encoded)

# Evaluate model performance
lr_mse = mean_squared_error(y_test, lr_predictions)
rf_mse = mean_squared_error(y_test, rf_predictions)

print("Linear Regression MSE:", lr_mse)
print("Random Forest Regression MSE:", rf_mse)
```

**Output**

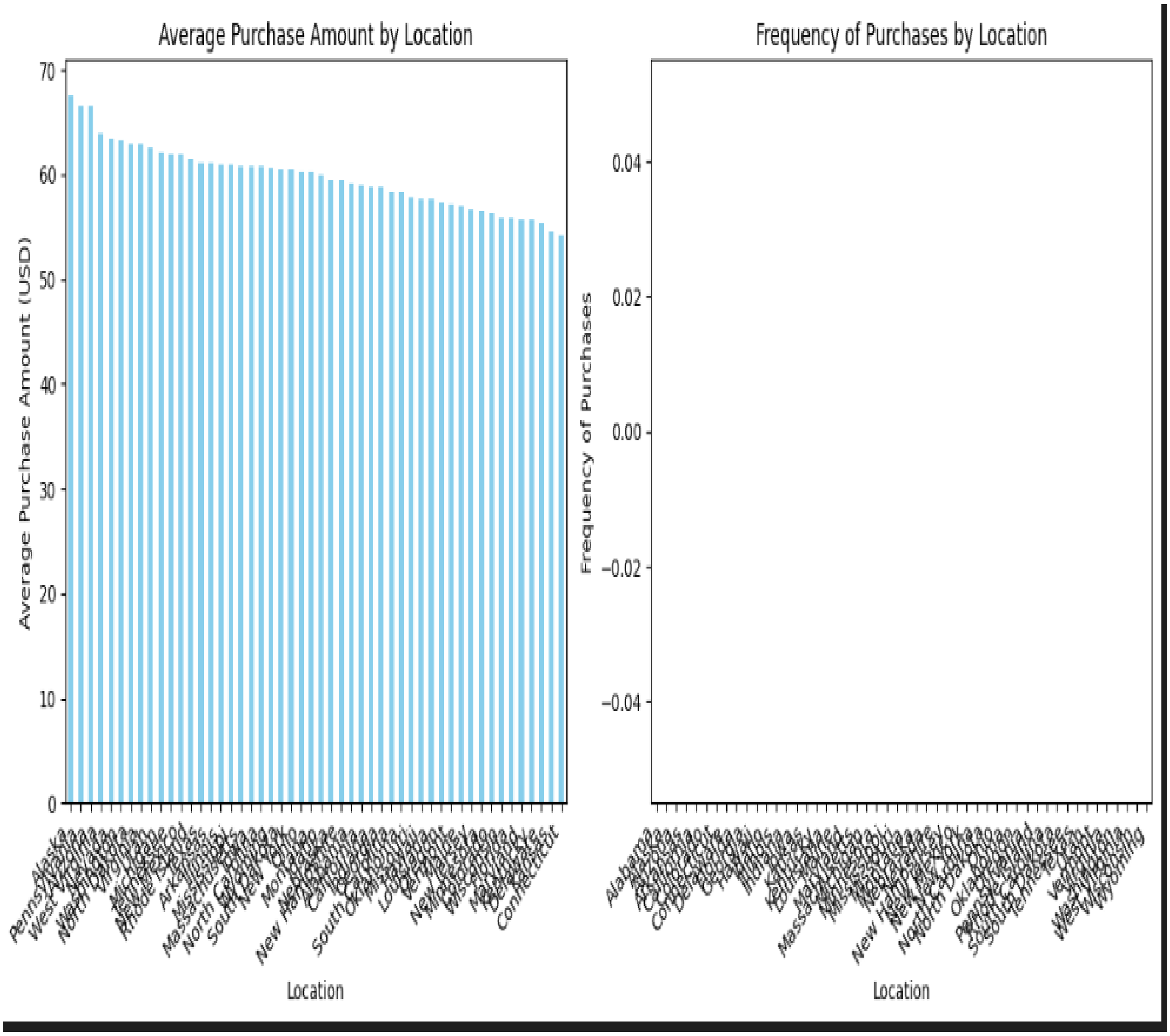



Figure 6.1: **Location Analysis**

Figure 6.1 describes about the Location Analysis of two graphs, one displaying the average purchase amount by location, and the other showing the frequency of purchases by location. The left graph reveals that some locations have higher average purchase amounts compared to others, with a relatively consistent trend across most locations. However, the right graph depicts a flat line close to zero for the frequency of purchases across all locations, suggesting an even distribution of purchase occurrences. This combination of data visualizations allows for an analysis of spending patterns and purchase frequencies across various locations, enabling insights into customer behavior and potential optimization strategies.

**output**

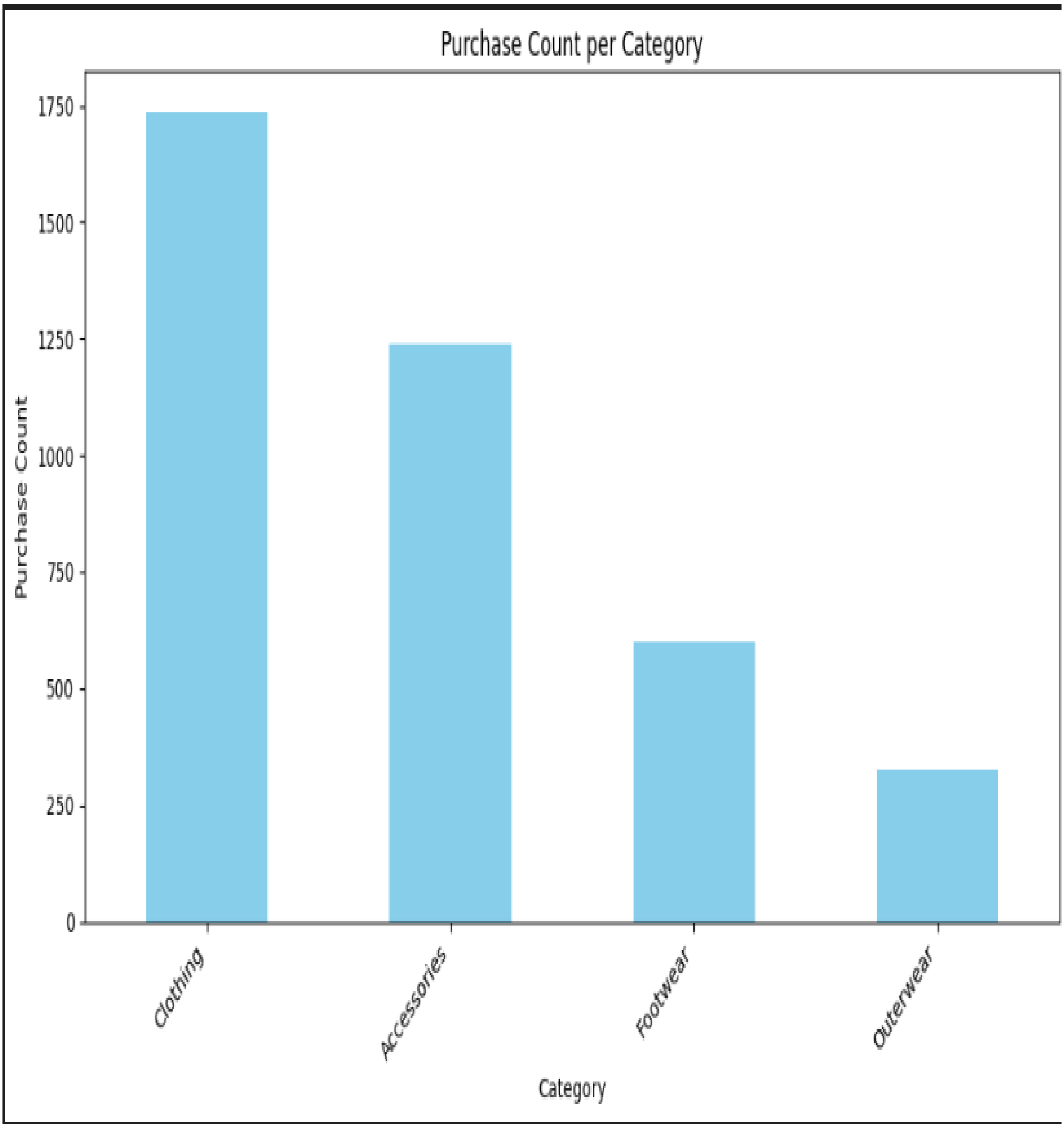



Figure 6.2: **Purchase Count per Category**

Figure 6.2 shows that purchase count per category displays a bar chart that represents the number of purchases made in different product categories. The y-axis shows the purchase count, while the x-axis lists the categories.The category with the highest purchase count is ”Clothing,” with a significantly larger bar compared to the others. This suggests that clothing items are the most frequently purchased products within the given data.The second-highest category is ”Accessories,” followed by ”Footwear” and ”Outerwear,” respectively.

# Chapter 7

# CONCLUSION AND FUTURE ENHANCEMENTS

## 7.1 Conclusion

This project analyzed how seasonality influences customer purchasing decisions by examining a large transaction dataset across different product categories, customer segments, and regions. The analysis uncovered significant seasonal variations in purchasing behavior, with peaks during holidays and for seasonal product lines impacted by climate. Customer demographics like age and location also played a role in shaping seasonal purchase patterns. To explore these insights interactively, a web application was built with dashboards and data visualizations that allow slicing and filtering the data. This application enables businesses to leverage seasonal trends for optimizing inventory, executing targeted marketing campaigns, and driving customer engagement through tailored promotions and recommendations. Future work could integrate demand forecasting models, recommendation engines, and real-time data updates into the application for an even more comprehensive and responsive approach to managing seasonal demand shifts.

## 7.2 Future Enhancements

While the current iteration of the ”Seasons of Shopping Analytical Application” provides valuable insights and a robust platform for businesses to explore and leverage seasonal patterns in customer purchase behavior, several enhancements can be implemented to further augment its capabilities and address evolving business needs.

1. Incorporating advanced demand forecasting models into the application can enable businesses to anticipate future seasonal trends more accurately. By leveraging techniques such as time series forecasting, machine learning, and ensemble methods, the application can generate reliable demand forecasts for different product categories, customer segments, and geographic regions. This capability would empower businesses to optimize inventory levels, production planning, and resource allocation based on predicted seasonal demand.

2. Recommendation Engine: Implementing a recommendation engine within the application can enhance customer engagement and drive personalized product recommendations based on seasonal preferences. By analyzing historical purchase data, customer demographics, and seasonal patterns, the recommendation engine can suggest relevant products, promotional offers, and personalized content tailored to individual customers or customer segments during specific seasons.

3. Real-time Data Ingestion and Updating: Integrating real-time data ingestion and updating mechanisms can ensure that the application remains up-to-date with the latest transaction data and market trends. This enhancement would enable businesses to continuously monitor and adapt to evolving seasonal patterns, allowing for timely adjustments to marketing strategies, inventory management, and operational decisions.

4. Advanced Data Visualization and Exploration: Incorporating advanced data visualization techniques, such as interactive maps, network diagrams, and dynamic dashboards, can further enhance the application's analytical capabilities. These visualizations can provide businesses with a more comprehensive and intuitive understanding of geographic and demographic seasonal patterns, enabling deeper insights and facilitating more informed decision-making processes.

5. Predictive Analytics and Prescriptive Recommendations: Extending the application's functionality to include predictive analytics and prescriptive recommendations can empower businesses with actionable insights. By leveraging machine learning techniques and optimization algorithms, the application can suggest data-driven strategies and prescriptive recommendations tailored to specific business objectives, such as maximizing revenue, optimizing inventory levels, or minimizing operational costs during seasonal peaks or troughs.

6. Integration with External Data Sources: Enabling the application to integrate with external data sources, such as weather data, economic indicators, and social media sentiment, can provide a more comprehensive understanding of the factors influencing seasonal purchase behavior. This integration can uncover new insights and correlations, allowing businesses to adapt their strategies proactively to external factors that may impact seasonal demand.

# Chapter 8

# PLAGIARISM REPORT

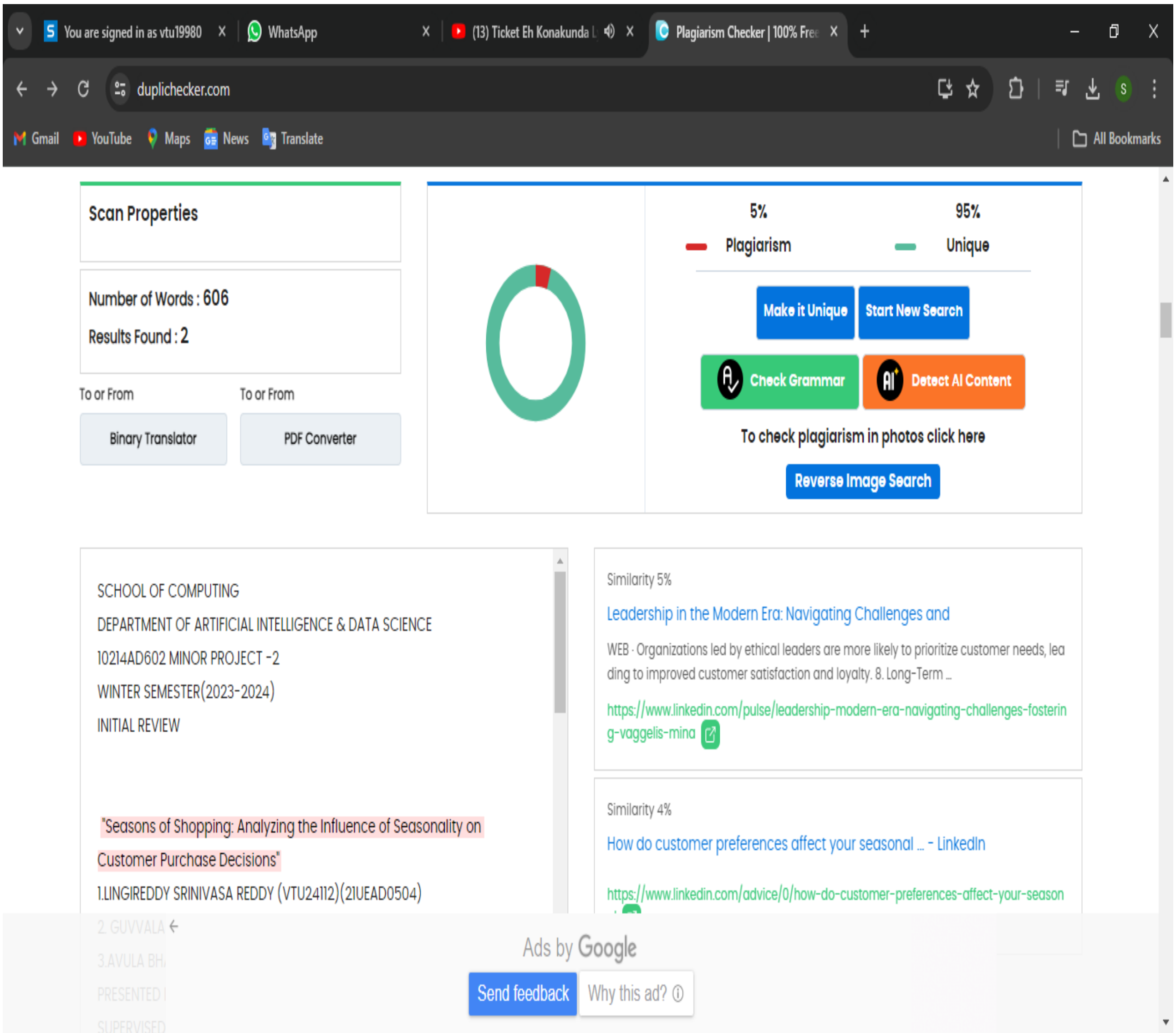



Figure 8.1: **PLAGIARISM REPORT**

# Chapter 9

# SOURCE CODE & POSTER PRESENTATION

## 9.1 Source Code

```
import numpy as np #for mathematical working on arrays
import pandas as pd #helps to work on dataframes or tables
import seaborn as sns #for visualising data
import plotly.express as px #for visualising data
import matplotlib.pyplot as plt #for visualising data
from matplotlib import style
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('/content/shopping_behavior_updated.csv')
df.head()
df.tail()
df.shape
df.columns
print(df)
print('dimensions:')
print(df.shape)
print('Information:')
df.info()
print(df.apply(lambda col: col.unique()))
df.nunique()
df.isnull().sum()
df.dropna(inplace=True)
df.isnull().sum()
numerical_columns = ['Age', 'Purchase Amount (USD)', 'Review Rating', 'Previous Purchases']
descriptive_stats = df[numerical_columns].describe()
print(descriptive_stats)
df['Frequency of Purchases'].head()
df['Frequency of Purchases'] = df['Frequency of Purchases'].astype(object)
df['Frequency of Purchases'] = pd.to_numeric(df['Frequency of Purchases'], errors='coerce')
# Group the data by gender
gender_grouped = df.groupby('Gender')
# Calculate average purchase amount by gender
average_purchase_amount = gender_grouped['Purchase Amount (USD)'].mean()
# Calculate frequency of purchases by gender
```

```python
# Assuming 'Frequency of Purchases' has been preprocessed to numeric format
purchase_frequency = gender_grouped['Frequency of Purchases'].mean()
# Count preferred product categories by gender
preferred_categories = gender_grouped['Category'].value_counts()
# Print the results
print("Average Purchase Amount by Gender:\n", average_purchase_amount)
print("Average Purchase Frequency by Gender:\n", purchase_frequency)
print("\nPreferred Product Categories by Gender:\n", preferred_categories)
# Group the data by season
season_grouped = df.groupby('Season')
# Calculate average purchase amount by season
average_purchase_amount = season_grouped['Purchase Amount (USD)'].mean()
# Count popular items purchased by season
popular_items = season_grouped['Item Purchased'].value_counts().groupby(level=0).nlargest(1)
# Calculate average review ratings by season
average_review_ratings = season_grouped['Review Rating'].mean()
# Print the results
print("Average Purchase Amount by Season:\n", average_purchase_amount)
print("\nPopular Items Purchased by Season:\n", popular_items)
print("\nAverage Review Ratings by Season:\n", average_review_ratings)
# Group the data by category
category_grouped = df.groupby('Category')
# Count purchases per category
purchase_count_per_category = category_grouped.size()
# Sort categories by purchase count in descending order
purchase_count_per_category = purchase_count_per_category.sort_values(ascending=False)
# Plot the results
plt.figure(figsize=(10, 6))
purchase_count_per_category.plot(kind='bar', color='skyblue')
plt.title('Purchase Count per Category')
plt.xlabel('Category')
plt.ylabel('Purchase Count')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
# Group the data by location
location_grouped = df.groupby('Location')
# Calculate average purchase amount by location
average_purchase_amount = location_grouped['Purchase Amount (USD)'].mean()
# Calculate frequency of purchases by location
purchase_frequency = location_grouped['Frequency of Purchases'].mean()
# Calculate total purchases by location
total_purchases = location_grouped.size()
# Visualize the results
plt.figure(figsize=(12, 6))
# Plot average purchase amount
plt.subplot(1, 2, 1)
average_purchase_amount.sort_values(ascending=False).plot(kind='bar', color='skyblue')
plt.title('Average Purchase Amount by Location')
plt.xlabel('Location')
```

```
plt.ylabel('Average Purchase Amount (USD)')
plt.xticks(rotation=45, ha='right')
# Plot frequency of purchases
plt.subplot(1, 2, 2)
purchase_frequency.sort_values(ascending=False).plot(kind='bar', color='salmon')
plt.title('Frequency of Purchases by Location')
plt.xlabel('Location')
plt.ylabel('Frequency of Purchases')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
# Select relevant numerical columns for correlation analysis
numerical_columns = ['Age', 'Purchase Amount (USD)', 'Review Rating', 'Frequency of Purchases']
# Calculate correlation matrix
correlation_matrix = df[numerical_columns].corr()
# Visualize correlation matrix using a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", vmin=-1, vmax=1)
plt.title('Correlation Matrix')
plt.show()
#customer segmentation
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import pandas as pd

# Load the dataset
df = pd.read_csv("/content/shopping_behavior_updated.csv")

# List of numeric features for clustering
numeric_features = ['Review Rating', 'Purchase Amount (USD)']

# Fill missing values with mean for numeric columns
df[numeric_features] = df[numeric_features].fillna(df[numeric_features].mean())

# Convert 'Frequency of Purchases' from string to numerical
frequency_mapping = {'Daily': 1, 'Weekly': 7, 'Fortnightly': 14, 'Monthly': 30}
df['Frequency of Purchases'] = df['Frequency of Purchases'].map(frequency_mapping)

# Fill missing values in 'Frequency of Purchases' with mean
df['Frequency of Purchases'] = df['Frequency of Purchases'].fillna(df['Frequency of Purchases'].mean
    ())

# Combine numeric features and 'Frequency of Purchases'
features = numeric_features + ['Frequency of Purchases']

```

```
# Standardize the features
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df[features])

# Determine the optimal number of clusters using the Elbow method
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_data)
    sse.append(kmeans.inertia_)

# Plot the Elbow curve to determine the optimal number of clusters
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), sse, marker='o', linestyle='--')
plt.xlabel('Number of Clusters')
plt.ylabel('SSE (Sum of Squared Errors)')
plt.title('Elbow Method for Optimal Number of Clusters')
plt.show()

# Choose the optimal number of clusters (e.g., by visually inspecting the Elbow curve)
optimal_num_clusters = 3

# Perform K-means clustering with the optimal number of clusters
kmeans = KMeans(n_clusters=optimal_num_clusters, random_state=42)
clusters = kmeans.fit_predict(scaled_data)

# Add cluster labels to the dataset
df['Cluster'] = clusters

# Visualize the clusters in a scatter plot
plt.figure(figsize=(10, 8))
for cluster in range(optimal_num_clusters):
    cluster_data = df[df['Cluster'] == cluster]
    plt.scatter(cluster_data['Frequency of Purchases'], cluster_data['Purchase Amount (USD)'],
                label=f'Cluster {cluster}', alpha=0.7)
plt.xlabel('Frequency of Purchases')
plt.ylabel('Purchase Amount (USD)')
plt.title('Customer Segmentation')
plt.legend()
plt.show()
#predictive modeling
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

# Assuming df is already loaded with your dataset
```

```
df = pd.read_csv("/content/shopping_behavior_updated.csv")

# Select relevant features and target variable
features = ['Age', 'Review Rating', 'Frequency of Purchases', 'Gender', 'Season']  # Add categorical
    variables here
target = 'Purchase Amount (USD)'

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df[features], df[target], test_size=0.2,
    random_state=42)

# Define categorical features for one-hot encoding
categorical_features = ['Gender', 'Season', 'Frequency of Purchases']  # Update with your
    categorical columns

# Convert categorical features to one-hot encoding
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
    ],
    remainder='passthrough'
)


X_train_encoded = preprocessor.fit_transform(X_train)
X_test_encoded = preprocessor.transform(X_test)

# Train linear regression model
lr_model = LinearRegression()
lr_model.fit(X_train_encoded, y_train)

# Train random forest regression model
rf_model = RandomForestRegressor(random_state=42)
rf_model.fit(X_train_encoded, y_train)

# Make predictions on the testing set
lr_predictions = lr_model.predict(X_test_encoded)
rf_predictions = rf_model.predict(X_test_encoded)

# Evaluate model performance
lr_mse = mean_squared_error(y_test, lr_predictions)
rf_mse = mean_squared_error(y_test, rf_predictions)

print("Linear Regression MSE:", lr_mse)
print("Random Forest Regression MSE:", rf_mse)
```

## 9.2 Poster Presentation

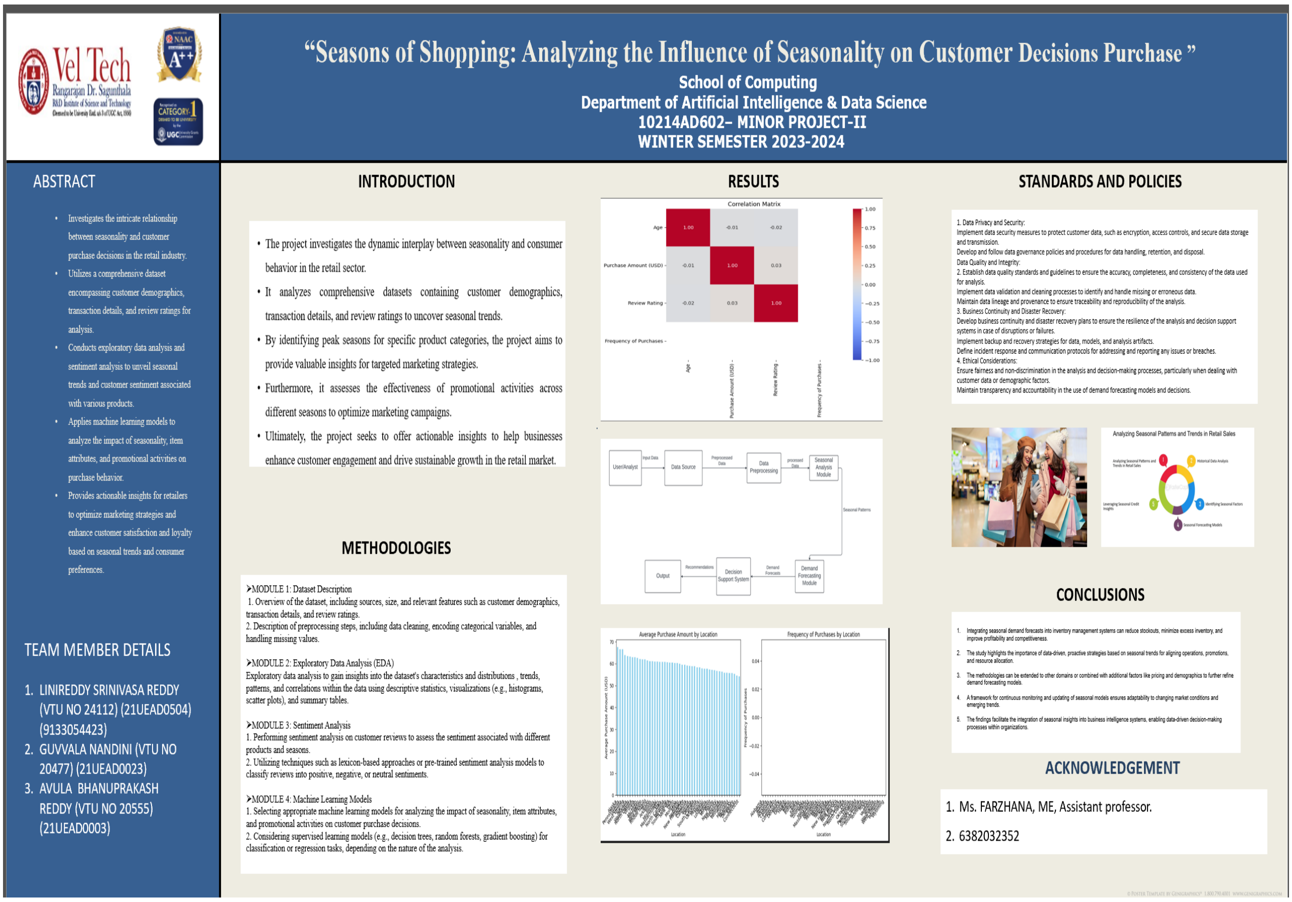



Figure 9.1: **Poster Presentation of Seasons of Shopping: Analyzing the Influence of Seasonality on Customer Purchase Decisions.**

# References

[1] Peterson, A. (2024). Evaluating the Impact of Price Fluctuations on Consumer Buying Decisions and Adaptation Strategies in the Automotive Market. Law and Economy, 3(4), 1-9.

[2] Necula, D., Ungureanu-Iuga, M., Ognean, L. (2024). Beyond the Traditional Mountain Emmental Cheese in "Ţara Dornelor", Romania: Consumer and Producer Profiles, and Product Sensory Characteristics. Agriculture, 14(4), 621.

[3] Rose, N., Dolega, L., Rowe, F. (2024). Using loyalty card data to understand the impact of weather on click collect behaviours in UK retailing. The International Review of Retail, Distribution and Consumer Research, 1-22.

[4] Akhilendra, S. P., Aravendan, M. (2023). A review on apparel fashion trends, visual merchandising and fashion branding. Intelligent Information Management, 15(3), 120-159.

[5] , S. P., Aravendan, M. (2023). Impact of fashion trends on visual merchandising for promoting fashion apparel brands. Journal of Service Science and Management, 16(4), 448-476.

[6] Oh, J. (2023). Classification and regression tree approach for the prediction of the seasonal apparel market: focused on weather factors. Journal of Fashion Marketing and Management: An International Journal, (ahead-of-print).

[7] Oh, J., Ha, K. J., Jo, Y. H. (2022). Use of weather factors in clothing studies in Korea and its implications: A review. Asia-Pacific Journal of Atmospheric Sciences, 58(5), 729-741.

[8] Sajja, S., Aggarwal, N., Mukherjee, S., Manglik, K., Dwivedi, S., Raykar, V. (2021, January). Explainable AI based interventions for pre-season decision making in fashion retail. In Proceedings of the 3rd ACM India Joint International Conference on Data Science Management of Data (8th ACM IKDD CODS 26th COMAD) (pp. 281-289).

[9] Ensafi, Y., Amin, S. H., Zhang, G., Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning–A comparative analysis. International Journal of Information Management Data Insights, 2(1), 100058.

[10] Martínez-de-Albéniz, V., Belkaid, A. (2021). Here comes the sun: Fashion goods retailing under weather fluctuations. European Journal of Operational Research, 294(3), 820-830.

[11] Seyedan, M., Mafakheri, F., Wang, C. (2022). Cluster-based demand forecasting using Bayesian model averaging: An ensemble learning approach. Decision Analytics Journal, 3, 100033.