How Can Advanced Data Science Approaches Uncover Patterns in Consumer Behavior and Guide Strategic Decision-Making in the Fashion Retail Sector?

By
XHESIKA FETO

Septemeber 2023

DECLARATION

I, Xhesika Feto declare that I am the sole author of this Project, that all the references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.
This has not previously been accepted as part of any other degree submission.

Signed: Xhesika Feto
Date: 07/09/2023

FORM OF CONSENT

I, Xhesika Feto, hereby consent that this project, submitted in partial fulfilment of the requirements for the award of the MSc degree, if successful may be made available in the paper or electronic format for inter-library loan or photocopying (subject to the law of copyright) and that the title ad abstract may be made available to outside organisations.

Signed: Xhesika Feto
Date: 07/09/2023

ABSTRACT

In an era where fashion choices speak volumes, deciphering the silent dialogues of customers becomes an art backed by science.. This research delves into customer interactions with fashion products, employing advanced data analytics techniques to distil insights that can drive business strategies. Utilizing a comprehensive dataset, the study implements both Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to transform data for optimized classification. The results underscore the potential of these techniques in predicting metrics such as customer satisfaction, interest, product recommendation likelihood, and seasonal shopping behaviour. By revealing patterns in customer preferences and purchase behaviours, this research provides actionable intelligence for the fashion retail industry, enabling enhanced marketing strategies, improved inventory management, and superior customer experiences.

ACKNOWLEDGEMENTS

My heartfelt gratitude goes to my parents for standing by me during this demanding phase. My partner's encouragement and understanding have been invaluable, and I cannot thank them enough. Lastly, I'm immensely thankful to my supervisor for their expert guidance and unwavering support throughout this journey.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background:

The fashion industry is an ever-changing entity that is influenced by a plethora of socio-cultural and economic factors. A decade ago, fashion aficionados gleaned trends from magazines, films and television. Today, the digital age has proliferated access to fashion, with everyone from celebrities to influencers broadcasting styles in real-time via the internet. With just a few swipes on a smartphone, individuals can shape their personal style narrative, drawing inspiration from global sources.

In this digital ecosystem, the retail fashion industry faces unique challenges. Not only are there diverse categories and novel clothing types emerging constantly, but customer preferences also seem to be shifting at an accelerated pace, sometimes changing multiple times within a single season. Such rapid transformation has consequential impacts, leading to increased inventory unsatisfied customers, and consequently, dwindling profits. In this case, the need for astute trend forecasting, aided by modern technologies like data mining and machine learning, has never been more pronounced.

1.2 Aim:

The retail fashion industry grapples with an overwhelming influx of data resulting from digitalization. The paramount challenge lies in effectively harnessing this data to comprehend and predict swiftly changing customer preferences and behavior, ensuring not only customer satisfaction but also optimal stock management, thereby averting significant losses.

1.3 Purpose of the study:

- To analyse the impact of modern technologies, particularly data mining and machine learning in predicting customer references in the fashion industry,
- To understand the significance of customer relationship management in driving sales and overall business strategy within the sector.
- To evaluate and compare the effectiveness of PCA and LDA in dimensionality reduction for the retail fashion dataset and understand their implications on classification tasks.

1.4 Significance of the study:

In an era marked by heightened competition and digital transformation, understanding customer behaviour through technological tools is no longer optional but imperative for survival. As per Dennis, Marsland, and Cockett (2001), target marketing strategies based on segmented customer data gave showcased the potential to bolster sales and profitability. Furthermore, given the relations of Brandther et al. (2021), the current global challenges, like the Covid-2019 pandemic, have further underlined the importance of leveraging big data and advanced analytics to maintain a competitive edge,

1.5 Scope of the study:

This study will delve into:

- The role of data analysis, cleaning, and preprocessing in preparing retail fashion datasets.
- An exploration of Market Basket Analysis using the Apriori Algorithm.
- The application and comparison of PC and LDA for dimensionality reduction.
- The efficiency of classification algorithms in predicting customer satisfaction, interest, product recommendation likelihood, and seasonal buying behaviour.

1.6 Dissertation Structure:

Following this introductory chapter, the dissertation will encompass a literature review, detailing prior research and methodologies in the realm. Subsequent chapters will address the methodology adopted in this study, data analysis, findings, and finally, the conclusions and recommendations.

2. LITERATURE REVIEW

Fashion has always been changing, but now it is changing faster than ever. In the past, we looked to magazines and movies to see the latest trends. Today, with the internet and social media, new styles and trends can become popular overnight. This fast-paced change presents a challenge for the fashion retail industry, Trends can change multiple times within a season, making it hard for retailers to keep up. This can lead to too much or too little stock and unhappy customers.

But there is a bright side. The digital age provides a lot of data. Every time a customer clicks, buys, or shares something online, it gives retailers a clue about what people want. By using modern technology like data mining and machine learning, retailers can make sense of this data. They can better understand and predict what customers want, helping them manage stock better and keep customers happy.

This literature review will explore the relationship between fashion and data. This paper will explain how fashion reflects society, will talk about the importance of planning in retail, how trends are predicted using classification, and how can data help retailers understand their customers better, Throughout the review, a detail explanation of those topics will be conducted and it will show how data can help to shape the future of fashion retail.

2.1 FASHION AND SOCIOCULTURAL FACTORS

Fashion, as an industry, is an ever-evolving phenomenon deeply intertwined with the fabric of society. It serves as a mirror reflecting the collective consciousness of culture, economy, and individual expression. Its fluid nature is a reflection of myriad forces at play, from economic shifts to sociocultural movements (Nambisan et al., 2017). The symbiotic relationship between fashion and society implies that as societal norms, values, and preferences evolve, so does fashion. Hence, businesses are not merely tracking hemlines but are decoding societal transitions.

Forecasting, in this context, becomes less about predicting the color of the season and more about understanding global sociocultural shifts. The longevity of trends, for instance, is not determined by designers in isolation but is co-created with consumers, influenced by global events, cultural nuances, and economic realities. As forecasters delve into cultural differences, they uncover a rich tapestry of consumer lifestyles, preferences, and demographics that shape fashion trajectories. The fashion ecosystem, thus, is not a monologue dictated by brands but a vibrant dialogue with consumers, where listening becomes as imperative as creating (Brandther et al., 2021).

While the fashion ecosystem is influenced by the global sociocultural shift and customer dialogues, the tangible manifestation of these influences can be seen most clearly in the realm of retail product development, where these trends come to life.

2.2 RETAIL PRODUCT DEVELOPMENT

In the context of fashion, retail is the final act – the tangible touchpoint where consumers interact with fashion. But beneath the glitz of store lights lies a meticulous process of product development, a journey from concept to closet. Retail product development, as a specialized division within the broader retail framework, operates at this intersection of creativity and commerce (Jedid-Jah-Jonder, 2004).

This division's tasks transcend mere product creation. It shoulders the onus of ensuring that products not only align with current fashion sentiments but also anticipate future trends. In an era where consumer preferences can shift overnight, powered by social media influencers or global events, the role of retail product development becomes even more challenging and crucial. The division needs to be agile, responsive, and deeply attuned to both global fashion movements and local consumer

preferences. Building on the significance of retail product development, predicting fashion trends becomes crucial. This brings us to elaborate on the process of fashion forecasting.

2.3 THE FASHION FORECASTING PROCESS

Fashion forecasting is an essential part of the fashion world. This process typically begins about two years before the selling season. It has three main steps which include: studying the environment, researching the market and analysing the products as shown in Figure 1 (Kim et al. 2021)

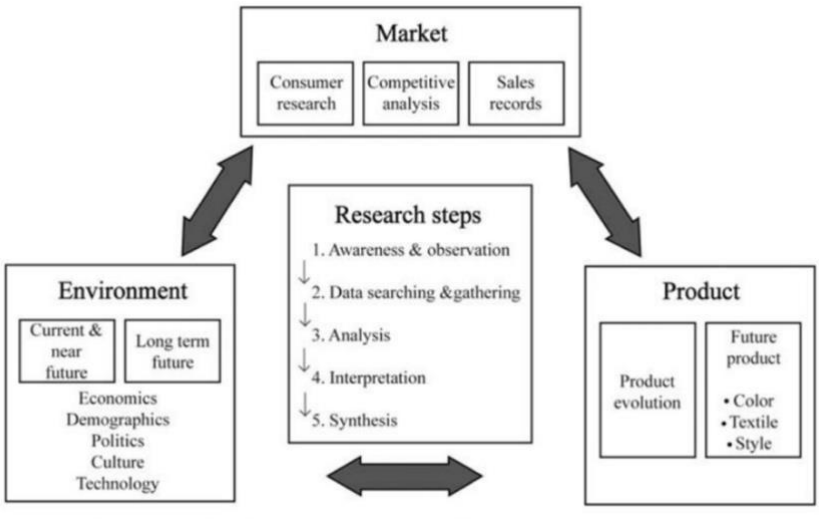


Figure 2.1 A model of how fashion trends are analysed and predicted. Source by Eundeok Kim.

Environmental Scanning: The first step involves studying broad topics like culture, economy, politics and society (Kim et al. 2021). It is important because it helps fashion experts to predict and respond to new trends. By understanding this bigger picture, businesses can make sure their products match what people will want in the future.

Market Research: Next, experts look into customer habits, competitors, and past sales (Kim et al. 2021). This step helps businesses figure out what customers like and how they spend. By looking at past sales, they can spot trends and plan future products. It also helps businesses stand out from their competitors.

Product Analysis: The last step involves closely examining past products. This helps experts predict future colours, materials, and styles. Looking back at older fashion trends helps in guessing what might come back into style. Techniques like Principal Component Analysis (PCA) help in this step by focusing on the important details and leaving out the unnecessary ones (Salih Hassan et al. 2021).

In summary, fashion forecasting is about understanding current styles and predicting future ones. It helps businesses plan products and marketing strategies that people will love. After all this, it's clear that fashion forecasting produces a lot of data. The next challenge is to make sense of this data, which is where data mining and understanding customer behaviour become crucial.

2.4 DATA MINING AND CUSTOMER SATISFACTION

The infusion of data science into retail has unlocked new avenues for understanding consumer behaviour. One such avenue is data mining, a technique that leverages programming methods to discern patterns among vast datasets. An illustrative example is the association between seemingly unrelated products in a shopping basket. For instance, a pattern revealing that fathers frequently purchase beer alongside nappies could guide targeted marketing strategies, elevating both sales and customer satisfaction (Dennis, Marsland, and Cockett, 2001).

Moreover, with the advent of digital platforms, consumer feedback has become abundant and accessible. Analyzing consumer satisfaction data, such as ratings and evaluation comments, offers retailers invaluable insights. A study examining the impact of COVID-19 on consumer satisfaction in Austrian retail chains revealed a significant decline in satisfaction levels during the pandemic. Factors like store layout, product availability, and waiting times were found to significantly influence consumer sentiment (Brandtner et al., 2021). Such insights, derived from data mining, enable retailers to adapt swiftly, ensuring sustained consumer satisfaction even in turbulent times.

With insights gained from data mining, it is evident that a system is needed to organize vast product ranges, thereby ensuring that insights are applied effectively

2.5 PRODUCT CATEGORISATION IN RETAIL

In the vast expanse of retail, where a multitude of products vie for consumer attention, the act of product categorization emerges as both an art and a science. The retail landscape is dotted with products, each with its unique set of attributes, catering to diverse consumer segments. In such a scenario, the ability to adeptly classify products becomes paramount, not just for operational efficiency but also for effective consumer engagement (Holý, Sokol, and Černý, 2017).

Product categorization hinges on a blend of quantitative and qualitative characteristics. Quantitatively, products might be categorized based on metrics like sales volume, profitability, or inventory turnover. Qualitatively, attributes such as brand perception, design uniqueness, or cultural relevance come into play. An astute categorization strategy acknowledges that products are not mere commodities; they carry with them stories, emotions, and aspirations.

The implications of product categorization extend beyond internal operations. From a consumer's perspective, categorization aids in product discovery, decision-making, and overall shopping experience. For instance, a well-categorized e-commerce platform can expedite the search process, leading to quicker purchase decisions and enhanced user satisfaction. On the flip side, a cluttered product landscape can overwhelm consumers, leading to decision fatigue and potential loss of sales.

Moreover, with the advent of machine learning, product categorization has transcended manual boundaries. Algorithms can now predict optimal categorizations based on historical data, current market trends, and forecasted consumer behavior. Such automated systems, while efficient, need to be employed judiciously, ensuring that the human touch in product categorization isn't entirely lost (Jedid-Jah-Jonder, 2004).

As products are adeptly categorized and the retail landscape becomes increasingly data-rich, therei a pressing need to manage and simplify this data. This brings us to the domain of dimensionality reduction and its synergy with machine learning.

2.6 DIMENSIONALITY REDUCTION AND MACHINE LEARNING

The increasing reliance of the fashion industry on data-driven decision-making has prompted a deeper exploration of dimensionality reduction techniques to streamline data complexity. Principal Component Analysis (PCA), as highlighted by G.T. Reddy et al. (2020), has emerged as a robust method for achieving this. PCA reduces data dimensionality while preserving essential information by creating new uncorrelated variables known as principal components. These components capture the most significant variations in the data, thus facilitating better interpretability and modelling. The adoption of dimensionality reduction techniques like PCA ensures that machine learning algorithms can process data more efficiently and effectively.

A notable comparison within dimensionality reduction techniques lies between PCA and Linear Discriminant Analysis (LDA), both of which play critical roles in the fashion industry's analytics landscape. PCA aims to maximize variance within the data, making it an effective tool for data simplification and visualization. It creates a set of orthogonal axes that capture the highest variance in the dataset, allowing for dimensionality reduction while retaining essential information.

In contrast, LDA focuses on class separation and is particularly suited for classification tasks. LDA finds the linear combinations of features that best separate different classes in the data. In the context of fashion analytics, LDA can be employed when the goal is not only to reduce dimensionality but also to enhance the discriminative power of the features for classification tasks like customer segmentation or product recommendation.

Machine learning, specifically supervised and unsupervised learning algorithms, has become pivotal in fashion analytics. Supervised learning models enable businesses to predict outcomes like customer satisfaction or purchase likelihood based on historical data and relevant features. Unsupervised learning techniques, on the other hand, uncover hidden patterns within vast datasets, offering valuable insights into consumer behavior, product associations, and market trends. Algorithms like Support Vector Machines (SVM), Decision Trees, and Random Forests have proven particularly effective in fashion prediction tasks. Harnessing the power of machine learning empowers fashion enterprises to make data-driven decisions, from personalized recommendations to inventory optimization, ultimately elevating customer satisfaction.

Dimensionality reduction and classification together give retailers powerful tools. The first makes data easier to handle, and the second uses this data to make educated predictions. As retail changes, the combination of these methods with traditional retail knowledge will play a big role in its future. While the potential of dimensionality reduction and machine learning is vast, integrating them into the fashion industry isn’t without challenges. Understanding these challenges and envisioning the road ahead is crucial.

2.7 CHALLENGES AND FUTURE DIRECTIONS

While the integration of data analytics and machine learning holds immense promise for the fashion industry, it is not without its challenges. One such challenge, as noted by Howkins (2004), is the curse of dimensionality (Farzana Anowar et al. 2021). As the fashion industry generates an ever-increasing volume of data, models trained on many features can become overly reliant on data, leading to overfitting and poor generalization on unseen data. Future directions in fashion analytics must address this challenge by exploring advanced techniques for feature selection, dimensionality reduction, and model regularization to improve model robustness.

Moreover, as fashion companies continue to embrace digital transformation and data-driven strategies, data privacy and ethical considerations have come to the forefront. Striking a balance between data-driven personalization and respecting consumer privacy is an ongoing challenge. Future research should delve into privacy-preserving machine learning techniques and explore frameworks for transparent data usage that build trust with consumers.

In conclusion, the fashion industry's journey towards data-driven decision-making, personalization, and seasonal adaptability is fueled by dimensionality reduction and machine learning. These technologies offer a pathway to enhance customer satisfaction and drive business success. However, they come with challenges, including the curse of dimensionality and ethical considerations, which require ongoing attention and innovation. As the fashion industry navigates these challenges, future directions in fashion analytics will undoubtedly focus on enhancing data-driven strategies while upholding privacy and ethical standards.

3. METHODOLOGY

As mentioned in the previous chapters, this study looks into how machine learning and data science can help understand customer habits in fashion retail. The main goal is simple: to use data to predict things like customer satisfaction, interests, product recommendations, and popular trends for different seasons. In today's digital age, there is a lot of data available, and fashion retailers often find it challenging to make sense of it all. This research combines traditional retail knowledge with modern data tools of dimensionality reduction like Principal Component Analysis and Linear Discriminant Analysis.
The aim is not only to add to the academic discussion about fashion and data but also to provide practical insights for the fashion industry.

The research starts with collecting a dataset that captures various customer behaviours in fashion. This is followed by an exhaustive exploration phase, where the data is visualized to discern latent patterns and prevailing trends. Recognizing the imperfections inherent in any dataset, a rigorous data cleaning process is instituted to rectify anomalies and address voids in the data. Feature engineering then takes precedence, where the data is refined and augmented to bolster its predictive potency. The journey progresses with the application of market basket analysis to discern patterns of product affiliations and purchasing tendencies. Subsequently, dimensionality reduction techniques, namely PCA and LDA, are harnessed to optimize the dataset, setting the stage for the application of machine learning. Various classifiers are then trained, and evaluated, and their performances juxtaposed. The culmination of this analytical odyssey is a comparative assessment, where models undergirded by PCA and LDA are pitted against each other to ascertain the superior approach.

3.1 DATA COLLECTION AND UNDERSTANDING

3.1.1. Data Collection Source

The dataset utilised in this paper was downloaded from Kaggle, a platform known for hosting a plethora of datasets catering to diverse analytical needs. It is worth noting that while an authentic dataset from a real fashion company would be ideal for such a study, there are inherent limitations in accessing proprietary data due to privacy concerns and competitive advantages. Given these constraints, a mock fashion dataset was deemed appropriate for the research. This dataset, though simulated, mirrors the intricacies and nuances one would expect in a genuine fashion retail dataset, making it a suitable candidate for this study.

3.1.2 Dataset Overview

The dataset encompasses a broad spectrum of attributes pertinent to the fashion retail domain. With 29,730 entries, the dataset is comprehensive, capturing a myriad of features related to both products and consumer behaviours.

- **Product Specific Features:**

Product Name	This serves as an unique identifier offering insights into the vast range of products.
Price	The cost of each product
Brand and Category	These offer insights into product segmentation within the fashion domain.
Description and Rating	Qualitative and quantitative assessments of products.
Review Count	An indicator of product popularity and consumer engagement.
Style Attributes	Showing the aesthetic and functional attributes of products.
Sizes and Colors	Vital product specification influencing purchasing decisions.
Season	Providing context on the seasonal relevance of products.

- **Consumer Behaviour Features:**

Purchase History	A qualitative metric, showing the product purchase frequencies
Age	A demographic indicator for understanding consumer segments.
Fashion Preferences	Attributes like Fashion Magazines and Fashion Influencers unveil consumer inclinations in the fashion context.
Time Period Highest Purchase	Metric that shows the purchasing patterns and behaviours.
Feedback metrics	Feedback, Customer Reviews and Social Media Comments offer a window into customer segments and perceptions.

3.2 Data Exploration and Visualisation

Before going deep in the analysis, it is important to understand and be familiarised with the dataset. Data exploration and visualisation serve as the initial steps in this enlightening journey, preparing the stage for more advanced analytical processes. Here is why they hold such significance:

- **Summary Statistics and Data Types:**
  Before any transformation or modelling, it is essential to understand what kind of data we are dealing with. By examining data types, we understand whether an attribute is categorical, numerical or timestamp. This recognition aids in determining the kind of operations or transformations that might be applicable or beneficial. Also, it is important to check for descriptive insights, in this case, summary statistics, such as mean, median, standard deviation and quartile values because they offer a clear view of the dataset. They provide an understanding of the data distribution, variability, and central tendencies, which can be instrumental in formulating initial hypotheses or identifying potential anomalies.

- **Missing and Duplicate Values:**

  If missing values are left unchecked, they can distort analytical outcomes, leading to unreliable or skewed results. Identifying and rectifying these gaps ensures that the subsequent analysis is grounded in complete and trustworthy data.
  On the other hand, duplicate values introduce redundancy into the dataset. They can artificially inflate certain metrics or patterns, leading to misinterpretations. Identifying and removing such duplicates ensures that each entry in the dataset is unique and contributes genuine information.

3.2.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a key step when working with data. It helps data experts dig into data sets, highlight their main features, and often uses charts and graphs to do so. EDA helps decide how best to handle data to find the needed answers. Through EDA, experts can find trends, and unusual data points, test ideas, or check basic beliefs about the data (ibm, 2023).

This process helps to better understand the data and how different parts of it relate to each other. It can also help decide if the chosen methods for studying the data are suitable. The idea of EDA began in the 1970s, thanks to an American expert named John Tukey, and it's still widely used today. When using EDA, it's a bit like taking a first look at the data from a study. It helps in spotting mistakes, checking basic beliefs, choosing suitable models, understanding how different data parts relate, and getting a

rough idea of how data variables influence each other (Chapter 4 Exploratory Data Analysis, n.d.). In simple terms, John Tukey once said that EDA is a lot like detective work, meaning it's about looking at data closely before deciding on a specific way to study it (Exploratory Data Analysis, 2019).

3.3 Data Cleaning

In the data exploration and understating phase, the presence of missing and duplicate values was identified, which are common challenges in data analysis. Ensuring the cleanliness and reliability of the data is important, as any inaccuracies or inconsistencies can lead to misleading results. Therefore in the data cleaning process, it is important to address there issues in an effective way.
Missing data can arise from various sources, such as human error during data entry, system glitches, or even due to the nature of data collection. Methods to deal with missing data include:

- Deletion: One method is to simply remove the rows with missing values, especially if the number of rows that contain missing values is minimal. This method is called listwise deletion. It might lead to a loss of valuable information if many rows are deleted.
- Mean/Mode/Median imputation: For numerical variables, missing values can be replaced with the mean or median of the entire column, For categorical variables, the mode can be used for imputation.
- Predictive models: These are advanced methods that involve using machine learning models like regression or k-nearest-neighbour to predict and impute missing values.

Duplicate values can occur due to various reasons such as data entry errors or merging datasets from different sources. Some methods to deal with duplicate data include:

- Identification: Before removing any duplicate rows, it is important to identify them. This involves comparing rows based on all or specific columns.
- Removal: Once the duplicate values are identified, they can be safely removed to ensure each entry in the dataset is unique.
- Avoidance: Implementing stricter data entry measures or using unique identities for each row can prevent the occurrence of duplicates in the future.
-

In essence, the data cleaning phase ensures that the dataset is polished, reliable, and ready for further analysis.

3.4 Feature Engineering

Feature engineering is a key step in machine learning. It's about changing and fine-tuning the data to make machine-learning models work better and give more accurate results. It is like tweaking ingredients in a recipe to make a dish taste better. The success of this step often depends on two things: understanding the problem you're trying to solve and knowing your data well. In other words, it's not just about being good with numbers and tech, but also about understanding the bigger picture of what you're working on. By doing good feature engineering, the machine learning model can better pick up patterns and make better predictions (domino.ai, n.d.).

In this paper, features engineering involves creating new features from existing ones, transforming features, and selecting the most relevant features to enhance the model’s predictive power. The methodology used in the Feature Engineering section involves:

1. **Identifying unique values of categorical features:**
   Before transforming or creating interactions, understanding the unique values of categorical variables is essential. It provides clarity on the diversity and range of the categorical columns, informing decisions about potential transformations, encodings, or even feature eliminations.

2. **Creating interaction features:**
   Developed features like Price_rating_interaction and Review_rating_interaction to capture the relationship between price and rating, and review and rating, respectively. Interaction features can sometimes capture complex relationships that individual features might miss. For instance, understanding how price and rating interact can provide insights into the perceived value of a product.

3. **Crating Age Groups using binning:**
   Transformed the continuous 'Age' feature into a categorical one with labels like 'Young adults', 'Millennials', 'Gen X', and 'Boomers'. Binning can help reduce the noise associated with minor fluctuations in age and can highlight broader generational trends. Different age groups might have varying buying behaviours, preferences, or brand loyalties.

4. **Sentiment analysis**
   Using TextBlob the sentiment polarity of features like 'feedback', 'customer reviews', and 'social media comments' was calculated. Understanding customer sentiment is important. Positive or negative sentiment can provide insights into product quality, customer service, or overall satisfaction, which can be used to predict future purchasing behaviour or brand loyalty. TextBlob is a Python library used for processing textual data. It provides a simple interface for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. TextBlob can evaluate a piece of text and classify its sentiment as positive, negative, or neutral. Additionally, it quantifies polarity (ranging between -1 and 1) and subjectivity (ranging between 0 and 1).
   Why it's important: Understanding the sentiment of a text can be crucial for numerous applications like gauging customer feedback, analyzing social media sentiment, or tracking brand reputation. The polarity score gives an idea of the sentiment's direction (positive or negative) while subjectivity hints at how subjective or objective the statement is.

5. **One hot encoding and Mapping**
   Transformed categorical columns like 'Brand', 'Category', 'Style', and 'Color' into a binary matrix representation. Additionally, performed mapping for 'description', 'purchase history', and 'season'. Machine learning models require numerical input. One-hot encoding and mapping are ways to convert categorical data into a format that can be provided to machine learning algorithms to improve prediction accuracy.

6. **Creating target features for classification**
   New features, such as 'Satisfaction', 'Is_Satisfied', 'Interested', 'Has_Purchased_Category', 'Frequent_Season', and 'Is_Holiday_Shopper', were created, intended to be used as targets for various classification tasks. Defining clear target variables is essential for supervised learning. These features represent actionable insights. For instance, understanding who is a 'Holiday Shopper' can inform targeted marketing campaigns during the holiday season.

In summary, feature engineering is about enhancing the raw data to make it more suitable and informative for modelling. Properly engineered features can significantly boost a model's performance and the insights drawn from it.

3.5 Market Basket Analysis

Market Basket Analysis is a powerful tool used to discover relationships between items in large datasets, such as products in orders or ingredients in recipes. The main idea is to determine sets of

items that frequently occur together. For example, if a person buys bread, they might also buy butter. Retailers can use these insights to place products together in a way that can increase sales.

In this dataset, MBA can be used to identify patterns like:

- Which clothing items are bought together?
- Do certain accessories align with specific apparel choices?
- Are there seasonal combinations that emerge?

By identifying these associations, retailers can make more informed decisions about marketing, product placements, and even inventory stocking. Before applying, an MBA, the data needs to be in a particular format: the transactions should be in rows with items in columns. For this reason, features like Brand, Category, Style attributes etc. were one-hot encoded as binary.

When performing Market Basket Analysis, one of the most popular algorithms is the Apriori algorithm. It aims to identify frequent item sets (sets of items that have a minimum support threshold) and then forms association rules from these item sets based on a certain confidence level.

3.5.1 Apriori metrics:

- **Support:** It is the relative frequency of the itemset in the transaction dataset.
- **Confidence:** Given two items, A and B, confidence measures the percentage of times that item B is bought, given that item A was bought. It is the conditional probability P(B|A)
- **Lift:** It is the ratio of the observed support to that expected if the two rules were independent. A lift of 1 means there is no association between items. A lift greater than 1 means items are more likely to be bought together. Less than 1 means items are less likely to be bought together.

$$Rule\ X \Longrightarrow Y \begin{cases} Support = \frac{Frequency\ (X,Y)}{N} \\ Confidence = \frac{Frequency\ (X,Y)}{Frequency(X)} \\ Lift = \frac{Support}{Support(X) * Support(Y)} \end{cases}$$

Equation 3.5: Apriori Metrics

Before applying the Apriori algorithm, the data needs to be encoded into a 1-hot encoded format because the algorithm works on binary data. This means each item needs to be represented as a separate column, and the presence or absence of the item in a transaction is represented as 1 or 0.

For a fashion retail dataset, understanding patterns in purchasing behaviour is crucial. If we know that certain clothing items are frequently bought together, it can inform decisions like bundled offers, or even in-store placements. Apriori helps in uncovering these patterns by identifying sets of items that frequently co-occur in transactions.

To sum up, the combination of Market Basket Analysis using the Apriori algorithm provides a robust method to discover hidden patterns in transaction data, which can lead to actionable insights for retailers. This dataset, representing fashion retail transactions, stands to benefit immensely from such insights, allowing for more strategic decisions around product offerings, promotions, and store layouts.

3.6 Dimensionality Reduction

In many real-world datasets, the number of features or variables can be vast. While more data often provides a richer representation, it can also introduce challenges:

- Computational Complexity: More features mean more computational resources are needed for data processing and modelling.

- Overfitting: With more features, models might overfit to the training data, capturing noise and reducing their generalization capability.
- Redundancy: Not all features are informative. Some might be redundant or irrelevant to the task at hand.

To address these challenges, dimensionality reduction techniques are employed. They transform the original features into a smaller set of features while retaining as much of the original data's variance or structure as possible. This not only makes computations faster but can also lead to more interpretable and generalizable models.

3.6.1 Principal Component Analysis (PCA):
PCA is a linear technique used for dimensionality reduction and feature extraction. It works by identifying the 'directions' or 'principal components' in the data that maximize variance. PCA identifies the axes in the dataset that maximize variance. The first principal component (PC1) captures the most variance, the second principal component (PC2) captures the second most, and so on. These components are orthogonal, meaning they're independent of each other.
PCA reduces the number of features while retaining most of the data's variance and often improves the performance of machine learning models by reducing overfitting. It is worth mentioning that it helps in visualizing high-dimensional data.

Given the multitude of features in the fashion dataset, using PCA can streamline the data, making it more manageable and potentially improving the predictive accuracy of subsequent models.

3.6.2 Linear Discriminant Analysis (LDA):
While PCA is unsupervised (it doesn't consider class labels), LDA is supervised. LDA aims to find the feature subspace that best separates different classes in the data. LDA seeks to maximize the distance between the means of different classes while minimizing the scatter (or variance) within each class. It's particularly suitable for classification tasks. It enhances class separability, potentially improving classification performance. Like PCA, it reduces dimensionality, making data more manageable and mitigating overfitting.

Given that the dataset involves predicting aspects like customer satisfaction and product recommendations, LDA can be pivotal. By focusing on class separability, it can enhance the performance of classifiers and provide insights into features that are most discriminative between different classes.

In summary, while both PCA and LDA reduce dimensionality, their mechanisms and objectives differ. PCA focuses on capturing maximum variance, irrespective of class labels, while LDA seeks to enhance the separability between distinct classes. Given the dataset's nature, employing both techniques and comparing their efficacy can provide a holistic understanding of the data and the best approaches for subsequent predictive modelling.

3.7 Modelling and Predictive Analysis

3.7.1 Logistic Regression:
Logistic Regression is a foundational classification technique that predicts the probability of a binary outcome based on one or more predictor variables. For the dataset used in this paper, it provides a baseline model, giving insights into how each feature affects the likelihood of outcomes such as customer satisfaction or product recommendation. Its simplicity, interpretability, and efficiency make it a starting point in many classification problems.

3.7.2 K-Nearest Neighbours (KNN):
KNN is a non-parametric, lazy learning algorithm. When a prediction is required for an unseen data instance, the KNN algorithm searches through the training dataset for the 'k' most similar instances. The prediction attribute of the most similar instances is summarized and returned as the prediction for the unseen instance. KNN can capture non-linear patterns and relationships in predicting outcomes like seasonal trends or customer interests.

3.7.3 Support Vector Machines (SVM):
SVM is a powerful linear classifier that works by finding the hyperplane that best separates the classes in the input feature space. When the data isn't linearly separable, SVM uses a kernel trick to transform it into a higher-dimensional space where a separating hyperplane can be found. Given its capacity to manage high-dimensional data, it's apt for the fashion dataset, especially after feature engineering and dimensionality reduction.

3.7.4 Naive Bayes:
This is a probabilistic classifier based on Bayes' theorem, making an assumption of independence among predictors. Despite its simplicity and the 'naive' assumption, it can be surprisingly effective and is especially suitable for large datasets. In the context of this fashion data set, it can be particularly useful for predicting outcomes based on text data, like feedback or reviews.

3.7.5 Decision Trees:
Decision Trees split the data into subsets based on the value of input features. This process is repeated recursively, resulting in a tree-like model of decisions. Decision Trees can provide a clear and visual structure of decision-making, highlighting the most important features leading to outcomes like purchase likelihood or product recommendations.

3.7.6 Random Forest:
Random Forest is an ensemble method, building multiple decision trees during training and outputting the mode of the classes for classification. It's known for its high accuracy, ability to handle large data sets with higher dimensionality, and its ability to handle missing values. Given the complexity and richness of the dataset, Random Forest can provide robust predictions and insights into feature importance.

After employing these classifiers, their performances are compared primarily using accuracy. Accuracy provides a straightforward metric to gauge how well each model predicts the desired outcomes. By juxtaposing the accuracy of these classifiers, especially after dimensionality reduction techniques like PCA and LDA, a comprehensive understanding of the best approach for this dataset can be garnered. This comparative study not only guides the selection of the optimal model but also deepens the understanding of the dataset's nuances and complexities.

3.8 Model evaluation and validation

3.8.1 Cross-Validation:
Cross-validation is a technique used to assess how the results of a statistical analysis will generalize to an independent dataset. One of the most common methods is k-fold cross-validation, where the original sample is randomly partitioned into 'k' equal-sized subsamples.
The cross-validation process is repeated 'k' times, with each of the 'k' subsamples used exactly once as the validation data. The results from the folds can then be averaged to produce a single estimation.

3.8.2 Confusion Matrix:
A confusion matrix is a table layout that allows visualization of the performance of an algorithm. It provides insights into the true positives, true negatives, false positives, and false negatives. This matrix forms the foundation for various metrics, such as precision, recall, F1-score, and specificity.

3.8.3 ROC Curve and AUC:
Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The Area Under the Curve (AUC) represents a measure of a model's ability to distinguish between the classes. An AUC of 1 indicates a perfect classifier, while an AUC of 0.5 represents a worthless classifier.

3.8.4 Feature Importance:
Especially for ensemble methods like Random Forest, it's essential to gauge the importance of each feature in predicting the outcome. Feature importance helps in understanding which attributes are the most influential in predicting the target variable, shedding light on the dataset's underlying structure and the factors driving the predictions.

3.9 Optimization and Fine-tuning
Once the models have been trained and evaluated, the next step involves optimizing and fine-tuning them to enhance their performance.

3.9.1 Hyperparameter Tuning:
Most machine learning algorithms come with a set of parameters that need to be set before training. These parameters, called hyperparameters, can profoundly influence the training process and, consequently, the model's performance. Techniques like grid search and random search can be employed to find the optimal hyperparameters for a given model.

3.9.2 Feature Selection:
While feature engineering involves creating new features, feature selection is about selecting the most important features and discarding the rest. Reducing the feature space can sometimes lead to better model performance by reducing the risk of overfitting.

3.10 Implementation and Deployment
The final phase involves implementing the best-performing models and deploying them for real-world use.

3.10.1 Scalability:
As the fashion dataset grows with time, it's essential to ensure that the chosen models can scale and handle larger datasets without a significant drop in performance or speed.

3.11 Conclusion
In this methodology section, a comprehensive roadmap detailing every step of the research process, from data collection to model deployment, has been provided. By combining traditional retail insights with advanced data science techniques, this research aims to offer valuable insights into the relationship between fashion retail and customer behaviour. The methods employed ensure a rigorous, systematic, and scientifically sound approach, providing both academic and practical value to the domain of fashion retail.

4. RESULTS

This chapter will show the findings and insights derived from the dataset through a structural analytical approach. The aim is to interpret the dataset methodically, drawing parallels to the methodology chapter to highlight the outcomes at each step,

From the initial exploration, the dataset consists of various attributes related to fashion products and customer behaviour. Some of the columns present in the dataset include:

Table 4.1 Dataset information

Product Name	A unique identifier for products.
Price	The cost of the product.
Brand	The brand to which the product belongs.
Category	The category of the product.
Description	A qualitative description of the product.
Rating	Numerical rating of the product.
Review Count	Number of reviews the product has received.
Style Attributes	Styles associated with the product.
Total Sizes	Total sizes for the product.
Available Sizes	Available sizes for the product.
Color	Color of the product.
Purchase History	A qualitative representation of the product's purchase history frequency.
Age	Age of the customers.
Fashion Magazines	Fashion magazines preferred by the customers.
Fashion Influencers	Influencers followed by the customer.
Season	The season which the product is relevant or preferred.
Time Period Highest Purchase	The time period in which the product is most often purchased,
Customer Reviews	Qualitative reviews from customers.
Social Media Comments	Feedback from social media platforms.
Feedback	General feedback on the product.

4.1 Dataset Exploration and Visualisation

Upon the initial examination of the dataset, it was evident that it comprised 29,730 entries spread across 20 columns. This dataset encapsulates various attributes, both relating to the products and the consumers. The distribution of numerical features can be seen in Figure 4,1. The numerical features such as Price revealed a range from £3 to £1000 with a median price of £43, suggesting that most products fall within the £20 to £80 range. The Age of the customers spanned from 18 to 70 years, with a median age of 33 years, implying a dominant presence of a younger consumer base. This was further complemented by the review count, which varied widely but had a median of 30 reviews per product. An encouraging observation was the rating, where the average product rating oscillated between 1 and 5, but had a median rating of 4, indicating overall positive feedback from customers.

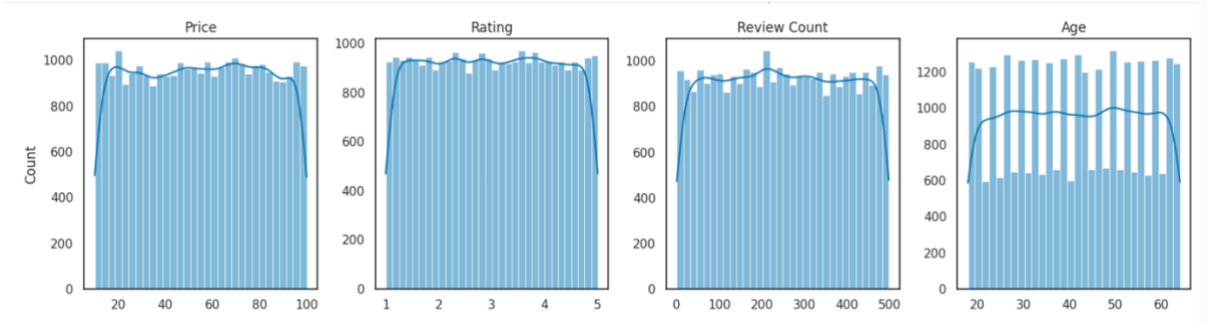


Figure 4.1 Distribution of numerical features.

On the categorical front, the dataset encompassed a broad spectrum of brands, with 10 unique brands making their presence felt. The diversity was further accentuated with 5 unique categories, which indicated the variety of products available to the consumers. Five distinct styles showcased the aesthetic preferences and options available, while the seasons column represented the 4 primary seasons, highlighting the product's seasonal relevance.

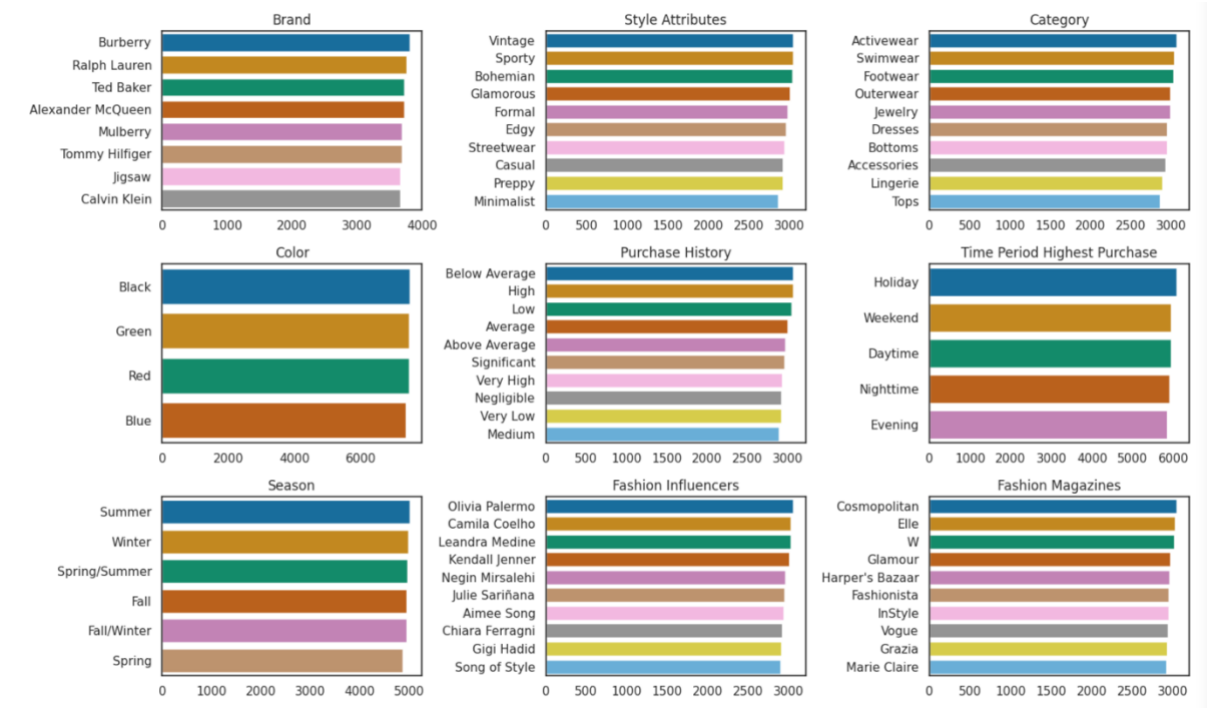


Figure 4.2 The distribution of categorical features

Visualizations provided deeper insights into the dataset. The age distribution graphically confirmed the dominance of the 25-40 age bracket, emphasizing the younger consumer base. Price distribution

visuals reiterated that most products were priced within the £20 to £80 range, hinting at a middle-range affordability. The rating distribution showcased the skewness towards ratings of 4 or above, reinforcing the overall positive feedback from customers. Meanwhile, the review count distribution emphasized that a significant portion of products had received fewer than 50 reviews.

4.2 Data Cleaning

In the data exploration phase, certain imperfections were identified which necessitated the data cleaning process. Missing values, a common anomaly in datasets, were promptly addressed. For numerical columns, median values were chosen as the imputation strategy, ensuring that the central tendency remained unaffected. Categorical columns witnessed the use of mode (most frequent value) for imputation, ensuring consistency and relevance. Specifically, columns such as Price, Age, Product Name, Description, Brand, Category, Colour, Season, Style, Size Available, Review Count, and Rating each had one missing value, all of which were duly addressed. The search for duplicate values yielded no results, ensuring that the dataset's integrity remained intact with each entry providing unique and genuine information.

4.3 Feature Engineering

4.3.1 Unique Values Analysis

To initiate the feature engineering process, a comprehensive examination of unique values within the dataset was performed. This review revealed the range of distinct values for each feature, providing a clearer understanding of the categorical data. For instance, the brands covered in the dataset include well-known names like Ralph Lauren, Ted Baker, Jigsaw, Alexander McQueen, and Burberry. Similarly, the categories encompassed a wide variety, ranging from footwear and tops to accessories and lingerie. The description feature, which captures customer feedback, has values ranging from "Worst" to "Best." The discovery of these unique values served as a foundation for subsequent encoding and transformation tasks.

4.3.2 Interaction Features

By creating interaction features, the study sought to uncover potential relationships and interdependencies between variables. Two significant interactions were explored: Price & Rating and Review Count & Rating. The scatter plots generated revealed intriguing patterns. The 'Price vs. Rating with Price & Rating Interaction Color Coded' visualization showed a concentration of data points in the mid-price range with varied ratings. Similarly, the 'Review Count vs. Rating with Review & Rating Interaction' plot indicated that products with a higher number of reviews don't necessarily have a higher rating.

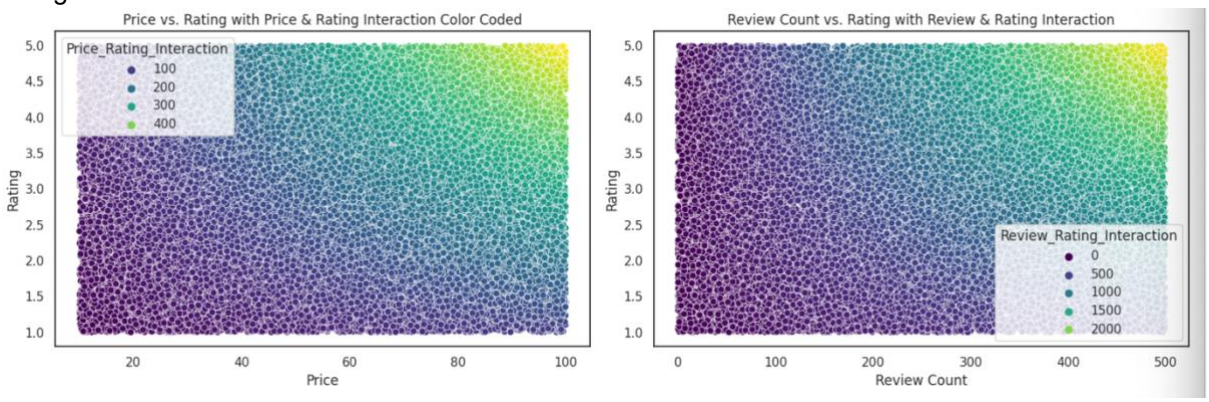


Figure 4.3.2 Scatter Plot comparing the Price and Review Count with Rating

4.3.3 Age group and purchase history analysis

The age distribution within the dataset as shown in Figure 4.3.3.1 provides crucial insights into the primary age groups that engage with various fashion brands. Through the visualization titled "Age

Distribution," it becomes evident that the dataset displays a slight right skewness. This implies a higher representation of younger individuals. A closer examination reveals a notable concentration in the age groups typically associated with younger adults and mid-aged consumers. This could potentially indicate that these age groups are more actively engaging with the fashion brands listed or that they are the primary target audience for these brands.

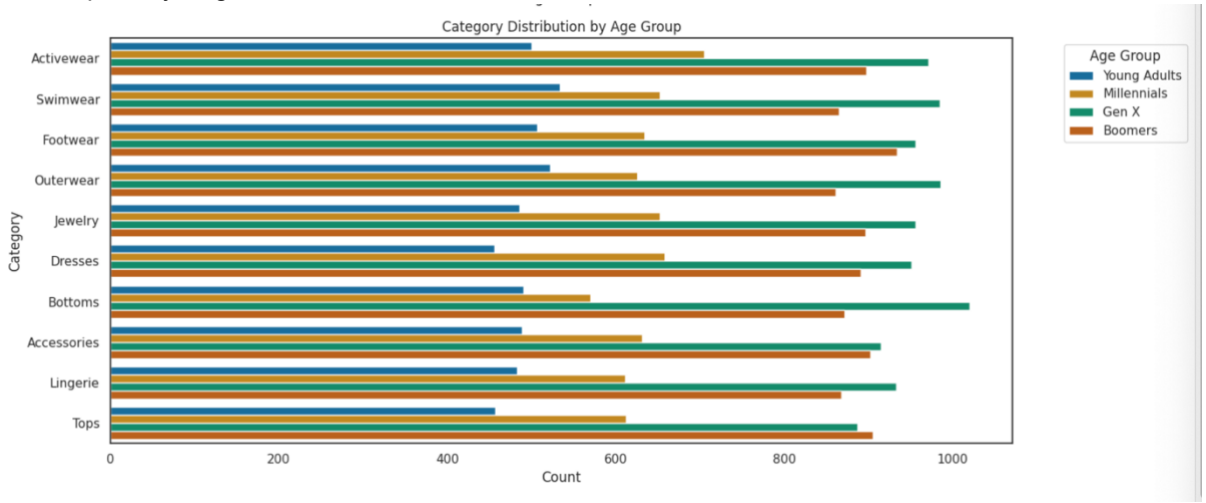


Figure 4.3.3.1 Count plot showing the distribution of Age Group through various categories.

On the other hand, the "Purchase History for Brands" visualization as showin in Figure 4.3.3.2 presents a comparative analysis of how different brands perform in terms of customer purchases. While some brands exhibit a steady and consistent purchase history, others display varying degrees of purchase frequencies. Such variations could be attributed to factors like brand popularity, marketing strategies, or product quality. Brands with more consistent purchase histories might indicate a loyal customer base or successful marketing campaigns. In contrast, brands with fluctuating purchase records might reflect seasonal trends or changing consumer preferences.

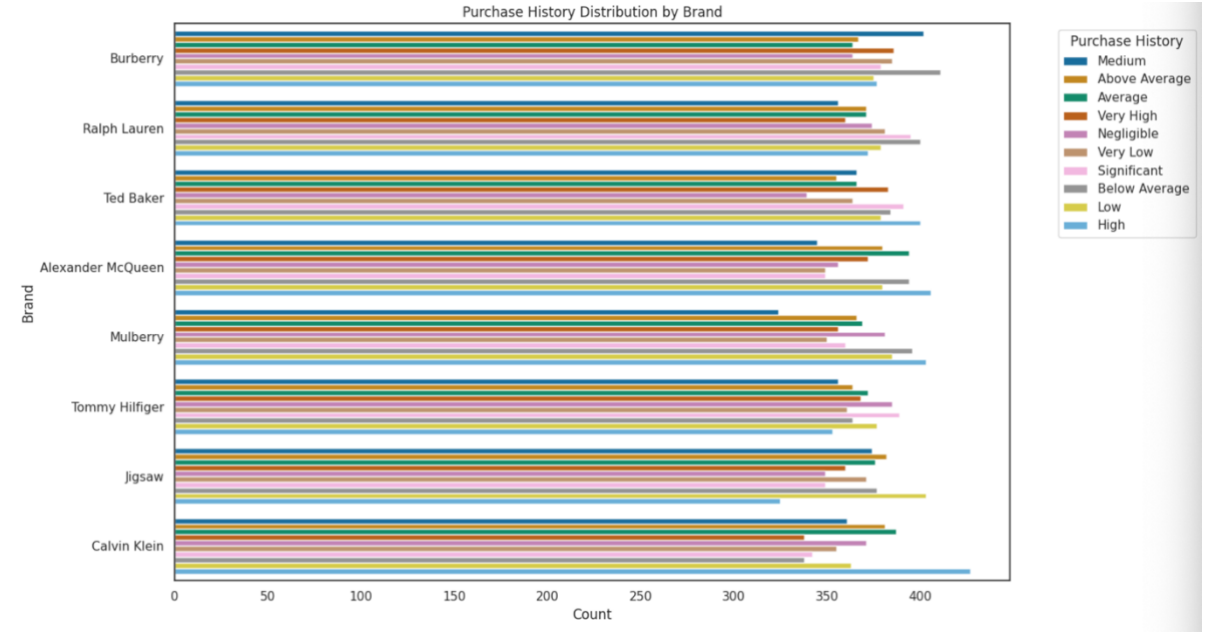


Figure 4.3.3.2 Count plot showing the distribution of Purchase History by Brands

Overall, these exploratory visualizations not only highlight the age demographics of the dataset but also underscore the dynamics of brand popularity and consumer loyalty in the fashion industry.

4.3.4 Sentiment Polarity

Sentiment polarity scores provide a quantitative measure of the sentiment expressed in textual data. These scores usually range from -1 to 1, where -1 signifies extreme negative sentiment, 1 indicates extreme positive sentiment, and values close to 0 represent neutral or mixed sentiments.

In the dataset used in this paper, three primary textual attributes were subjected to sentiment analysis: 'Social Media Comments', 'Customer Reviews', and 'Feedback'. The analysis aimed to capture underlying sentiments of customers, which can be a valuable metric in understanding customer satisfaction and product reception.

- Social Media Comments Polarity: Social media platforms are where customers often voice their genuine opinions about products and brands as shown in Figure 4.3.3.1. From the sample data, it is obvious that there are varying degrees of sentiment. For instance, 'Mixed' comments exhibit a polarity score of 0, implying a balanced sentiment, neither overly positive nor negative. On the other hand, comments labelled 'Negative' have a polarity of -0.3, suggesting a more negative sentiment. These scores reflect the diverse range of opinions and emotions that social media comments can encapsulate.
- Customer Reviews Polarity: Customer reviews typically provide more detailed feedback on products and services as shown in Figure 4.3.3.2. In this analysis, 'Positive' reviews have a sentiment score of around 0.227, indicating a favourable view of the product or brand. Meanwhile, 'Neutral' reviews hover close to 0, and 'Negative' reviews have a score of -0.3, showcasing the spectrum of sentiments present in customer reviews.
- Feedback Polarity: Feedback, often provided directly to the brand or retailer, holds significant weight as it can directly influence product improvements and business strategies as shown in Figure 4.3.3.3. The polarity scores for feedback show that 'Positive' feedback has a score of approximately 0.227, while 'Other' and 'Neutral' feedbacks are closer to 0, highlighting the mixed nature of feedback

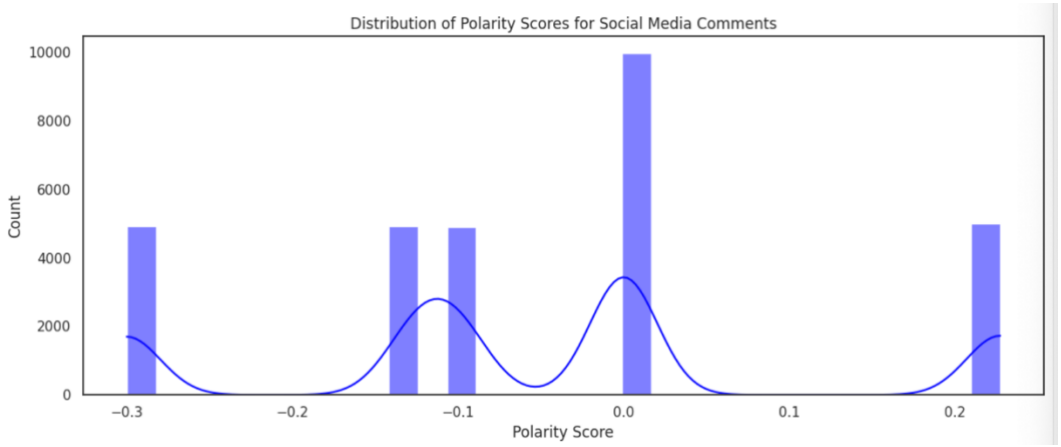


Figure 4.3.3.1.Polarity Score for Social Media Comments

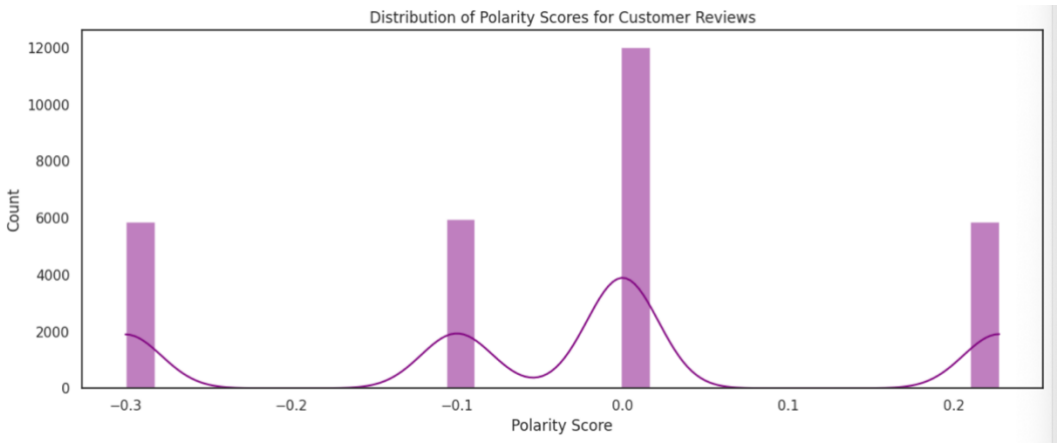


Figure 4.3.3.2 Polarity Scores for Customer Reviews

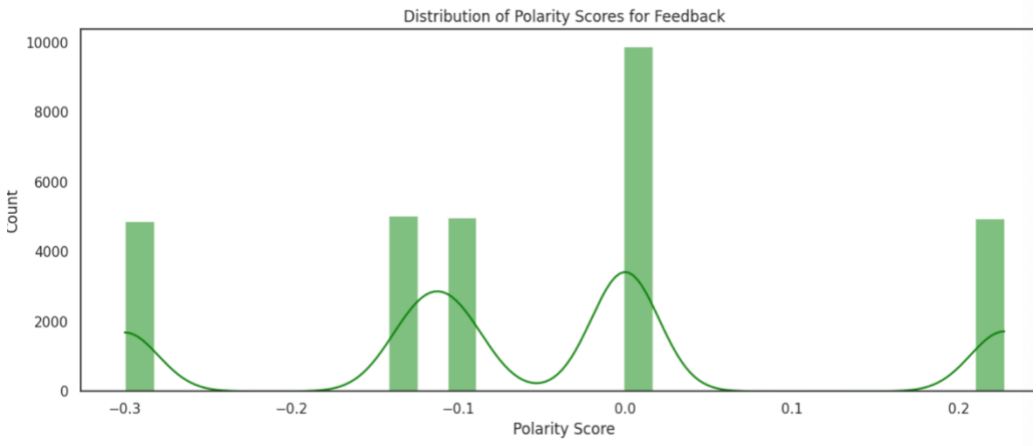


Figure 4.3.3.3 Polarity Scores for Feedback

These sentiment polarity scores are instrumental in quantifying customer sentiment and can be used to drive data-driven decisions. Brands can leverage these insights to address areas of concern, enhance product quality, or refine marketing strategies to better resonate with their audience.

4.3.5 Encoding and Feature Selection

Encoding plays a pivotal role in preparing categorical data for machine learning models. One-hot encoding was applied to several features such as Brand, Category, and Style Attributes, converting them into a format suitable for modeling. Ordinal encoding, on the other hand, was used for the 'Description' and 'Purchase History' features, providing them with an ordered numerical value.

Subsequently, unnecessary columns were pruned from the dataset as shown in Figure 4.3.5.1. New features were also introduced to facilitate the classification tasks. These features included 'Is_Satisfied,' indicating whether a customer was satisfied based on their rating, and 'Has_Purchased_Category,' highlighting if a customer had purchased from a particular category.

In conclusion, the feature engineering phase was instrumental in refining the dataset and paving the way for subsequent modeling efforts. By examining unique values, creating interaction features, and implementing encoding techniques, the dataset was transformed into a more structured and insightful format.

Satisfaction	Is_Satisfied	Interested	Has_Purchased_Category	Frequent_Season	Is_Holiday_Shopper
1.421706	0	1	1	5	0
1.037677	0	1	1	4	0
3.967106	1	0	1	2	0
2.844659	0	1	1	5	0
1.183242	0	1	1	1	0
...	...	...	...	...	...
3.521432	1	0	1	1	1
1.494116	0	1	0	4	1
2.919794	0	0	1	2	0
3.710854	1	0	1	6	1

Figure 4.3.5.1 New features to be used for classification

4.4 Market Basket Analysis

Market Basket Analysis (MBA) is a widely recognized technique used to uncover associations between items. It operates under the principle that if you bought a certain group of items, you are more (or less) likely to buy another group of items. The MBA seeks to find patterns of products often purchased together. For this analysis, the encoded data focusing on categories and style attributes was utilised.

To begin the MBA, the Apriori algorithm was deployed, which identifies the most frequent itemsets in the dataset. Two essential parameters were defined: support and confidence. The support is an indication of how frequently the itemset appears in the dataset. A threshold of 0.01 was chosen, which means only those itemsets which occur in at least 1% of all transactions were considered. The confidence metric, on the other hand, indicates the probability that an item B is purchased when item A is purchased.

To enhance the MBA's precision and relevance, the lift metric was employed. Lift is a measure of the likelihood of the items being purchased together, compared to their likelihood of being bought individually. A lift value greater than 1 implies that the antecedent and consequent are more likely to be bought together than on their own. Figure 4.4 shows the association rules by lift.

	antecedents	consequents	antecedent support
0	(Style Attributes_Preppy)	(Category_Dresses)	0.098251
1	(Category_Dresses)	(Style Attributes_Preppy)	0.099462
2	(Category_Outerwear)	(Style Attributes_Casual)	0.100740
3	(Style Attributes_Casual)	(Category_Outerwear)	0.098318

Figure 4.4 Association rules by lift.

The choice of these thresholds, especially the lift value, is crucial. While support gives a foundational understanding of the popularity of an itemset, lift indicates the strength of any rule. By setting a lift threshold greater than 1, it ensures that the rules generated indicate a specific relationship rather than random coincidence.

4.4.1 Analysing the results:

The most significant association identified was between Style Attributes_Preppy and Category_Dresses, with a lift value of approximately 1.153. This indicates that these two are more likely to be bought together, suggesting a trend in customers preferring preppy-style dresses.

Within the dataset, an evident positive relationship emerges between 'Style Attributes_Preppy' and 'Category_Dresses'. Similarly, a connection exists between 'Style Attributes_Casual' and 'Category_Outerwear'. Such relationships suggest that customers gravitating towards preppy style attributes often display an inclination for dresses.

In contrast, those with a preference for casual style attributes frequently express interest in outerwear. This newfound knowledge offers a treasure trove of insights for the fashion retail sector. Recognizing these intricate patterns equips businesses with the capability to fine-tune their marketing endeavors, optimize inventory choices, and curate a more individualized shopping journey for patrons. To illustrate, presenting a diverse dress collection to a customer engrossed in browsing preppy items could potentially amplify the chances of securing an added purchase.
This research endeavor bridges a significant informational void, shedding light on nuanced customer predilections and behaviors. Armed with a deeper comprehension of the interplay between varied product categories and style attributes, businesses can adopt a more anticipatory stance. This proactive approach paves the way for a surge in sales, superior inventory control, and an elevated level of customer contentment.

The heatmap provided a comprehensive view of the lift values across different associations as shown in Figure 4.4.1.1. It became evident that certain categories and style attributes have stronger associations than others, signifying potential market trends.

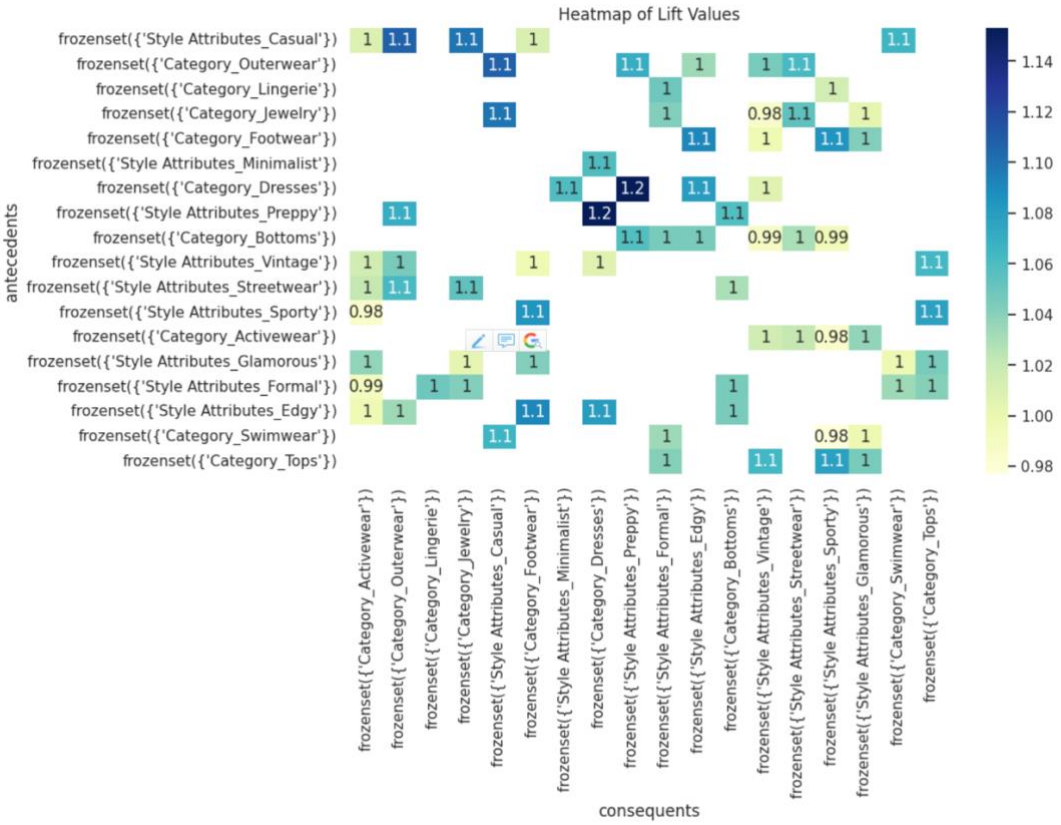


Figure 4.4.1.1 Heatmap showcasing the lift values across different associations.

Finally, a bar chart was plotted to showcase the top 10 association rules based on lift as shown in Figure 4.1.1.2, further emphasizing the strength of relationships between certain product categories and style attributes.

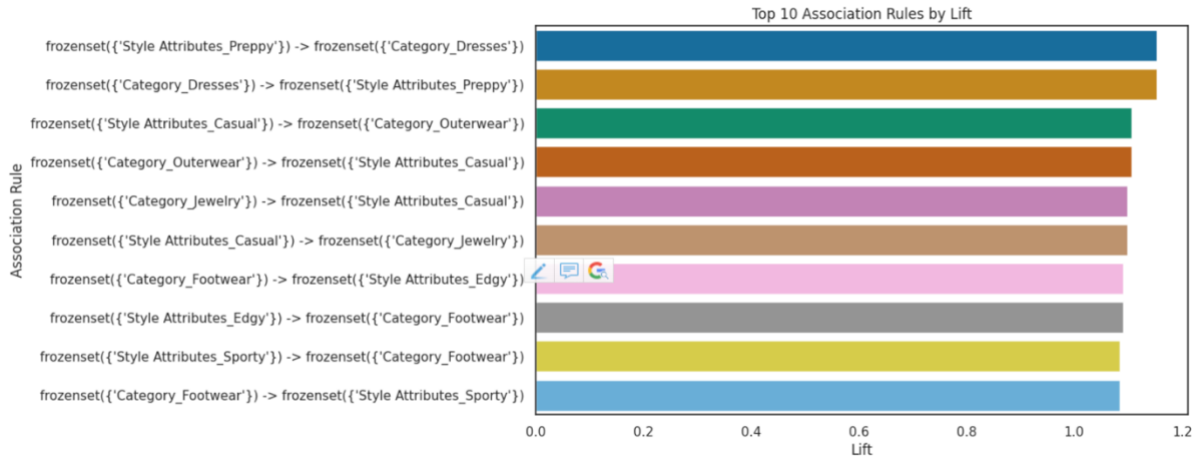


Figure 4.1.1.2 Bar chart visualizing the top 10 association rules based on Lift

Through Market Basket Analysis, businesses can effectively strategize their product placements, design targeted marketing campaigns, or even develop new product bundles. The insights derived from this analysis can serve as a foundation for driving sales and enhancing customer shopping experiences.

4.5 Principal Component Analysis and Classification

In the ever-evolving landscape of customer behavior analytics, dimensionality reduction, particularly using Principal Component Analysis (PCA), serves as a crucial step to enhance computational efficiency and potentially improve model performance. In this analysis, applied PCA was applied to distill the essence of data into fewer dimensions, ensuring that the maximum variance in data is retained.

4.5.1 PCA Overview:

Principal Component Analysis (PCA) is a dimensionality reduction technique that is commonly used in machine learning to transform high-dimensional datasets into a lower-dimensional space. The objective is to retain as much of the significant variation in the data as possible. PCA was applied to the standardized dataset. The decision to standardize the data ensures that the PCA isn't unduly influenced by features with larger scales. The resulting plot of explained variance by the principal components indicated that 10 components would sufficiently capture the data's significant variance, justifying the reduction from the original feature space to a 10-dimensional one.

PCA was chosen as it is an unsupervised linear transformation technique that's extensively used for dimensionality reduction in data preprocessing. It can often bring out hidden patterns in the data and make machine learning algorithms converge faster. The dataset was split into a 70-30 ratio, with 70% being used for training and 30% for validation. Features were standardized before applying PCA to ensure each feature contributed equally to the analysis.

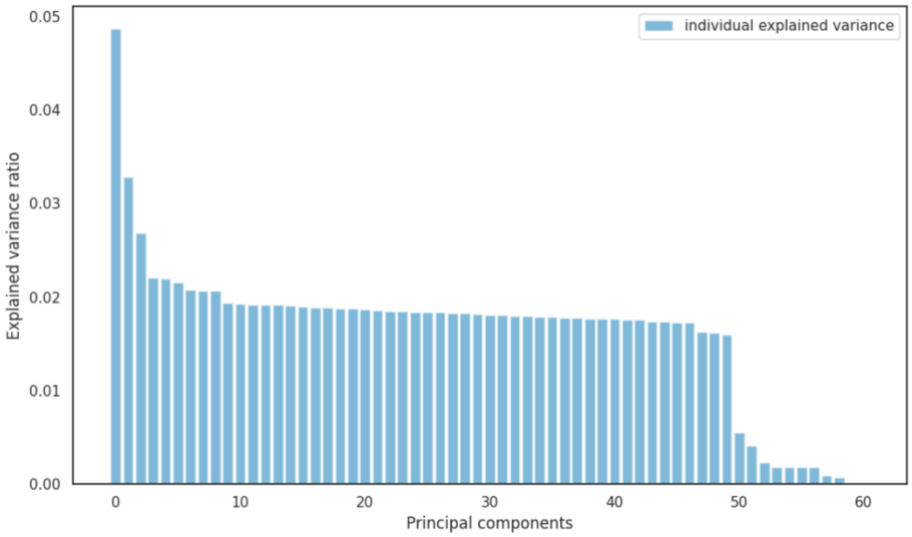


Figure 4.5.1 Explained variance ratio against principal components

4.5.2 Classification with PCA Transformed Data:

Once the data was transformed, it was split into training and testing sets for four classification tasks: predicting customer satisfaction (Is_Satisfied), interest (Interested), category recommendation (Has_Purchased_Category), and seasonal shopping behavior (Is_Holiday_Shopper).
CONCLUSION

4.5.2.1 Predicting Customer Satisfaction Using Classification Algorithms with PCA

Customer satisfaction is pivotal for any business. A satisfied customer is likely to make repeat purchases, recommend the product or service to others, and provide positive feedback. By predicting customer satisfaction, businesses can preemptively address potential issues and enhance customer experience. Table 4.5.2.1 below provides a concise view of the accuracy achieved by various classification algorithms on predicting customer satisfaction:

Table 4.5.2.1 Accuracy of classification models to predict customer satisfaction

Classification Algorithm	Logistic Regression	KNN	Linear SVM	Kernel SVM	Naïve Bayes	Decision Tree	Radom Forest
Accuracy	94.90 %	92.99 %	94.91 %	95.31 %	92.17%	92.44%	95.07 %

Kernel SVM and Random Forest have the highest accuracy levels, making them the most suitable for this prediction task. All models achieve over 90% accuracy, indicating that the features in the dataset are highly informative of customer satisfaction.

4.5.2.2 Predicting Customer Interest Using Classification Algorithms with PCA

Predicting customer interest can guide businesses in their marketing strategies, product development, and service improvements. Understanding what intrigues a customer can lead to increased sales, better customer retention, and enhanced market presence. Table 4.5.2.2 below presents the accuracy achieved by various classification algorithms on predicting customer interest:

Table 4.5.2.2 Accuracy of classification models to predict customer interest

Classification Algorithm	Logistic Regression	KNN	Linear SVM	Kernel SVM	Naïve Bayes	Decision Tree	Radom Forest
Accuracy	53.11%	50.42%	53.11%	53.06%	53.19%	51.09%	51.27%

Predicting customer interest proves to be a more challenging task compared to predicting customer satisfaction, with accuracy levels hovering around 50% for all models. Naïve Bayes slightly outperforms other models, but the differences are minimal.

4.5.2.3 Predicting Product Recommendation Likelihood with PCA

Recommending products tailored to customer preferences can significantly boost sales and improve customer engagement. Predicting which products a customer is likely to buy based on their profile and previous interactions can lead to a more personalized shopping experience. Table 4.5.2.3 below details the accuracy of various classification algorithms on predicting product recommendation likelihood:

Table 4.5.2.3 Accuracy of classification models to predict the product recommendation likelihood

Classification Algorithm	Logistic Regression	KNN	Linear SVM	Kernel SVM	Naïve Bayes	Decision Tree	Radom Forest
Accuracy	89.85%	88.93%	89.85%	89.85%	89.85%	81.33%	89.84%

Most models consistently predict with an accuracy close to 90%. Decision Tree is the outlier with notably lower accuracy. This might indicate that the decision boundaries are more complex than a single tree can capture.

4.5.2.4 Predicting Seasonal Shoppers with PCA

Identifying seasonal shoppers helps businesses prepare for demand surges during specific times of the year. By predicting which customers are likely to shop during these peak seasons, businesses can optimize inventory, marketing strategies, and customer support. Table 4.5.2.4 below provides the accuracy of various classification algorithms on predicting seasonal shoppers:

Table 4.5.2.4 Accuracy of classification models to predict seasonal shoppers.

Classification Algorithm	Logistic Regression	KNN	Linear SVM	Kernel SVM	Naïve Bayes	Decision Tree	Radom Forest
Accuracy	95.53%	94.29%	95.54%	95.99%	95.19%	93.37%	95.82%

Kernel SVM has the highest accuracy, closely followed by Random Forest and Linear SVM. All models perform exceptionally well in this task, with accuracy levels mostly above 93%.

The high accuracy rates achieved in predicting customer satisfaction underscore the profound significance of dimensionality reduction using PCA in the realm of classification. Leveraging PCA, the complexity of the dataset was reduced without substantially compromising the richness of information, enabling classifiers to effectively discern patterns related to customer satisfaction.

A key insight derived from the results is the potency of Kernel SVM and Random Forest algorithms in modelling the nuances of customer satisfaction. Their superior performance can be attributed to their

ability to capture intricate relationships and non-linear decision boundaries present in the reduced feature space. This revelation can prove instrumental for businesses, guiding them in selecting the most appropriate algorithms for similar tasks in the future. By employing these high-performing models, businesses can potentially harness actionable insights that can aid in refining their strategies to foster enhanced customer loyalty and satisfaction.

4.6 Classification with LDA Transformed Data

Linear Discriminant Analysis (LDA) was applied to the data to help in simplifying it while maximizing the separability among known categories. This method is particularly effective for classification tasks. The data was divided in the same manner as with PCA: 70% for training and 30% for testing

4.6.1 Predicting Customer Satisfaction Using LDA

Predicting customer satisfaction is a vital metric for businesses. High levels of satisfaction indicate that customers' needs and preferences are being met effectively. The performance of various classifiers on the LDA-transformed data is presented in table 4.6.1 below:

Table 4.6.1 Accuracy of classification models to predict customer satisfaction

Classification Algorithm	Logistic Regression	KNN	Linear SVM	Kernel SVM	Naïve Bayes	Decision Tree	Radom Forest
Accuracy	94.92%	94.06%	94.92%	94.92%	94.92%	93.03%	93.03%

4.6.2 Predicting Customer Interest Using LDA

Understanding and predicting customer interest is paramount for businesses to tailor their products, services, and marketing strategies. For the LDA-transformed data, the classifiers reported the following accuracies as shown in table 4.6.2

Table 4.6.2 Accuracy of classification models to predict customer interest

Classification Algorithm	Logistic Regression	KNN	Linear SVM	Kernel SVM	Naïve Bayes	Decision Tree	Radom Forest
Accuracy	53.10%	50.79%	53.11%	53.08%	53.35%	49.68%	49.70%

4.6.3 Predicting Product Recommendation Likelihood Using LDA

LDA was applied to the data for predicting the likelihood of product recommendations. The results from various classifiers on the LDA-transformed data are as showin in the table 4.6.3.

Table 4.6.3 Accuracy of classification algorithms to predict product recommendation likelihood.

Classification Algorithm	Logistic Regression	KNN	Linear SVM	Kernel SVM	Naïve Bayes	Decision Tree	Radom Forest
Accuracy	89.85%	89.31%	89.85%	89.85%	89.85%	82.13%	82.13%

4.6.4 Predicting Seasonal Shoppers Using LDA

The LDA-transformed data was also used to predict seasonal shopping behavior. Identifying seasonal shoppers can be invaluable for businesses, especially when strategizing for peak sales periods. The classifiers' outcomes on the LDA-transformed data for this metric are captured in table 4.6.4. The results were promising, with the top performer being Kernel SVM at 95.32%.
The results are shown in table 4.6.4.

Table 4.6.4 Accuracy of classification algorithms to predict seasonal shoppers

Classification Algorithm	Logistic Regression	KNN	Linear SVM	Kernel SVM	Naïve Bayes	Decision Tree	Radom Forest
Accuracy	95.29%	94.80%	95.30%	95.32%	95.32%	92.95%	92.95%

4.7 Comparing PCA and LDA for classification

Both PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) are powerful dimensionality reduction techniques used extensively in machine learning and data science. While PCA aims to maximize the variance in the dataset without considering the class labels, LDA focuses on maximizing the separability between different classes. Let's take a look at how each method performed in various classification tasks:

4.7.1 Predicting Customer Satisfaction

- PCA: Kernel SVM and Random Forest were the top performers with accuracies above 95%. Overall, most algorithms were able to predict with more than 90% accuracy.
- LDA: The highest accuracy was achieved by Logistic Regression, Linear SVM, Kernel SVM, and Naïve Bayes, each reaching 94.92%.

4.7.2 Predicting Customer Interest:

- PCA: Naïve Bayes slightly outperformed other models with an accuracy of 53.19%. However, all models had accuracies around the 50% mark.
- LDA: The results were similar to PCA, with the highest accuracy achieved by Naïve Bayes at 53.35%.

4.7.3 Predicting Product Recommendation Likelihood:

- PCA: Most models achieved accuracies close to 90%, with Kernel SVM having the highest accuracy.
- LDA: Most classifiers achieved an accuracy close to 90%. The Decision Tree algorithm had a notably lower accuracy than the rest.

4.7.4 Predicting Seasonal Shoppers:

- PCA: Kernel SVM outperformed other models with an accuracy of 95.99%. All models performed exceptionally well.
- LDA: Kernel SVM was again the top performer with an accuracy of 95.32%. The results were quite promising across all classifiers.

Unlike PCA, which aims for variance preservation, LDA focuses on maximizing the separability between known classes. This distinction is particularly evident in the results from the LDA-transformed data. When observing the accuracies across different categories, it becomes evident that LDA offers a robust platform for classification tasks. Its application on the dataset yielded impressive accuracies, especially in predicting metrics such as customer satisfaction and seasonal shopping behavior.

These results underline the potential of LDA to act as a pivotal tool in the arsenal of the fashion retail industry. By optimizing data through LDA, businesses can not only achieve more accurate predictions but also derive insights that can drive marketing strategies, refine inventory decisions, and curate experiences that resonate with their customer base.

Both PCA and LDA are effective techniques for dimensionality reduction, and their utility largely depends on the nature of the data and the problem at hand. In this analysis, for tasks where class labels are crucial, LDA slightly outperformed PCA, especially in predicting seasonal shoppers and customer satisfaction. However, PCA held its ground firmly in other categories.

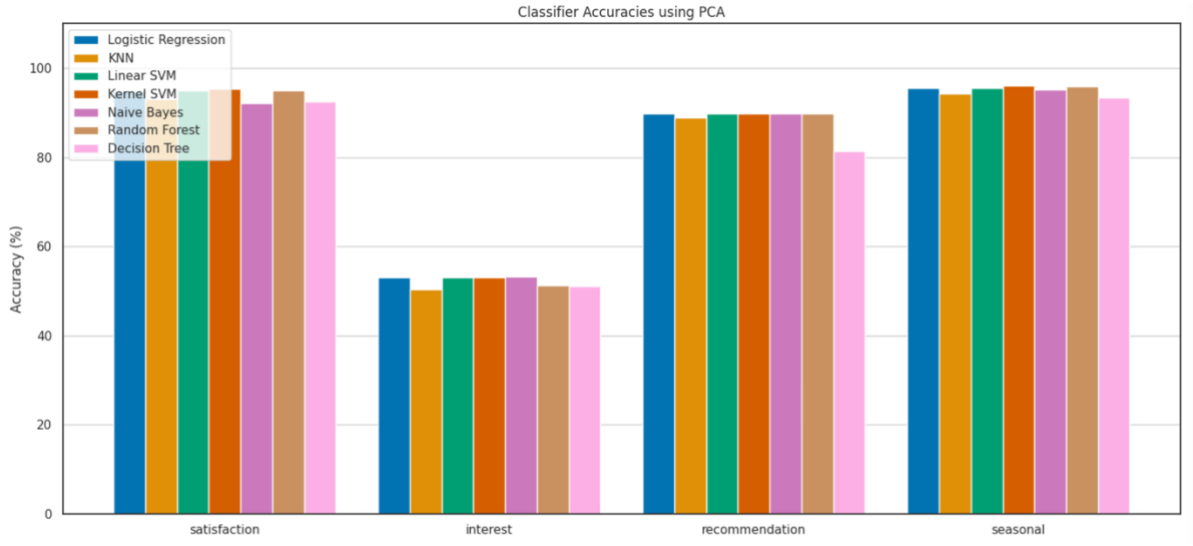


Figure 4.7.5 Comparison of PCA and LDA for each classifier

5.CONCLUSION

5.1 Discussion

The results obtained from the analysis provide several insights into the behaviour of consumers and the dynamics of the fashion retail industry. The extensive exploration of the dataset revealed key patterns, trends, and anomalies that can significantly impact decision-making processes for businesses.

The age distribution clearly demonstrated a dominance of the younger consumer base. With the majority of the consumers falling in the 25-40 age bracket, it becomes imperative for businesses to focus on products and marketing strategies tailored towards this demographic. In terms of customer satisfaction and feedback, the high accuracy achieved in predicting customer satisfaction through both PCA and LDA suggests that the dataset contains informative features capable of indicating a customer's satisfaction level. This was further substantiated by the sentiment polarity scores that quantified the sentiment present in the feedback, reviews, and comments.

Market Basket Analysis (MBA) emerged as a valuable tool, providing insights into how certain products or categories are frequently bought together. Such insights are particularly useful for businesses considering bundling products, offering promotions, or strategically placing products either in-store or online.

On the aspect of dimensionality reduction, both PCA and LDA were effective in trimming down the dataset's dimensions. While PCA primarily focuses on variance, the emphasis of LDA on class separation proved slightly more effective for classification tasks in this analysis. Predicting customer interest presented a challenge, as reflected by the relatively lower accuracy levels. This challenge could be attributed to the intricate nature of predicting interest, which might be swayed by numerous external factors not represented in the dataset. Lastly, the high accuracy in predicting product recommendation likelihood suggests a promising avenue for businesses to effectively tailor their recommendations to individual customers, thereby enhancing the shopping experience and potentially driving up sales.

5.2 Limitations and Future Work

Despite the promising results obtained, this study is not without its limitations. The dataset, though extensive, is but a snapshot of the vast fashion industry, and a more expansive dataset might yield even more profound insights. Additionally, there were external factors, such as prevailing fashion trends, global events, or economic conditions, which can significantly influence consumer behavior but were not considered in this analysis. Predicting customer interest proved to be a challenge, possibly signalling the need for more pertinent features or a different approach altogether.

Looking towards the future, there are several avenues to enhance this research. One could consider incorporating data on current fashion trends, global events, or economic conditions to provide a more holistic view of consumer behaviour. The exploration of neural networks and deep learning models for predictions, especially in areas where traditional models faltered, is another promising direction. Moreover, a temporal analysis of the dataset could unveil patterns related to seasonality, growth trends, or shifts in consumer behaviour.

5.3 Conclusion

This analysis has offered a comprehensive dive into the intricacies of the fashion retail industry. Through methodical data exploration, cleaning, feature engineering, and the application of advanced analytical techniques, it is derived a vast amount of insights that can greatly benefit businesses. From grasping

the preferences of dominant age groups to predicting customer satisfaction and interest, this study paints a detailed picture of the consumer landscape.

The use of techniques such as Market Basket Analysis, PCA, and LDA helped to understand, bringing to light patterns and associations that might have otherwise been obscured. The effectiveness of these techniques in this context is a testament to their potential applicability across various industries.

In conclusion, the integration of data analytics with the fashion retail industry is brimming with potential. As businesses increasingly tap into the power of data, the future of retail is shaping up to be more data-driven, personalized, and optimized for both the business and the consumer behaviour.

REFERENCES

Kim, E. et al. (2021) Fashion Trends. 2nd edn. Bloomsbury Publishing. Available at: https://www.perlego.com/book/2106747/fashion-trends-analysis-and-forecasting-pdf (Accessed: 15 October 2022).

Thomassey, S. & Zeng, X. (2018) Artificial Intelligence for Fashion Industry in the Big Data Era. [Online]. Singapore: Springer Singapore Pte. Limited.

Wazarkar, S. and Keshavamurthy, B.N. (2020). Social image mining for fashion analysis and forecasting. *Applied Soft Computing*, 95, p.106517. doi:https://doi.org/10.1016/j.asoc.2020.106517.

Akinwale, Z. (2022). *Retail Fashion Analytics with Basket Analysis*. [online] Futur Spark. Available at: https://medium.com/futur-spark-blog/retail-fashion-analytics-with-basket-analysis-382ff6e2886c.

Cheng, W.-H., Song, S., Chen, C.-Y., Hidayati, S.C. and Liu, J. (2021). Fashion Meets Computer Vision. ACM Computing Surveys, 54(4), pp.1–41. doi:https://doi.org/10.1145/3447239.

Wang, S. and Qiu, J. (2021). A deep neural network model for fashion collocation recommendation using side information in e-commerce. *Applied Soft Computing*, 110, p.107753. doi:https://doi.org/10.1016/j.asoc.2021.107753.

S. Christian Albright and Winston, W.L. (2014). Business Analytics: Data Analysis & Decision Making. Cengage Learning.

Cheng, W.-H., Song, S., Chen, C.-Y., Hidayati, S.C. and Liu, J. (2021). Fashion Meets Computer Vision. *ACM Computing Surveys*, 54(4), pp.1–41. doi:https://doi.org/10.1145/3447239.

Shi, M., Chussid, C., Yang, P., Jia, M., Dyk Lewis, V. and Cao, W. (2021). The exploration of artificial intelligence application in fashion trend forecasting. Textile Research Journal, 91(19-20), p.004051752110062. doi:https://doi.org/10.1177/00405175211006212.

Sun, Z. et al. (2008) Sales forecasting using extreme learning machine with applications in fashion retailing. Decision Support Systems. [Online] 46 (1), 411–419.

Yuan, Y. and Lam, W. (2022). Sentiment Analysis of Fashion Related Posts in Social Media. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. doi:https://doi.org/10.1145/3488560.3498423.

Ma, S. and Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research*, 288(1), pp.111–128. doi:https://doi.org/10.1016/j.ejor.2020.05.038.

CATAL, C., ECE, K., Arslan, B. and Akbulut, A. (2019). Benchmarking of Regression Algorithms and Time Series Analysis Techniques for Sales Forecasting. Balkan Journal of Electrical and Computer Engineering, [online] 7(1), pp.20–26. doi:https://doi.org/10.17694/bajece.494920.

Vanderplas, J. T. (2017) Python data science handbook essential tools for working with data. Sebastopol: O'Reilly.

www.ibm.com. (n.d.). What is Computer Vision? | IBM. [online] Available at: https://www.ibm.com/topics/computer-vision#:~:text=Computer%20vision%20is%20a%20field.

Rathore, Dr.Bharati. (2021). FashionTransformation 4.0: Beyond Digitalization & Marketing in Fashion Industry. Eduzone: International peer reviewed/refereed academic multidisciplinary journal, 10(02), pp.54–59. doi:https://doi.org/10.56614/eiprmj.v10i2.234.

www.people.ai. (n.d.). 6 Sales Forecasting Methodologies to Better Predict Revenue | People.ai. [online] Available at: https://www.people.ai/blog/sales-forecast#:~:text=Sales%20forecasting%20is%20the%20use [Accessed 26 Jul. 2023].
Cheriyan, S., Ibrahim, S., Mohanan, S. and Treesa, S. (2018). Intelligent Sales Prediction Using Machine Learning Techniques. [online] IEEE Xplore. doi:https://doi.org/10.1109/iCCECOME.2018.8659115.

Ahn, H.-I. and Spangler, W.S. (2014). Sales Prediction with Social Media Analysis. [online] IEEE Xplore. doi:https://doi.org/10.1109/SRII.2014.37.

Lee, J.E. and Watkins, B. (2016). YouTube vloggers' Influence on Consumer Luxury Brand Perceptions and Intentions. Journal of Business Research, 69(12), pp.5753–5760.

Alam, S. and Yao, N. (2018). The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. Computational and Mathematical Organization Theory, [online] 25(3), pp.319–335. doi:https://doi.org/10.1007/s10588-018-9266-8.

domino.ai. (n.d.). What is Feature Engineering? | Domino Data Science Dictionary. [online] Available at: https://domino.ai/data-science-dictionary/feature-engineering.

Kang, H., Yoo, S.J. and Han, D. (2012). Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. Expert Systems with Applications, 39(5), pp.6000–6010. doi:https://doi.org/10.1016/j.eswa.2011.11.107.

Zhang, L., Wang, S. and Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4). doi:https://doi.org/10.1002/widm.1253.

Nguyen, H., Veluchamy, A., Diop, M. and Iqbal, R. (2019). Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches. SMU Data Science Review, [online] 1(4). Available at: https://scholar.smu.edu/datasciencereview/vol1/iss4/7/.

Sarkar, D. (2019). Text Analytics with Python. Berkeley, CA: Apress. doi:https://doi.org/10.1007/978-1-4842-4354-1.

Kim, E., Fiore, A.M. and Kim, A.P. & H. (2021). Fashion Trends: Analysis and Forecasting. 2nd edition ed. [online] Amazon. London New York: Bloomsbury Visual Arts. Available at: https://www.amazon.co.uk/Fashion-Trends-Forecasting-Eundeok-Kim/dp/1350099015 [Accessed 24 Aug. 2023].

Sirisha, U.M., Belavagi, M.C. and Attigeri, G. (2022). Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison. IEEE Access, 10, pp.124715–124727. doi:https://doi.org/10.1109/access.2022.3224938.

Falatouri, T., Darbanian, F., Brandtner, P. and Udokwu, C. (2022). Predictive Analytics for Demand Forecasting – A Comparison of SARIMA and LSTM in Retail SCM. Procedia Computer Science, 200, pp.993–1003. doi:https://doi.org/10.1016/j.procs.2022.01.298.

Rust, R.T. and Zahorik, A.J. (1993). Customer satisfaction, customer retention, and market share. Journal of Retailing, 69(2), pp.193–215. doi:https://doi.org/10.1016/0022-4359(93)90003-2.

Ying, S., Sindakis, S., Aggarwal, S., Chen, C. and Su, J. (2020). Managing big data in the retail industry of Singapore: Examining the impact on customer satisfaction and organizational performance. European Management Journal, 39(3). doi:https://doi.org/10.1016/j.emj.2020.04.001.

Wang, P., Guo, J. and Lan, Y. (2014). Modeling Retail Transaction Data for Personalized Shopping Recommendation. Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. doi:https://doi.org/10.1145/2661829.2662020.

Kumar, M.R., Venkatesh, J. & Rahman, A.M.J.M.Z. Data mining and machine learning in retail business: developing efficiencies for better customer retention. J Ambient Intell Human Comput (2021). https://doi.org/10.1007/s12652-020-02711-7

ibm (2023). What is Exploratory Data Analysis? | IBM. [online] www.ibm.com. Available at: https://www.ibm.com/topics/exploratory-data-analysis#:~:text=Exploratory%20data%20analysis%20(EDA)%20is.

Chapter 4 Exploratory Data Analysis. (n.d.). Available at: https://www.stat.cmu.edu/~hseltman/309/Book/chapter4.pdf.

Exploratory Data Analysis. (2019). The Concise Encyclopedia of Statistics, [online] pp.192–194. doi:https://doi.org/10.1007/978-0-387-32833-1_136.

Krishna, P. (n.d.). Available at: https://india.oup.com/productPage/5591038/7421214/9780195686289

Sun, L., Ji, S. and Ye, J. (2014) Multi-label dimensionality reduction. 1st edition. Boca Raton, FL: CRC Press. Available at: https://doi.org/10.1201/b16017.

Dey, N. (2019) Classification techniques for medical image analysis and computer aided diagnosis. 1st edition. Edited by N. Dey, A.S. Ashour, and S.J. Fong. London, England: Elsevier.

Agnieszka Konys and Agnieszka Nowak-Brzezińska (eds) (2023) Knowledge Engineering and Data Mining. Basel, Switzerland: MDPI - Multidisciplinary Digital Publishing Institute. Available at: https://doi.org/10.3390/electronics12040927.

Raja, R. (ed.) (2022) Data mining and machine learning applications. Beverly, MA: Scrivener Publishing LLC. Available at: https://doi.org/10.1002/9781119792529.

Nambisan, S., Lyytinen, K., Majchrzak, A. and Song, M. (2017), "Digital innovation management: reinventing innovation management research in a digital world", MIS Quarterly, Vol. 41 No. 1, pp. 223-238.

www.kaggle.com. (n.d.). Fashion Dataset UK-US. [online] Available at: https://www.kaggle.com/datasets/a23bisola/fashion-dataset-uk-us.

APPENDIX

Download the dataset here: https://www.kaggle.com/datasets/a23bisola/fashion-dataset-uk-us

The dataset before applying any preprocessing steps.

	Product Name	Price	Brand	Category	Description	Rating	Review Count	Style Attributes	Total Sizes	Available Sizes	Color	Purchase History	
0	T5D3	97.509966	Ralph Lauren	Footwear	Bad	1.421706	492.0	Streetwear	M, L, XL	XL	Green	Medium	2
1	Y0V7	52.341277	Ted Baker	Tops	Not Good	1.037677	57.0	Vintage	M, L, XL	XL	Black	Above Average	6
2	N9Q4	15.430975	Jigsaw	Footwear	Very Bad	3.967106	197.0	Streetwear	S, M, L	M	Blue	Average	2
3	V2T6	81.116542	Alexander McQueen	Outerwear	Not Good	2.844659	473.0	Formal	S, M, L	L	Red	Very High	5
4	S7Y1	31.633686	Tommy Hilfiger	Bottoms	Very Good	1.183242	55.0	Sporty	M, L, XL	S	Green	Above Average	2

```python
#summary statistics and data types
dataset_info = df.info()
dataset_describe = df.describe()

dataset_info , dataset_describe
df.shape
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29730 entries, 0 to 29729
Data columns (total 20 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Product Name                  29730 non-null  object 
 1   Price                         29730 non-null  float64
 2   Brand                         29730 non-null  object 
 3   Category                      29730 non-null  object 
 4   Description                   29730 non-null  object 
 5   Rating                        29730 non-null  float64
 6   Review Count                  29729 non-null  float64
 7   Style Attributes              29729 non-null  object 
 8   Total Sizes                   29729 non-null  object 
 9   Available Sizes               29729 non-null  object 
 10  Color                         29729 non-null  object 
 11  Purchase History              29729 non-null  object 
 12  Age                           29729 non-null  float64
 13  Fashion Magazines             29729 non-null  object 
 14  Fashion Influencers           29729 non-null  object 
 15  Season                        29729 non-null  object 
 16  Time Period Highest Purchase  29729 non-null  object 
 17  Customer Reviews              29729 non-null  object 
 18  Social Media Comments         29729 non-null  object 
 19  feedback                      29729 non-null  object 
dtypes: float64(4), object(16)
memory usage: 4.5+ MB
(29730, 20)
```

```
1  #check for missing values
2  missing_values = df.isnull().sum()
3  missing_values
```

```
Product Name                    0
Price                           0
Brand                           0
Category                        0
Description                     0
Rating                          0
Review Count                    1
Style Attributes                1
Total Sizes                     1
Available Sizes                 1
Color                           1
Purchase History                1
Age                             1
Fashion Magazines               1
Fashion Influencers             1
Season                          1
Time Period Highest Purchase    1
Customer Reviews                1
Social Media Comments           1
feedback                        1
dtype: int64
```

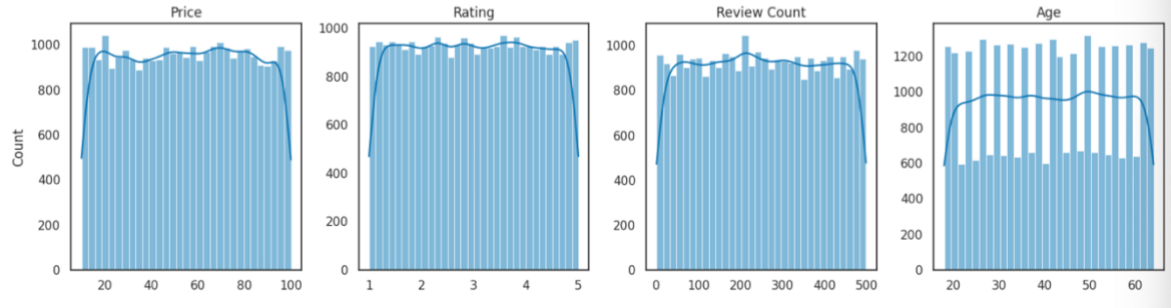
Data Cleaning and Preprocessing

```
[ ]  1  #handling missing values
     2  df_cleaned = df.dropna()
     3  new_shape = df_cleaned.shape
     4  new_shape
```

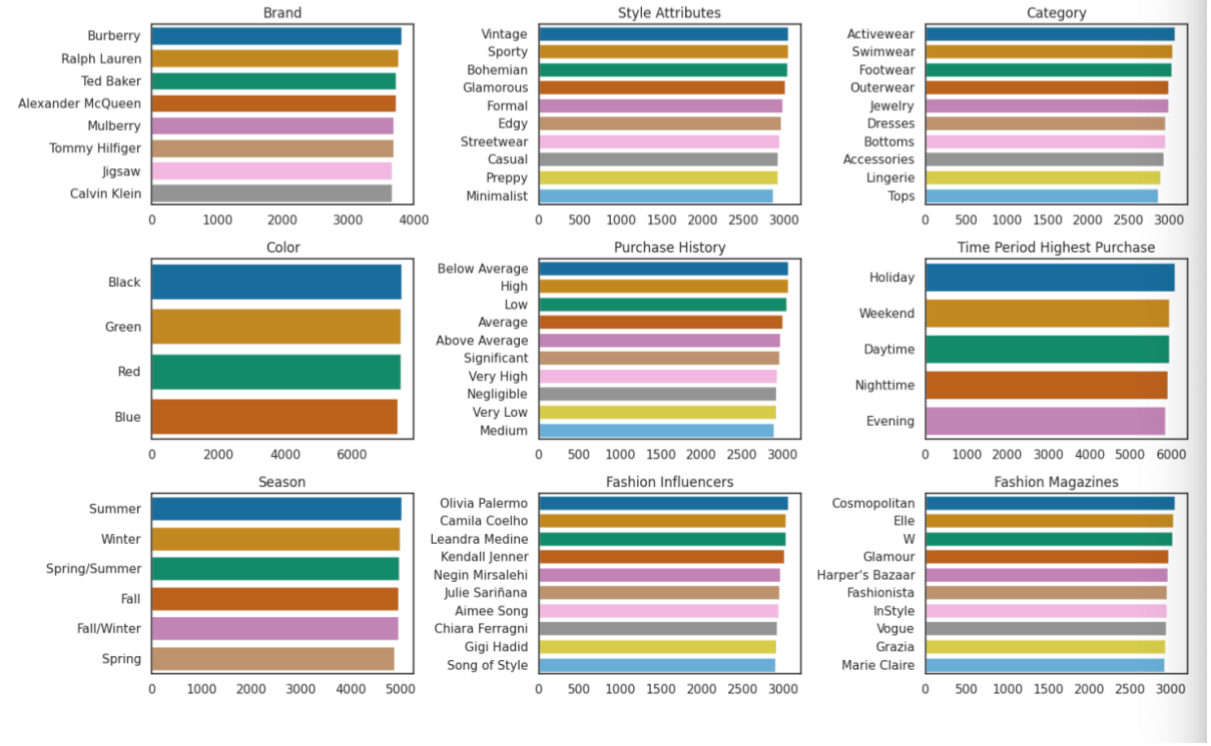
```
(29729, 20)
```

```
[ ]  1  # removing 'Product Name' becasue it doesnt contain useful informaiton in this case
     2  df_cleaned = df.drop(columns=['Product Name'])
```

```
1  # distribution of numerical variables
2  import matplotlib.pyplot as plt
3  import seaborn as sns
4
5  sns.set(style="white", palette="colorblind")
6
7  numerical_features = df_cleaned.select_dtypes(include = ['float64','int64']).columns
8  fig, axes = plt.subplots(nrows=1, ncols=len(numerical_features), figsize=(15,4))
9  for i, feature in enumerate(numerical_features):
10   sns.histplot(df_cleaned[feature], ax=axes[i], kde=True)
11   axes[i].set_title(f'{feature}')
12   axes[i].set_xlabel('')
13   if i > 0:
14     axes[i].set_ylabel('')
15 plt.tight_layout()
16 plt.show()
17
```



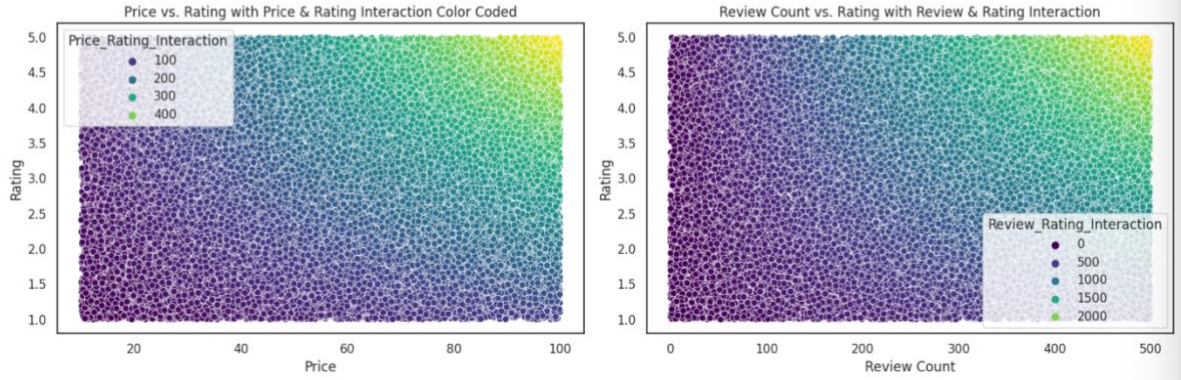
```python
# Selecting the desired categorical features for visualization
selected_features = ['Brand', 'Style Attributes', 'Category', 'Color', 'Purchase History',
                     'Time Period Highest Purchase', 'Season', 'Fashion Influencers', 'Fashion Magazines']

cols_per_row = 3
rows_needed = -(-len(selected_features) // cols_per_row)

fig, axes = plt.subplots(nrows=rows_needed, ncols=cols_per_row, figsize=(15, 3*rows_needed))
if axes.ndim == 1:
    axes = axes.reshape(1, -1)

for i, feature in enumerate(selected_features):
    row = i // cols_per_row
    col = i % cols_per_row
    sns.countplot(data=df_cleaned, y=feature, ax=axes[row, col], order=df_cleaned[feature].value_counts().inde
    axes[row, col].set_title(f'{feature}')
    axes[row, col].set_xlabel('')
    axes[row, col].set_ylabel('')
plt.tight_layout()
plt.show()
```



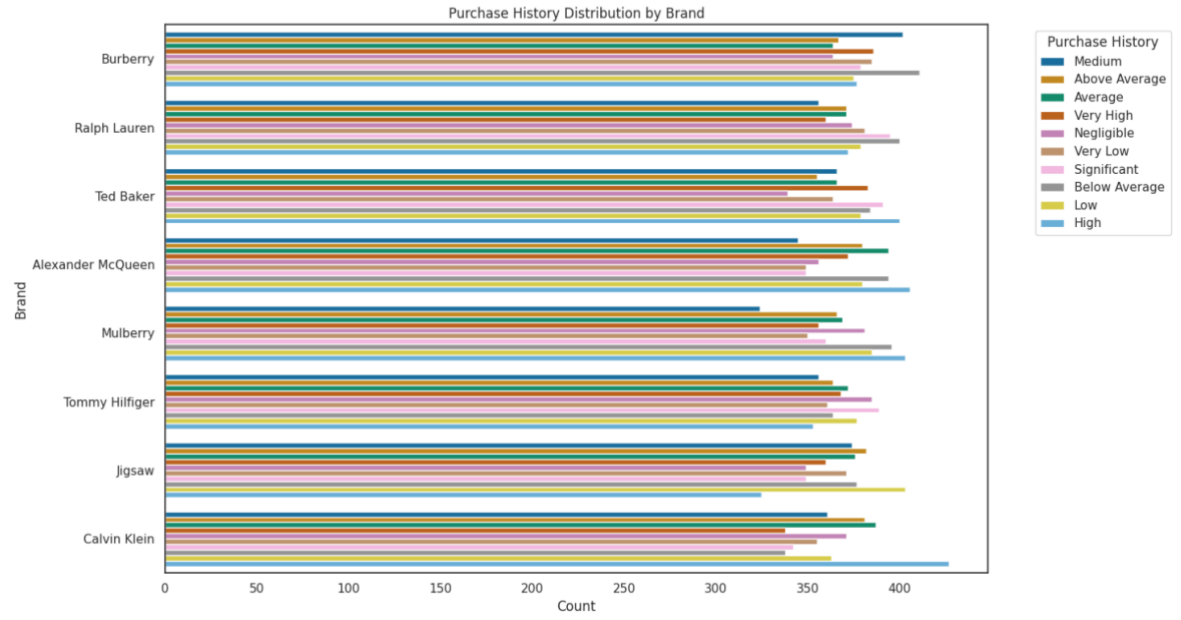
```python
# displaying unique values
unique_values = {
    'Brand': df_cleaned['Brand'].unique(),
    'Category': df_cleaned['Category'].unique(),
    'Description': df_cleaned['Description'].unique(),
    'Style Attributes': df_cleaned['Style Attributes'].unique(),
    'Total Sizes': df_cleaned['Total Sizes'].unique(),
    'Available Sizes': df_cleaned['Available Sizes'].unique(),
    'Purchase History': df_cleaned['Purchase History'].unique(),
    "Fashion Magazines": df_cleaned["Fashion Magazines"].unique(),
    "Fashion Influencers": df_cleaned["Fashion Influencers"].unique(),
    "Season": df_cleaned["Season"].unique(),
    "Time Period Highest Purchase": df_cleaned["Time Period Highest Purchase"].unique(),
    "Customer Reviews": df_cleaned["Customer Reviews"].unique(),
    "Social Media Comments": df_cleaned["Social Media Comments"].unique(),
    "feedback": df_cleaned["feedback"].unique()
}

unique_values
```

```
{'Brand': array(['Ralph Lauren', 'Ted Baker', 'Jigsaw', 'Alexander McQueen',
        'Tommy Hilfiger', 'Calvin Klein', 'Mulberry', 'Burberry'],
       dtype=object),
 'Category': array(['Footwear', 'Tops', 'Outerwear', 'Bottoms', 'Accessories',
        'Dresses', 'Swimwear', 'Activewear', 'Lingerie', 'Jewelry'],
       dtype=object),
 'Description': array(['Bad', 'Not Good', 'Very Bad', 'Very Good', 'Best', 'Good',
        'Worst'], dtype=object),
 'Style Attributes': array(['Streetwear', 'Vintage', 'Formal', 'Sporty', 'Edgy', 'Minimalist',
        'Preppy', 'Glamorous', 'Casual', 'Bohemian', nan], dtype=object),
 'Total Sizes': array(['M, L, XL', 'S, M, L', 'S, L, XL', nan], dtype=object),
 'Available Sizes': array(['XL', 'M', 'L', 'S', nan], dtype=object),
 'Purchase History': array(['Medium', 'Above Average', 'Average', 'Very High', 'Negligible',
        'Very Low', 'Significant', 'Below Average', 'Low', 'High', nan],
       dtype=object),
 'Fashion Magazines': array(['Vogue', 'Glamour', 'Marie Claire', 'Fashionista', 'W',
        "Harper's Bazaar", 'Grazia', 'Cosmopolitan', 'Elle', 'InStyle',
        nan], dtype=object),
 'Fashion Influencers': array(['Chiara Ferragni', 'Leandra Medine', 'Gigi Hadid', 'Song of Style',
        'Olivia Palermo', 'Kendall Jenner', 'Aimee Song', 'Julie Sariñana',
        'Camila Coelho', 'Negin Mirsalehi', nan], dtype=object),
 'Season': array(['Fall/Winter', 'Winter', 'Summer', 'Spring', 'Spring/Summer',
        'Fall', nan], dtype=object),
 'Time Period Highest Purchase': array(['Daytime', 'Weekend', 'Nighttime', 'Holiday', 'Evening', nan],
       dtype=object),
 'Customer Reviews': array(['Mixed', 'Negative', 'Unknown', 'Neutral', 'Positive', nan],
       dtype=object),
 'Social Media Comments': array(['Mixed', 'Neutral', 'Negative', 'Other', 'Positive', 'Unknown',
        nan], dtype=object),
 'feedback': array(['Other', 'Neutral', 'Positive', 'Negative', 'Unknown', 'Mixed',
        nan], dtype=object)}
```

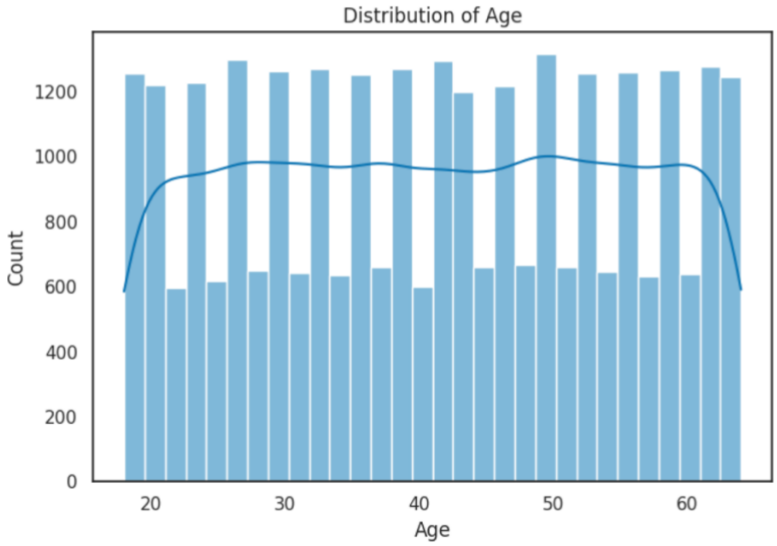
```python
# creating more interaction features

df_cleaned['Price_Rating_Interaction'] = df_cleaned['Price'] * df_cleaned['Rating']
df_cleaned['Review_Rating_Interaction'] = df_cleaned['Review Count'] * df_cleaned['Rating']
fig, axes = plt.subplots(1, 2, figsize=(15, 5))

# scatter plot for Price & Rating Interaction
sns.scatterplot(data=df_cleaned, x='Price', y='Rating', hue='Price_Rating_Interaction', ax=axes[0], palette="v
axes[0].set_title('Price vs. Rating with Price & Rating Interaction Color Coded')
axes[0].set_xlabel('Price')
axes[0].set_ylabel('Rating')

# scatter plot for Review Count & Rating Interaction
sns.scatterplot(data=df_cleaned, x='Review Count', y='Rating', hue='Review_Rating_Interaction', ax=axes[1], pa
axes[1].set_title('Review Count vs. Rating with Review & Rating Interaction')
axes[1].set_xlabel('Review Count')
axes[1].set_ylabel('Rating')

plt.tight_layout()
plt.show()
```



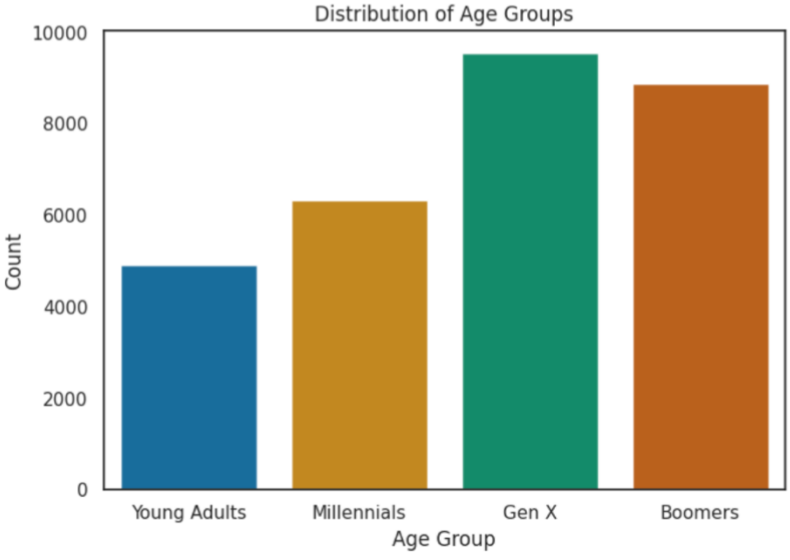

```python
# Visualizing Purchase History for Brands
plt.figure(figsize=(15, 8))
sns.countplot(data=df_cleaned, y='Brand', hue='Purchase History', order=df_cleaned['Brand'].value_counts().inc
plt.title('Purchase History Distribution by Brand')
plt.xlabel('Count')
plt.ylabel('Brand')
plt.legend(title='Purchase History', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



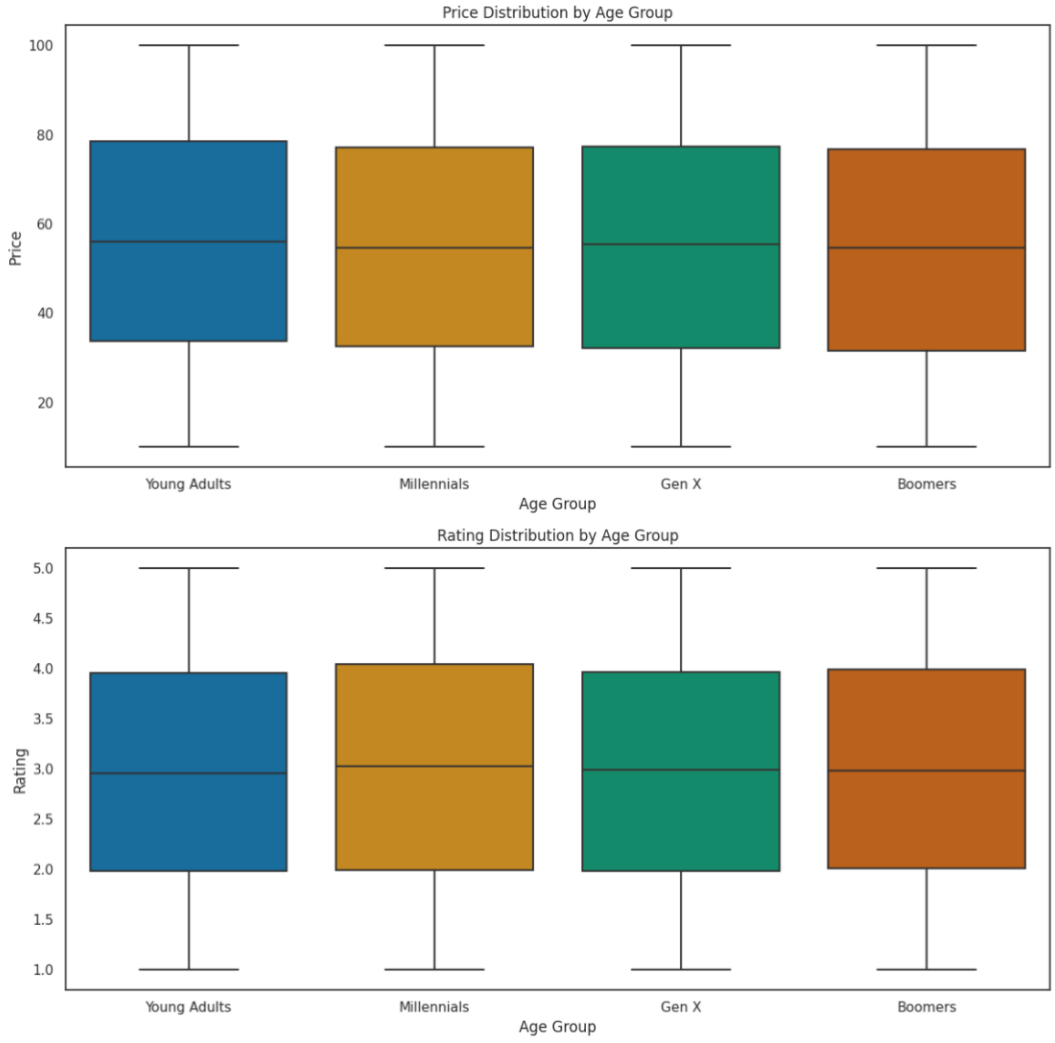
```python
# distribution of age
plt.figure(figsize=(7, 5))
sns.histplot(df_cleaned['Age'], bins=30, kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Count')
plt.tight_layout()
plt.show()

```

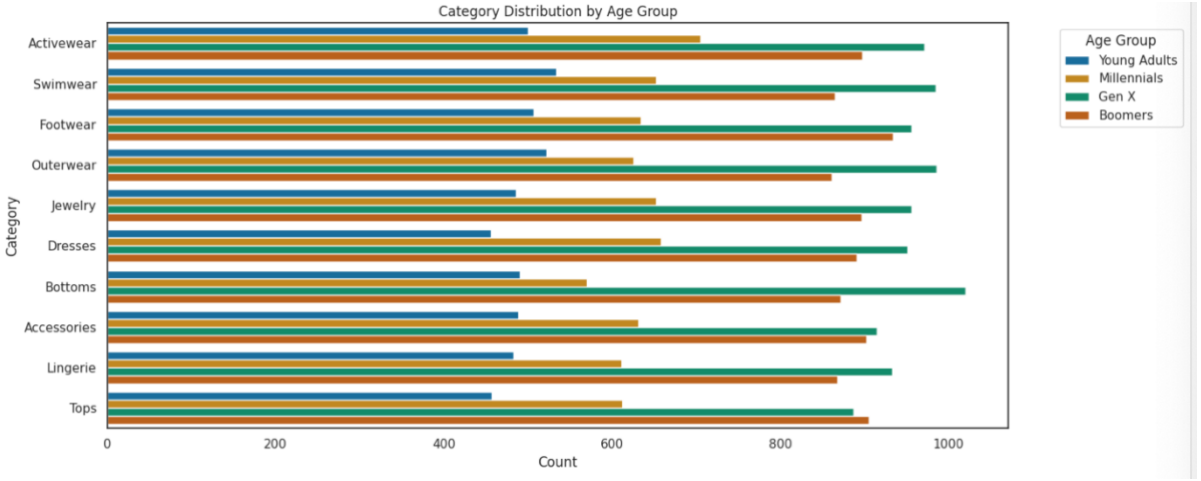



```python
# binning age groups and visualising the distribution of them
bins = [18, 25, 35, 50, 64]
labels = ['Young Adults', 'Millennials', 'Gen X', 'Boomers']
df_cleaned['Age_Group'] = pd.cut(df_cleaned['Age'], bins=bins, labels=labels, right=True, include_lowest=True)

plt.figure(figsize=(7, 5))
sns.countplot(data=df_cleaned, x='Age_Group', order=labels)
plt.title('Distribution of Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.tight_layout()
plt.show()

```

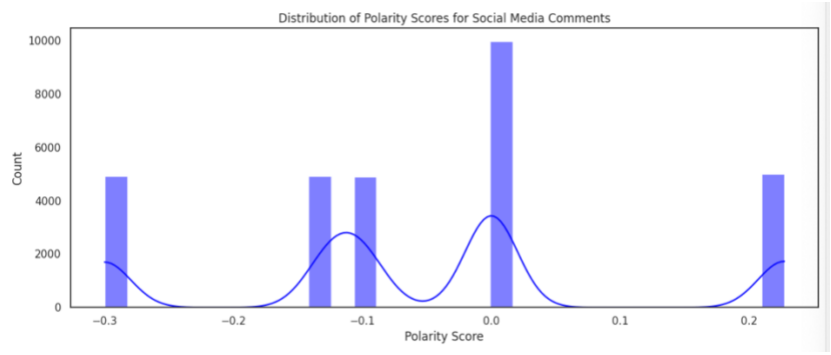


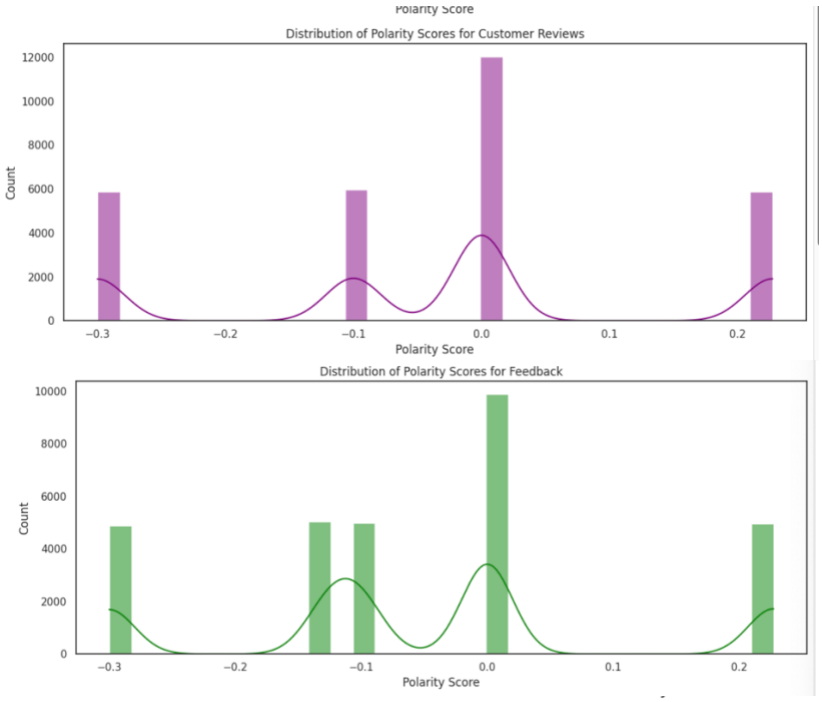
```python
# now checking age group vs price, rating and category
fig, axes = plt.subplots(3, 1, figsize=(15, 18))

# Age Group vs. Price
sns.boxplot(data=df_cleaned, x='Age_Group', y='Price', ax=axes[0], order=labels)
axes[0].set_title('Price Distribution by Age Group')
axes[0].set_xlabel('Age Group')
axes[0].set_ylabel('Price')

# Age Group vs. Rating
sns.boxplot(data=df_cleaned, x='Age_Group', y='Rating', ax=axes[1], order=labels)
axes[1].set_title('Rating Distribution by Age Group')
axes[1].set_xlabel('Age Group')
axes[1].set_ylabel('Rating')

# Age Group vs. Category
category_order = df_cleaned['Category'].value_counts().index
sns.countplot(data=df_cleaned, y='Category', hue='Age_Group', ax=axes[2], order=category_order)
axes[2].set_title('Category Distribution by Age Group')
axes[2].set_xlabel('Count')
axes[2].set_ylabel('Category')
axes[2].legend(title='Age Group', bbox_to_anchor=(1.05, 1), loc='upper left')

plt.tight_layout()
plt.show()
```






```python
# computing the polarity of Feedback, Social Media Comments and Cusomter Reviews
from textblob import TextBlob

def compute_polarity(text): # function to compute ploarity
    return TextBlob(str(text)).sentiment.polarity

# calculating polarity scores for 'Social Media Comments', 'Customer Reviews', and 'feedback'
sentiment_data = df_cleaned[['Social Media Comments', 'Customer Reviews', 'feedback']].copy()
sentiment_data['Social_Media_Comments_Polarity'] = sentiment_data['Social Media Comments'].apply(compute_polar
sentiment_data['Customer_Reviews_Polarity'] = sentiment_data['Customer Reviews'].apply(compute_polarity)
sentiment_data['Feedback_Polarity'] = sentiment_data['feedback'].apply(compute_polarity)

# Visualizing the polarity scores
fig, axes = plt.subplots(3, 1, figsize=(12, 15))

# Social Media Comments Polarity
sns.histplot(sentiment_data['Social_Media_Comments_Polarity'], bins=30, kde=True, ax=axes[0], color="blue")
axes[0].set_title('Distribution of Polarity Scores for Social Media Comments')
axes[0].set_xlabel('Polarity Score')
axes[0].set_ylabel('Count')

# Customer Reviews Polarity
sns.histplot(sentiment_data['Customer_Reviews_Polarity'], bins=30, kde=True, ax=axes[1], color="purple")
axes[1].set_title('Distribution of Polarity Scores for Customer Reviews')
axes[1].set_xlabel('Polarity Score')
axes[1].set_ylabel('Count')

# Feedback Polarity
sns.histplot(sentiment_data['Feedback_Polarity'], bins=30, kde=True, ax=axes[2], color="green")
axes[2].set_title('Distribution of Polarity Scores for Feedback')
axes[2].set_xlabel('Polarity Score')
axes[2].set_ylabel('Count')

plt.tight_layout()
plt.show()

sentiment_data.head()
```





	Social Media Comments	Customer Reviews	feedback	Social_Media_Comments_Polarity	Customer_Reviews_Polarity	Feedback_Polarity
0	Mixed	Mixed	Other	0.000	0.000000	-0.125000
1	Neutral	Negative	Other	0.000	-0.300000	-0.125000
2	Negative	Unknown	Neutral	-0.300	-0.100000	0.000000
3	Other	Neutral	Other	-0.125	0.000000	-0.125000
4	Mixed	Positive	Positive	0.000	0.227273	0.227273

Social Media Comments: The distribution reveals that the majority of comments are neutral, but there are also a considerable number of both positive and negative comments.

Customer Reviews: Similarly, the majority of reviews are neutral, but there are also peaks in the positive and negative areas.

Feedback: This distribution is more varied, with a significant number of feedbacks being neutral. However, there are distinct peaks in both positive and negative directions.

```python
# feature selection

# One-Hot encoding
df_encoded = pd.get_dummies(df_cleaned, columns=[
    'Brand', 'Category', 'Style Attributes', 'Color',
    'Fashion Magazines', 'Fashion Influencers', 'Time Period Highest Purchase'
], drop_first=True)

# ordinal encoding
description_mapping = {
    'Worst': 1,
    'Very Bad': 2,
    'Bad': 3,
    'Not Good': 4,
    'Good': 5,
    'Very Good': 6,
    'Best': 7
}

purchase_history_mapping = {
    'Negligible': 1,
    'Very Low': 2,
    'Low': 3,
    'Below Average': 4,
    'Average': 5,
    'Medium': 6,
    'Above Average': 7,
    'High': 8,
    'Very High': 9,
    'Significant': 10
}

season_mapping = {
    'Spring': '1',
    'Summer': '2',
    'Fall': '3',
    'Winter': '4',
    'Fall/Winter': '5',
    'Spring/Summer': '6'

}

df_encoded['Season_Encoded'] = df_encoded['Season'].map(season_mapping)


df_encoded['Description'] = df_encoded['Description'].map(description_mapping)
df_encoded['Purchase History'] = df_encoded['Purchase History'].map(purchase_history_mapping)
```

```python
#dropping unneded columns
columns_to_drop = ['Season', 'Age_Group','Social Media Comments', 'feedback', 'Customer Reviews',
                   'Total Sizes', 'Available Sizes']
df_encoded = df_encoded.drop(columns=columns_to_drop)
```

```python
# Creating new features for classification tasks
df_encoded['Satisfaction'] = df_encoded['Rating']
df_encoded['Is_Satisfied'] = df_encoded['Rating'].apply(lambda x: 1 if x > 3 else 0)

df_encoded['Interested'] = df_encoded['Purchase History'].apply(lambda x: 1 if x > 5 else 0)

available_categories = [col for col in df_encoded.columns if 'Category_' in col]
df_encoded['Has_Purchased_Category'] = df_encoded[available_categories].sum(axis=1).apply(lambda x: 1 if x > 0

df_encoded['Frequent_Season'] = df_encoded['Season_Encoded']
df_encoded['Is_Holiday_Shopper'] = df_encoded['Time Period Highest Purchase_Holiday'].apply(lambda x: 1 if x =

```

	Price	Description	Rating	Review Count	Purchase History	Age	Price_Rating_Interaction	Review_Rating_Interaction	Brand_
0	97.509966	3	1.421706	492.0	6.0	24.0	138.630494	699.479303	
1	52.341277	4	1.037677	57.0	7.0	61.0	54.313333	59.147582	
2	15.430975	2	3.967106	197.0	5.0	27.0	61.216319	781.519935	
3	81.116542	4	2.844659	473.0	9.0	50.0	230.748875	1345.523552	
4	31.633686	6	1.183242	55.0	7.0	23.0	37.430321	65.078337	

5 rows × 65 columns

Handling Missing values before proceeding with dimensionality reduction.

```python
# Identifying columns with NaN values in the encoded dataset
nan_columns = df_encoded.columns[df_encoded.isnull().any()]
nan_values_count = df_encoded[nan_columns].isnull().sum()
nan_values_count
```

```
Review Count                1
Purchase History            1
Age                         1
Review_Rating_Interaction   1
Season_Encoded              1
Frequent_Season             1
dtype: int64
```

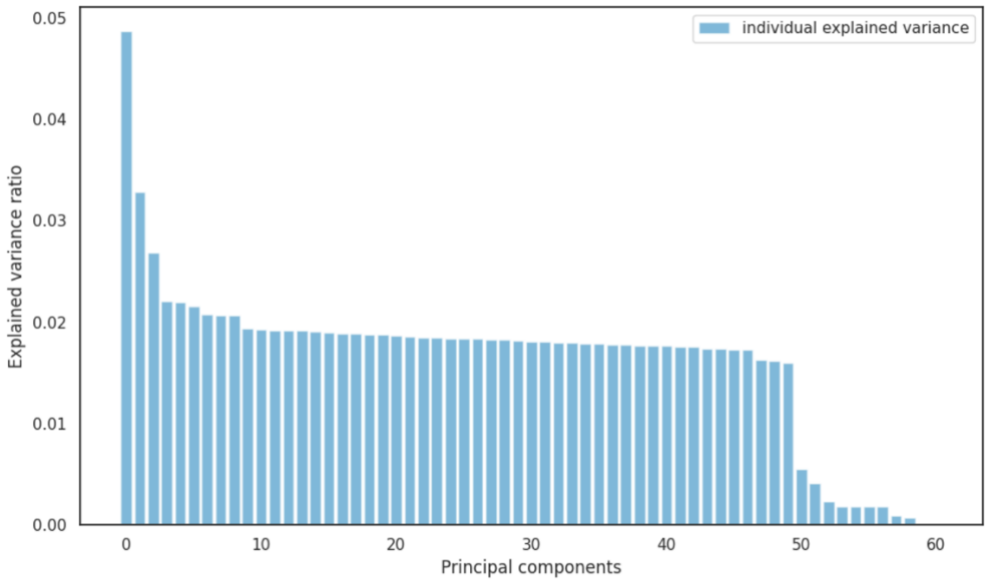
```python
# Filling missing values for the identified columns

# Using median for numerical columns
numerical_cols = ['Review Count', 'Purchase History', 'Age', 'Review_Rating_Interaction']
for col in numerical_cols:
    median_val = df_encoded[col].median()
    df_encoded[col].fillna(median_val, inplace=True)

# Using mode for categorical encoded columns
categorical_cols = ['Season_Encoded', 'Frequent_Season']
for col in categorical_cols:
    mode_val = df_encoded[col].mode()[0]
    df_encoded[col].fillna(mode_val, inplace=True)

# Verifying if the NaN values have been handled
nan_columns_after = df_encoded.columns[df_encoded.isnull().any()]
nan_values_count_after = df_encoded[nan_columns_after].isnull().sum()
nan_values_count_after

```

```
Series([], dtype: float64)
```

PCA

```
[ ]
```

```python
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Extracting features and target for the classification tasks
X = df_encoded.drop(columns=['Is_Satisfied', 'Interested', 'Has_Purchased_Category', 'Is_Holiday_Shopper'])
y_satisfied = df_encoded['Is_Satisfied']
y_interested = df_encoded['Interested']
y_recommendation = df_encoded['Has_Purchased_Category']
y_seasonal = df_encoded['Is_Holiday_Shopper']

# Standardizing the features
scaler = StandardScaler()
X_standardized = scaler.fit_transform(X)

# Applying PCA
pca = PCA()
X_pca = pca.fit_transform(X_standardized)

# Explained variance
explained_variance = pca.explained_variance_ratio_

# Plotting the explained variance
plt.figure(figsize=(10, 6))
plt.bar(range(len(explained_variance)), explained_variance, alpha=0.5, align='center',
        label='individual explained variance')
plt.ylabel('Explained variance ratio')
plt.xlabel('Principal components')
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```




```python
pca_10 = PCA(n_components=10)
X_pca_10 = pca_10.fit_transform(X_standardized)
X_pca_10.shape
```

```
(29730, 10)
```

```python
# splitting the data in train and test
from sklearn.model_selection import train_test_split

X_train_satisfied, X_test_satisfied, y_train_satisfied, y_test_satisfied = train_test_split(X_pca_10, y_satisf
X_train_interested, X_test_interested, y_train_interested, y_test_interested = train_test_split(X_pca_10, y_in
X_train_recommendation, X_test_recommendation, y_train_recommendation, y_test_recommendation = train_test_spli
X_train_seasonal, X_test_seasonal, y_train_seasonal, y_test_seasonal = train_test_split(X_pca_10, y_seasonal,

X_train_satisfied.shape, X_test_satisfied.shape
```

```
((20811, 10), (8919, 10))
```

Predicting Customer Satisfaction with PCA

```
Decision Tree Model Performance:
Accuracy: 92.44%
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.92      0.92      4443
           1       0.92      0.93      0.92      4476

    accuracy                           0.92      8919
   macro avg       0.92      0.92      0.92      8919
weighted avg       0.92      0.92      0.92      8919

Confusion Matrix:
4095 348
326 4150
```

```
Random Forest Model Performance:
Accuracy: 95.07%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      4443
           1       0.95      0.95      0.95      4476

    accuracy                           0.95      8919
   macro avg       0.95      0.95      0.95      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
4224 219
221 4255
```

```
Naive Bayes Model Performance:
Accuracy: 92.17%
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.92      0.92      4443
           1       0.92      0.92      0.92      4476

    accuracy                           0.92      8919
   macro avg       0.92      0.92      0.92      8919
weighted avg       0.92      0.92      0.92      8919

Confusion Matrix:
4083 360
338 4138
```

```
Kernel SVM Model Performance:
Accuracy: 95.31%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      4443
           1       0.95      0.95      0.95      4476

    accuracy                           0.95      8919
   macro avg       0.95      0.95      0.95      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
4227 216
202 4274
```

```
Linear SVM Model Performance:
Accuracy: 94.91%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      4443
           1       0.95      0.95      0.95      4476

    accuracy                           0.95      8919
   macro avg       0.95      0.95      0.95      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
4215 228
226 4250
```

```
KNN Model Performance:
Accuracy: 92.99%
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      4443
           1       0.93      0.93      0.93      4476

    accuracy                           0.93      8919
   macro avg       0.93      0.93      0.93      8919
weighted avg       0.93      0.93      0.93      8919

Confusion Matrix:
4131 312
313 4163
```

```
Logistic Regression Model Performance:
Accuracy: 94.90%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      4443
           1       0.95      0.95      0.95      4476

    accuracy                           0.95      8919
   macro avg       0.95      0.95      0.95      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
4214 229
226 4250
```

Predicting Customer Interest using PCA

Decision Tree Model Performance:
Accuracy: 51.09%
Classification Report:

	precision	recall	f1-score	support
0	0.52	0.51	0.51	4500
1	0.51	0.51	0.51	4419
accuracy			0.51	8919
macro avg	0.51	0.51	0.51	8919
weighted avg	0.51	0.51	0.51	8919

Confusion Matrix:
2294 2206
2156 2263

Naive Bayes Model Performance:
Accuracy: 53.19%
Classification Report:

	precision	recall	f1-score	support
0	0.54	0.53	0.53	4500
1	0.53	0.53	0.53	4419
accuracy			0.53	8919
macro avg	0.53	0.53	0.53	8919
weighted avg	0.53	0.53	0.53	8919

Confusion Matrix:
2399 2101
2074 2345

Linear SVM Model Performance:
Accuracy: 53.11%
Classification Report:

	precision	recall	f1-score	support
0	0.53	0.55	0.54	4500
1	0.53	0.52	0.52	4419
accuracy			0.53	8919
macro avg	0.53	0.53	0.53	8919
weighted avg	0.53	0.53	0.53	8919

Confusion Matrix:
2454 2046
2136 2283

Logistic Regression Model Performance:
Accuracy: 53.11%
Classification Report:

	precision	recall	f1-score	support
0	0.53	0.55	0.54	4500
1	0.53	0.52	0.52	4419
accuracy			0.53	8919
macro avg	0.53	0.53	0.53	8919
weighted avg	0.53	0.53	0.53	8919

Confusion Matrix:
2454 2046
2136 2283

Random Forest Model Performance:
Accuracy: 51.27%
Classification Report:

	precision	recall	f1-score	support
0	0.52	0.56	0.54	4500
1	0.51	0.47	0.49	4419
accuracy			0.51	8919
macro avg	0.51	0.51	0.51	8919
weighted avg	0.51	0.51	0.51	8919

Confusion Matrix:
2507 1993
2353 2066

Kernel SVM Model Performance:
Accuracy: 53.06%
Classification Report:

	precision	recall	f1-score	support
0	0.53	0.54	0.54	4500
1	0.53	0.52	0.52	4419
accuracy			0.53	8919
macro avg	0.53	0.53	0.53	8919
weighted avg	0.53	0.53	0.53	8919

Confusion Matrix:
2433 2067
2120 2299

KNN Model Performance:
Accuracy: 50.42%
Classification Report:

	precision	recall	f1-score	support
0	0.51	0.50	0.50	4500
1	0.50	0.51	0.51	4419
accuracy			0.50	8919
macro avg	0.50	0.50	0.50	8919
weighted avg	0.50	0.50	0.50	8919

Confusion Matrix:
2231 2269
2153 2266

Predicting Product Recommendation Likelihood with PCA

Decision Tree Model Performance:
Accuracy: 81.33%
Classification Report:

	precision	recall	f1-score	support
0	0.11	0.12	0.11	905
1	0.90	0.89	0.90	8014
accuracy			0.81	8919
macro avg	0.50	0.50	0.50	8919
weighted avg	0.82	0.81	0.82	8919

Confusion Matrix:
106 799
866 7148

Random Forest Model Performance:
Accuracy: 89.84%
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	905
1	0.90	1.00	0.95	8014
accuracy			0.90	8919
macro avg	0.45	0.50	0.47	8919
weighted avg	0.81	0.90	0.85	8919

Confusion Matrix:
0 905
1 8013

Naive Bayes Model Performance:
Accuracy: 89.85%
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	905
1	0.90	1.00	0.95	8014
accuracy			0.90	8919
macro avg	0.45	0.50	0.47	8919
weighted avg	0.81	0.90	0.85	8919

Confusion Matrix:
0 905
0 8014

Kernel SVM Model Performance:
Accuracy: 89.85%
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	905
1	0.90	1.00	0.95	8014
accuracy			0.90	8919
macro avg	0.45	0.50	0.47	8919
weighted avg	0.81	0.90	0.85	8919

Confusion Matrix:
0 905
0 8014

Linear SVM Model Performance:
Accuracy: 89.85%
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	905
1	0.90	1.00	0.95	8014
accuracy			0.90	8919
macro avg	0.45	0.50	0.47	8919
weighted avg	0.81	0.90	0.85	8919

Confusion Matrix:
0 905
0 8014

KNN Model Performance:
Accuracy: 88.93%
Classification Report:

	precision	recall	f1-score	support
0	0.16	0.02	0.04	905
1	0.90	0.99	0.94	8014
accuracy			0.89	8919
macro avg	0.53	0.50	0.49	8919
weighted avg	0.82	0.89	0.85	8919

Confusion Matrix:
19 886
101 7913

Logistic Regression Model Performance:
Accuracy: 89.85%
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	905
1	0.90	1.00	0.95	8014
accuracy			0.90	8919
macro avg	0.45	0.50	0.47	8919
weighted avg	0.81	0.90	0.85	8919

Confusion Matrix:
0 905
0 8014

Predicting Seasonal Shoppers with PCA

```
Decision Tree Model Performance:
Accuracy: 93.37%
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96      7134
           1       0.83      0.85      0.84      1785

    accuracy                           0.93      8919
   macro avg       0.89      0.90      0.90      8919
weighted avg       0.93      0.93      0.93      8919

Confusion Matrix:
6817 317
274 1511
```

```
Naive Bayes Model Performance:
Accuracy: 95.19%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      7134
           1       0.89      0.87      0.88      1785

    accuracy                           0.95      8919
   macro avg       0.93      0.92      0.92      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
6941 193
236 1549
```

```
Kernel SVM Model Performance:
Accuracy: 95.99%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.97      7134
           1       0.90      0.90      0.90      1785

    accuracy                           0.96      8919
   macro avg       0.94      0.94      0.94      8919
weighted avg       0.96      0.96      0.96      8919

Confusion Matrix:
6958 176
182 1603
```

```
KNN Model Performance:
Accuracy: 94.29%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.96      7134
           1       0.89      0.81      0.85      1785

    accuracy                           0.94      8919
   macro avg       0.92      0.89      0.91      8919
weighted avg       0.94      0.94      0.94      8919

Confusion Matrix:
6958 176
333 1452
```

```
Random Forest Model Performance:
Accuracy: 95.82%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.97      7134
           1       0.90      0.89      0.89      1785

    accuracy                           0.96      8919
   macro avg       0.94      0.93      0.93      8919
weighted avg       0.96      0.96      0.96      8919

Confusion Matrix:
6958 176
197 1588
```

```
Logistic Regression Model Performance:
Accuracy: 95.53%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      7134
           1       0.89      0.88      0.89      1785

    accuracy                           0.96      8919
   macro avg       0.93      0.93      0.93      8919
weighted avg       0.96      0.96      0.96      8919

Confusion Matrix:
6944 190
209 1576
```

```
Linear SVM Model Performance:
Accuracy: 95.54%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      7134
           1       0.89      0.88      0.89      1785

    accuracy                           0.96      8919
   macro avg       0.93      0.93      0.93      8919
weighted avg       0.96      0.96      0.96      8919

Confusion Matrix:
6945 189
209 1576
```

LDA

```
[ ]
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA

# Define the LDA object
lda = LDA(n_components=1)

# Fit and transform the training data
X_train_lda_satisfied = lda.fit_transform(X_train_satisfied, y_train_satisfied)
X_test_lda_satisfied = lda.transform(X_test_satisfied)

# Similarly for other datasets
X_train_lda_interested = lda.fit_transform(X_train_interested, y_train_interested)
X_test_lda_interested = lda.transform(X_test_interested)

X_train_lda_recommendation = lda.fit_transform(X_train_recommendation, y_train_recommendation)
X_test_lda_recommendation = lda.transform(X_test_recommendation)

X_train_lda_seasonal = lda.fit_transform(X_train_seasonal, y_train_seasonal)
X_test_lda_seasonal = lda.transform(X_test_seasonal)

```

Predicting customer satisfaction using LDA

```
[ ]
# Evaluating classifiers using LDA-transformed data
evaluation_strings_lda = {}

for name, clf in classifiers.items():
    evaluation_strings_lda[name] = evaluate_classifier_detailed(clf, X_train_lda_satisfied, y_train_satisfied

# To print the output for each classifier:
for name, result in evaluation_strings_lda.items():
    print(result)
    print("\n" + "-"*50 + "\n")
```

```
222 4254
```

```
Decision Tree Model Performance:
Accuracy: 93.03%
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      4443
           1       0.93      0.93      0.93      4476

    accuracy                           0.93      8919
   macro avg       0.93      0.93      0.93      8919
weighted avg       0.93      0.93      0.93      8919

Confusion Matrix:
4133 310
312 4164
```

```
Random Forest Model Performance:
Accuracy: 93.03%
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      4443
           1       0.93      0.93      0.93      4476

    accuracy                           0.93      8919
   macro avg       0.93      0.93      0.93      8919
weighted avg       0.93      0.93      0.93      8919

Confusion Matrix:
4133 310
312 4164
```

```
Naive Bayes Model Performance:
Accuracy: 94.92%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      4443
           1       0.95      0.95      0.95      4476

    accuracy                           0.95      8919
   macro avg       0.95      0.95      0.95      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
4212 231
222 4254
```

```
Logistic Regression Model Performance:
Accuracy: 94.92%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      4443
           1       0.95      0.95      0.95      4476

    accuracy                           0.95      8919
   macro avg       0.95      0.95      0.95      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
4216 227
226 4250
```

```
KNN Model Performance:
Accuracy: 94.06%
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      4443
           1       0.94      0.94      0.94      4476

    accuracy                           0.94      8919
   macro avg       0.94      0.94      0.94      8919
weighted avg       0.94      0.94      0.94      8919

Confusion Matrix:
4166 277
253 4223
```

```
Kernel SVM Model Performance:
Accuracy: 94.92%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      4443
           1       0.95      0.95      0.95      4476

    accuracy                           0.95      8919
   macro avg       0.95      0.95      0.95      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
4212 231
222 4254
```

```
Linear SVM Model Performance:
Accuracy: 94.92%
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      4443
           1       0.95      0.95      0.95      4476

    accuracy                           0.95      8919
   macro avg       0.95      0.95      0.95      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
4218 225
228 4248
```

Predicting cusotmer interest using LDA

```
# Evaluating classifiers using LDA-transformed data
evaluation_strings_lda = {}

for name, clf in classifiers.items():
    evaluation_strings_lda[name] = evaluate_classifier_detailed(clf, X_train_lda_interested, y_train_inte

# To print the output for each classifier:
for name, result in evaluation_strings_lda.items():
    print(result)
    print("\n" + "-"*50 + "\n")
```

```
2189 2230
```

```
Decision Tree Model Performance:
Accuracy: 49.68%
Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.49      0.49      4500
           1       0.49      0.51      0.50      4419

    accuracy                           0.50      8919
   macro avg       0.50      0.50      0.50      8919
weighted avg       0.50      0.50      0.50      8919

Confusion Matrix:
2197 2303
2185 2234
```

```
Random Forest Model Performance:
Accuracy: 49.70%
Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.49      0.49      4500
           1       0.49      0.51      0.50      4419

    accuracy                           0.50      8919
   macro avg       0.50      0.50      0.50      8919
weighted avg       0.50      0.50      0.50      8919

Confusion Matrix:
2198 2302
2184 2235
```

```
Naive Bayes Model Performance:
Accuracy: 53.35%
Classification Report:
              precision    recall  f1-score   support

           0       0.54      0.52      0.53      4500
           1       0.53      0.54      0.54      4419

    accuracy                           0.53      8919
   macro avg       0.53      0.53      0.53      8919
weighted avg       0.53      0.53      0.53      8919

Confusion Matrix:
2359 2141
2020 2399

--------------------------------------------------
```

```
Kernel SVM Model Performance:
Accuracy: 53.08%
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.56      0.54      4500
           1       0.53      0.50      0.52      4419

    accuracy                           0.53      8919
   macro avg       0.53      0.53      0.53      8919
weighted avg       0.53      0.53      0.53      8919

Confusion Matrix:
2504 1996
2189 2230
```

```
KNN Model Performance:
Accuracy: 50.79%
Classification Report:
              precision    recall  f1-score   support

           0       0.51      0.51      0.51      4500
           1       0.50      0.51      0.51      4419

    accuracy                           0.51      8919
   macro avg       0.51      0.51      0.51      8919
weighted avg       0.51      0.51      0.51      8919

Confusion Matrix:
2285 2215
2174 2245
```

```
Linear SVM Model Performance:
Accuracy: 53.11%
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.55      0.54      4500
           1       0.53      0.52      0.52      4419

    accuracy                           0.53      8919
   macro avg       0.53      0.53      0.53      8919
weighted avg       0.53      0.53      0.53      8919

Confusion Matrix:
2454 2046
2136 2283
```

```
Logistic Regression Model Performance:
Accuracy: 53.10%
Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.55      0.54      4500
           1       0.53      0.52      0.52      4419

    accuracy                           0.53      8919
   macro avg       0.53      0.53      0.53      8919
weighted avg       0.53      0.53      0.53      8919

Confusion Matrix:
2454 2046
2137 2282
```

Predicting product recommendation likelyhood using LDA

```
# Evaluating classifiers using LDA-transformed data
evaluation_strings_lda = {}

for name, clf in classifiers.items():
    evaluation_strings_lda[name] = evaluate_classifier_detailed(clf, X_train_lda_recommendation, y_train

# To print the output for each classifier:
for name, result in evaluation_strings_lda.items():
    print(result)
    print("\n" + "-"*50 + "\n")
```

```
Logistic Regression Model Performance:
Accuracy: 89.85%
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       905
           1       0.90      1.00      0.95      8014

    accuracy                           0.90      8919
   macro avg       0.45      0.50      0.47      8919
weighted avg       0.81      0.90      0.85      8919

Confusion Matrix:
0 905
0 8014
```

```
KNN Model Performance:
Accuracy: 89.31%
Classification Report:
              precision    recall  f1-score   support

           0       0.14      0.01      0.02       905
           1       0.90      0.99      0.94      8014

    accuracy                           0.89      8919
   macro avg       0.52      0.50      0.48      8919
weighted avg       0.82      0.89      0.85      8919

Confusion Matrix:
9 896
57 7957
```

```
Naive Bayes Model Performance:
Accuracy: 89.85%
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       905
           1       0.90      1.00      0.95      8014

    accuracy                           0.90      8919
   macro avg       0.45      0.50      0.47      8919
weighted avg       0.81      0.90      0.85      8919

Confusion Matrix:
0 905
0 8014
```

```
Linear SVM Model Performance:
Accuracy: 89.85%
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       905
           1       0.90      1.00      0.95      8014

    accuracy                           0.90      8919
   macro avg       0.45      0.50      0.47      8919
weighted avg       0.81      0.90      0.85      8919

Confusion Matrix:
0 905
0 8014
```

```
Kernel SVM Model Performance:
Accuracy: 89.85%
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       905
           1       0.90      1.00      0.95      8014

    accuracy                           0.90      8919
   macro avg       0.45      0.50      0.47      8919
weighted avg       0.81      0.90      0.85      8919

Confusion Matrix:
0 905
0 8014
```

```
Random Forest Model Performance:
Accuracy: 82.13%
Classification Report:
              precision    recall  f1-score   support

           0       0.11      0.10      0.10       905
           1       0.90      0.90      0.90      8014

    accuracy                           0.82      8919
   macro avg       0.50      0.50      0.50      8919
weighted avg       0.82      0.82      0.82      8919

Confusion Matrix:
92 813
781 7233
```

```
Decision Tree Model Performance:
Accuracy: 82.13%
Classification Report:
              precision    recall  f1-score   support

           0       0.11      0.10      0.10       905
           1       0.90      0.90      0.90      8014

    accuracy                           0.82      8919
   macro avg       0.50      0.50      0.50      8919
weighted avg       0.82      0.82      0.82      8919

Confusion Matrix:
92 813
781 7233
```

Predicting seasonal shoppers using LDA

```python
# Evaluating classifiers using LDA-transformed data
evaluation_strings_lda = {}

for name, clf in classifiers.items():
    evaluation_strings_lda[name] = evaluate_classifier_detailed(clf, X_train_lda_seasonal, y_train_seasonal, X

# To print the output for each classifier:
for name, result in evaluation_strings_lda.items():
    print(result)
    print("\n" + "-"*50 + "\n")
```

```
Logistic Regression Model Performance:
Accuracy: 95.29%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      7134
           1       0.89      0.87      0.88      1785

    accuracy                           0.95      8919
   macro avg       0.93      0.92      0.93      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
6940 194
226 1559
```

```
KNN Model Performance:
Accuracy: 94.80%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      7134
           1       0.87      0.87      0.87      1785

    accuracy                           0.95      8919
   macro avg       0.92      0.92      0.92      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
6896 238
226 1559
```

```
Linear SVM Model Performance:
Accuracy: 95.30%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      7134
           1       0.89      0.87      0.88      1785

    accuracy                           0.95      8919
   macro avg       0.93      0.92      0.93      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
6941 193
226 1559
```

```
Kernel SVM Model Performance:
Accuracy: 95.32%
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      7134
           1       0.89      0.88      0.88      1785

    accuracy                           0.95      8919
   macro avg       0.93      0.93      0.93      8919
weighted avg       0.95      0.95      0.95      8919

Confusion Matrix:
6934 200
217 1568
```

Naive Bayes Model Performance:
Accuracy: 95.32%
Classification Report:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	7134
1	0.88	0.88	0.88	1785
accuracy			0.95	8919
macro avg	0.93	0.93	0.93	8919
weighted avg	0.95	0.95	0.95	8919

Confusion Matrix:
6926 208
209 1576

Random Forest Model Performance:
Accuracy: 92.95%
Classification Report:

	precision	recall	f1-score	support
0	0.96	0.95	0.96	7134
1	0.82	0.83	0.83	1785
accuracy			0.93	8919
macro avg	0.89	0.89	0.89	8919
weighted avg	0.93	0.93	0.93	8919

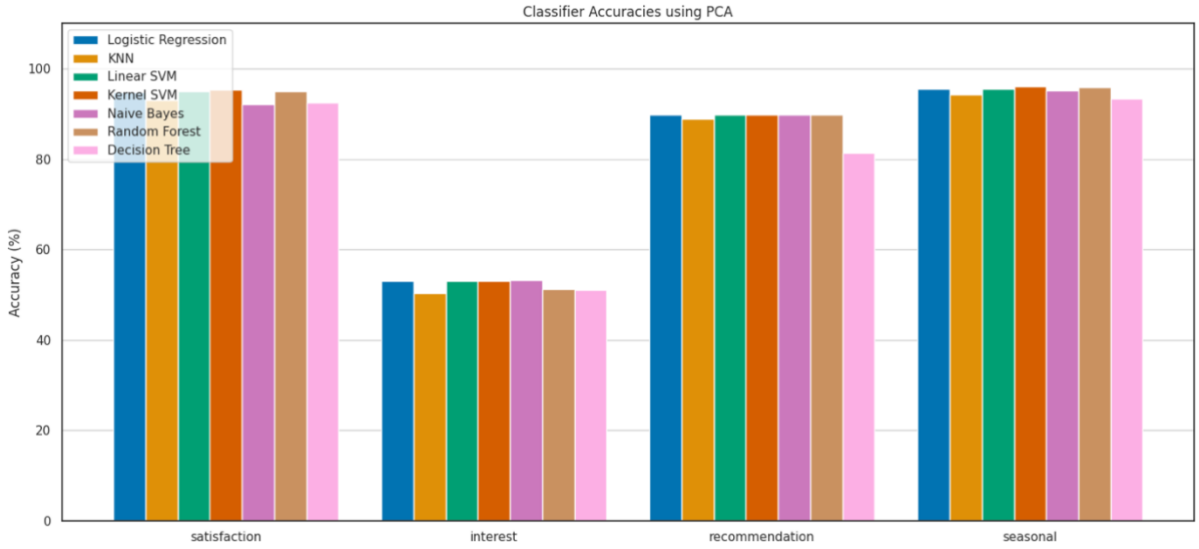
Confusion Matrix:
6803 331
298 1487

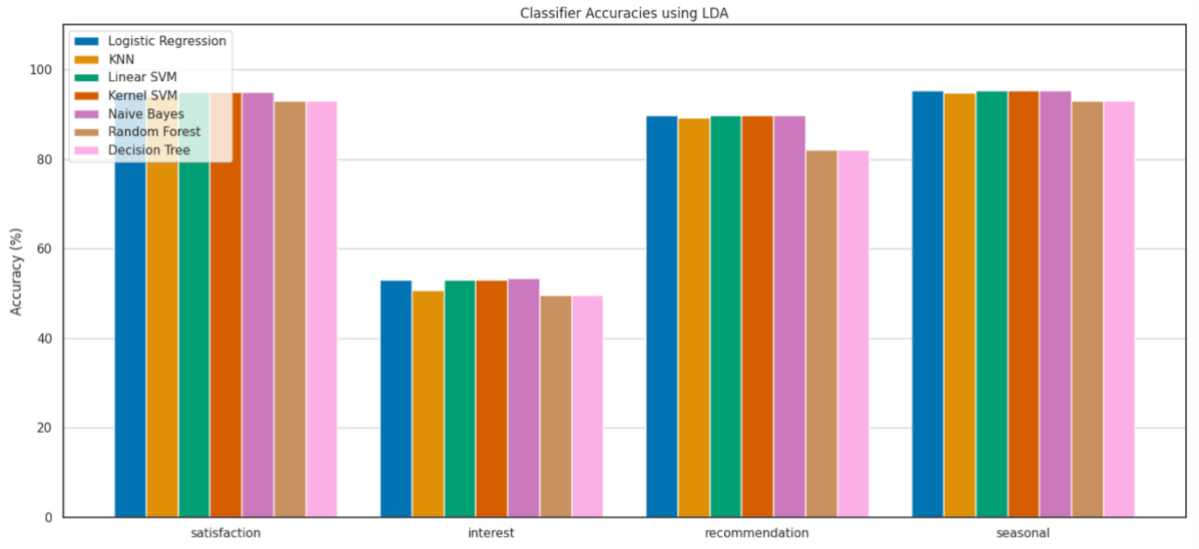
Decision Tree Model Performance:
Accuracy: 92.95%
Classification Report:

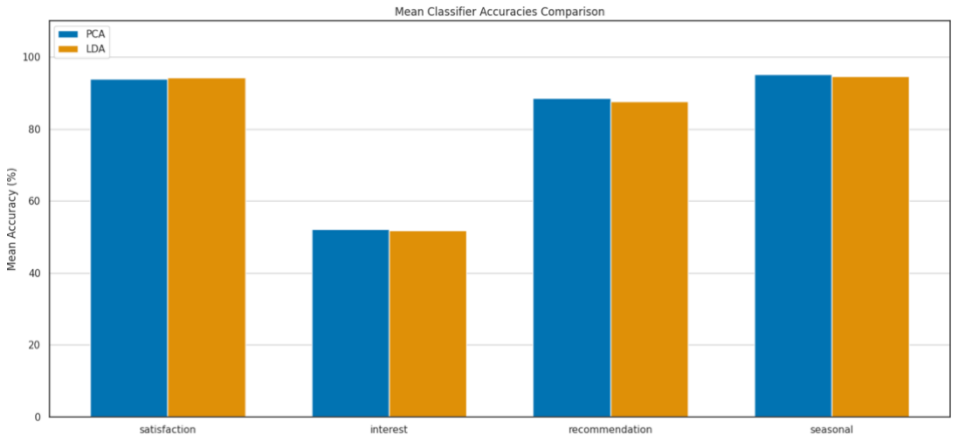
	precision	recall	f1-score	support
0	0.96	0.95	0.96	7134
1	0.82	0.83	0.83	1785
accuracy			0.93	8919
macro avg	0.89	0.89	0.89	8919
weighted avg	0.93	0.93	0.93	8919

Confusion Matrix:
6803 331
298 1487

Comparing PCA vc LDA results







Market Basket Analysis

```
[ ]  1  mba_data = df_encoded[[col for col in df_encoded.columns if 'Category_' in col or 'Style Attributes_' in col]]
     2  mba_data.head()
```

	Category_Activewear	Category_Bottoms	Category_Dresses	Category_Footwear	Category_Jewelry	Category_Lingerie
0	0	0	0	1	0	0
1	0	0	0	0	0	0
2	0	0	0	1	0	0
3	0	0	0	0	0	0
4	0	1	0	0	0	0

```
[ ]  1  from mlxtend.frequent_patterns import apriori
     2  from mlxtend.frequent_patterns import association_rules
     3
     4  frequent_itemsets = apriori(mba_data, min_support=0.01, use_colnames=True)
     5  rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.1)
     6  rules.sort_values(by='lift', ascending=False).head()
```

```
/usr/local/lib/python3.10/dist-packages/mlxtend/frequent_patterns/fpcommon.py:110: DeprecationWarning: DataFrames w
  warnings.warn(
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	z
0	(Style Attributes_Preppy)	(Category_Dresses)	0.098251	0.099462	0.011268	0.114687	1.153073	0.001496	1.017197	
1	(Category_Dresses)	(Style Attributes_Preppy)	0.099462	0.098251	0.011268	0.113290	1.153073	0.001496	1.016961	
2	(Category_Outerwear)	(Style Attributes_Casual)	0.100740	0.098318	0.010965	0.108848	1.107100	0.001061	1.011816	
3	(Style Attributes_Casual)	(Category_Outerwear)	0.098318	0.100740	0.010965	0.111529	1.107100	0.001061	1.012144	

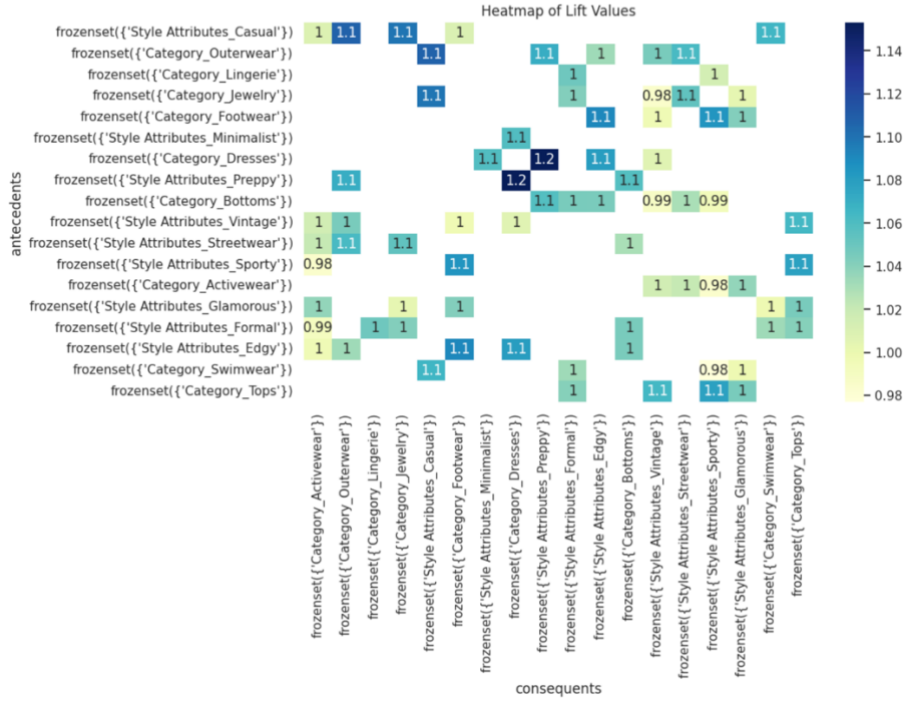
```
1  from mlxtend.frequent_patterns import apriori
2  from mlxtend.frequent_patterns import association_rules
3
4  frequent_itemsets = apriori(mba_data, min_support=0.01, use_colnames=True)
5  rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.1)
6  rules.sort_values(by='confidence', ascending=False).head()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not
  and should_run_async(code)
/usr/local/lib/python3.10/dist-packages/mlxtend/frequent_patterns/fpcommon.py:110: DeprecationWarning: DataFrames w
  warnings.warn(
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	z
25	(Style Attributes_Preppy)	(Category_Dresses)	0.098251	0.099462	0.011268	0.114687	1.153073	0.001496	1.017197	
26	(Category_Dresses)	(Style Attributes_Preppy)	0.099462	0.098251	0.011268	0.113290	1.153073	0.001496	1.016961	
51	(Style Attributes_Casual)	(Category_Outerwear)	0.098318	0.100740	0.010965	0.111529	1.107100	0.001061	1.012144	
30	(Style Attributes_Edgy)	(Category_Footwear)	0.099529	0.101951	0.011066	0.111186	1.090586	0.000919	1.010391	
35	(Category_Footwear)	(Style Attributes_Sporty)	0.101951	0.102489	0.011335	0.111184	1.084842	0.000886	1.009783	

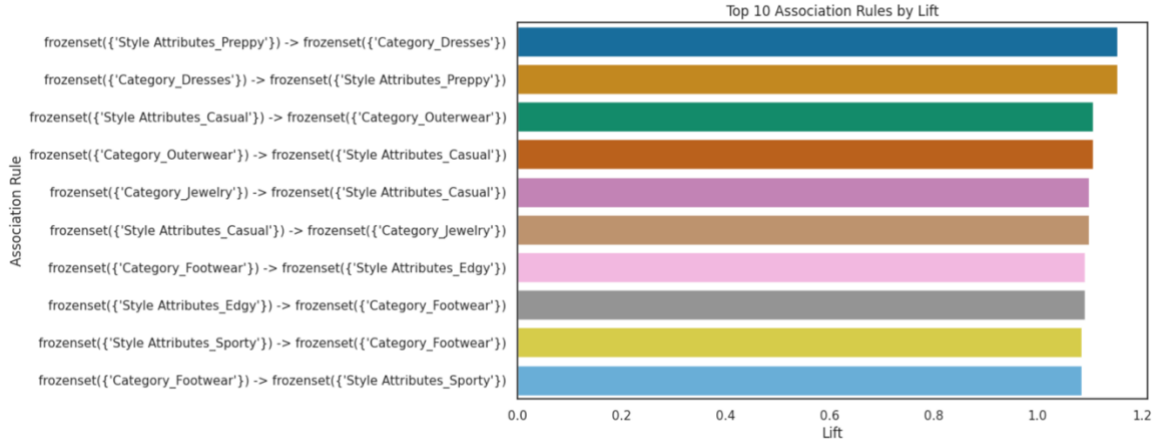
```python
#heatmap

import seaborn as sns
import matplotlib.pyplot as plt
pivot = rules.pivot(index='antecedents', columns='consequents', values='lift')
plt.figure(figsize=(10, 6))
sns.heatmap(pivot, annot=True, cmap="YlGnBu")
plt.title('Heatmap of Lift Values')
plt.show()
```

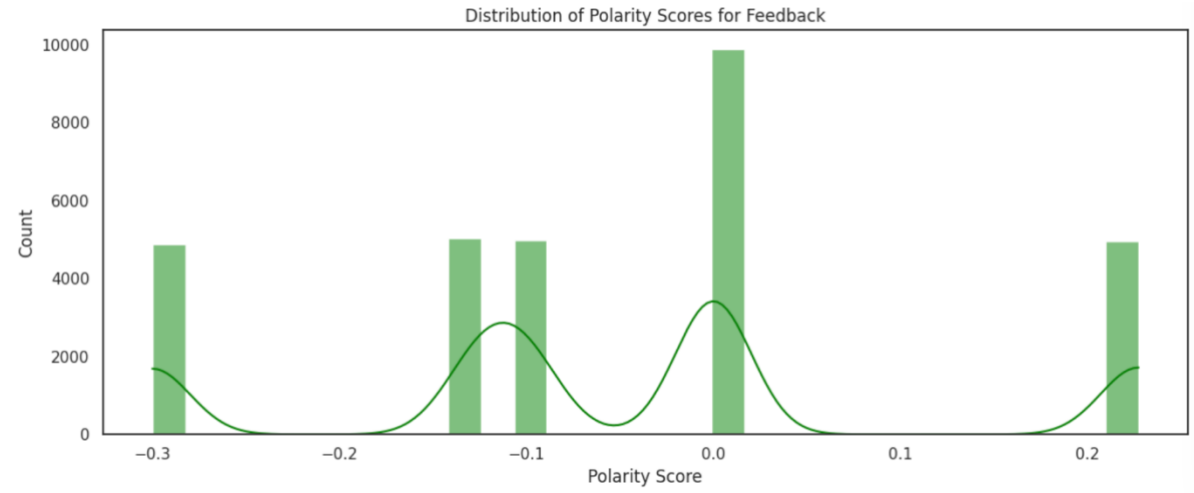
```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not
  and should_run_async(code)
```

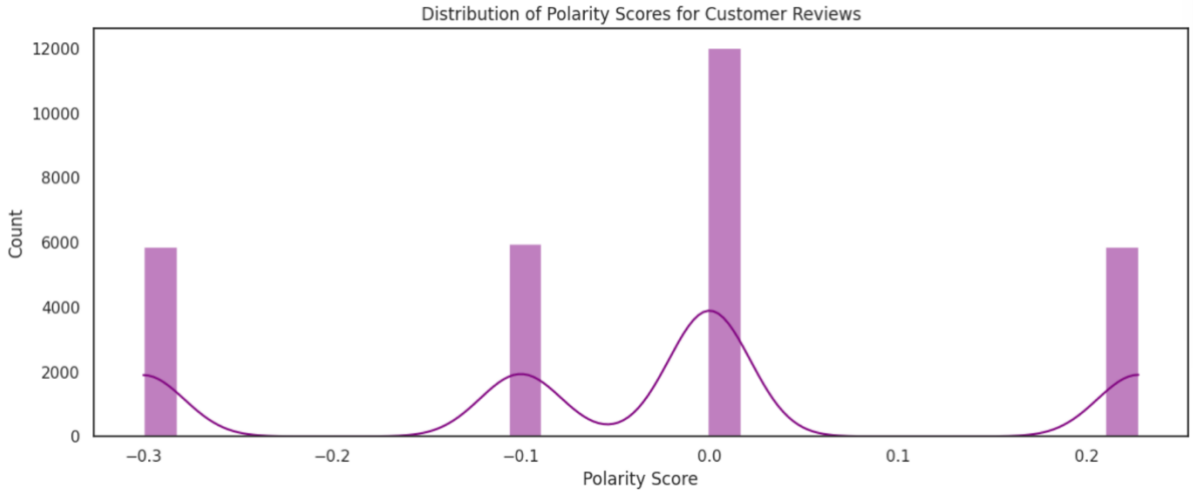


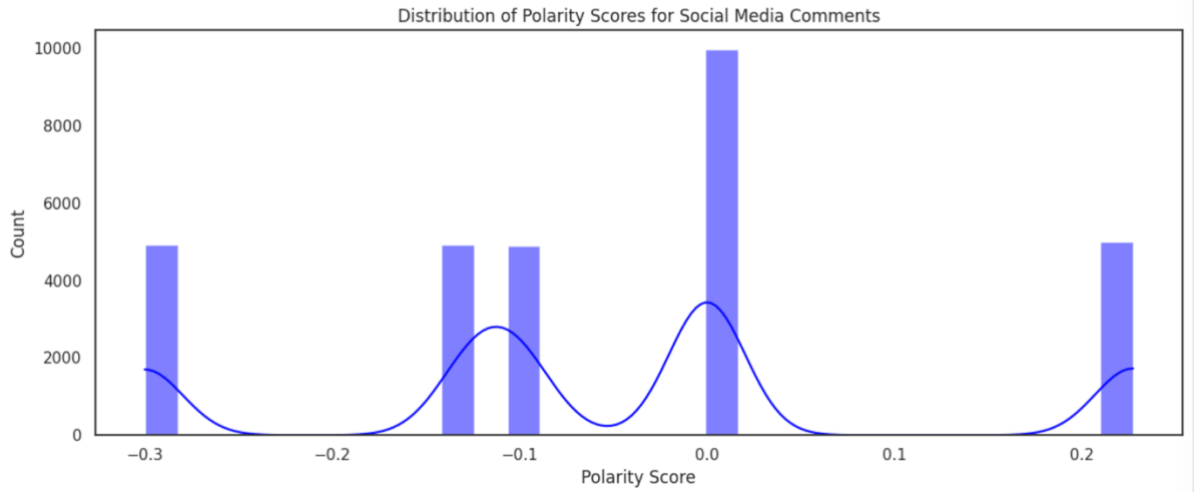

```python
# bar chart
# Sort rules by lift
top_rules = rules.sort_values(by='lift', ascending=False).head(10)

# Plot bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x=top_rules['lift'], y=top_rules['antecedents'].astype(str) + ' -> ' + top_rules['consequents'].astype(str))
plt.title('Top 10 Association Rules by Lift')
plt.xlabel('Lift')
plt.ylabel('Association Rule')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transform_cell` automatically in the future. Please pa
  and should_run_async(code)
```




Distribution of Polarity Scores for Feedback
Count
Polarity Score
10000
8000
6000
4000
2000
0
−0.3
−0.2
−0.1
0.0
0.1
0.2


Distribution of Polarity Scores for Customer Reviews
Count
Polarity Score
12000
10000
8000
6000
4000
2000
0
−0.3
−0.2
−0.1
0.0
0.1
0.2


Distribution of Polarity Scores for Social Media Comments
Count
Polarity Score
10000
8000
6000
4000
2000
0
−0.3
−0.2
−0.1
0.0
0.1
0.2