



# UNIVERSITY OF PADOVA

---

DEPARTMENT OF MATHEMATICS

*MASTER THESIS IN DATA SCIENCE*

## **LOST REVENUE RECOGNITION IN E-COMMERCE: IDENTIFYING CAUSES AND IMPLICATIONS**

*SUPERVISOR*

PROF. TOMASO ERSEGHE  
UNIVERSITY OF PADOVA

*CO-SUPERVISOR*

MATTIA ZOCCORATO  
CHIRON

*MASTER CANDIDATE*

BERKE FURKAN KUSMENOGLU

*ACADEMIC YEAR*

2022-2023



THIS THESIS IS DEDICATED TO MY FAMILY AND MY CAT NOX.  
FOR THEIR ENDLESS LOVE, SUPPORT, AND ENCOURAGEMENT.



# Abstract

This paper presents a comprehensive study aimed at improving revenue recognition and overall e-commerce platform efficiency. We begin by addressing the issue of slow-loading pages, which is a common source of revenue loss. We implement dimensionality reduction techniques and analyze feature importance through the development of a binary feature code algorithm to effectively track problematic resources. This method allows for the timely identification and resolution of issues.

Furthermore, our research investigates the complicated interpretation of bounce rates in e-commerce environments. We present an algorithm for identifying valid bounces from those that occur during specific marketing campaigns. As a result, a campaign effect score is generated, allowing for a more precise assessment of the campaign's impact on user engagement.

We propose a propensity-based product recommendation algorithm to further optimize revenue potential. We provide personalized recommendations that improve user experiences and conversions by calculating user propensity scores with the SVC decision function and assigning weighted values to product categories.

Furthermore, we explore the benefits of user behavior analysis in the detection of problems. User behavior algorithms, such as "rage clicking" and "dead clicking," reveal underlying issues such as broken links and unresponsive elements. This enables businesses to address issues and avoid revenue loss.

In conclusion, this paper highlights innovative strategies and algorithms that address critical e-commerce challenges. Businesses can optimize revenue recognition, improve user experiences, and thrive in the unstable digital marketplace by using data-driven insights. This research provides a plan of action for e-commerce businesses looking to improve their financial health and overall growth.



# Contents

ABSTRACT	v
LIST OF FIGURES	ix
LIST OF TABLES	xi
LISTING OF ACRONYMS	xiii
1 INTRODUCTION	1
2 FINDING CAUSES OF SLOW PAGES	7
2.1 Dimensionality Reduction . . . . .	8
2.1.1 Binary Encoding . . . . .	9
2.1.2 Feature Importance . . . . .	10
2.1.3 Clustering Resources . . . . .	12
2.1.4 Distribution of Feature Codes . . . . .	14
2.1.5 Analyzing Feature Codes . . . . .	16
2.1.6 Implementing Algorithm on 'Page Speed Resource' Dataset . . . . .	22
2.1.7 Bounce-Campaign Algorithm . . . . .	26
3 RECOMMENDER SYSTEM	33
3.1 Evaluating optimal funnels with user clustering and purchase propensity evaluation . . . . .	33
3.2 Creating Ratings With User Interaction . . . . .	40
3.2.1 Creating the variable . . . . .	42
3.2.2 Unveiling Hidden Patterns: Clustering Analysis with K-Means . . . . .	47
3.2.3 Calculating the Customer Propensity Of Purchase . . . . .	49
3.2.4 Product Category . . . . .	51
3.2.5 Implication of Product Category . . . . .	52
4 DETECTING ERROR WITH USER BEHAVIOUR	57
4.1 Rage Click . . . . .	59
4.1.1 Rage Click Algorithm . . . . .	59
4.2 Dead Click . . . . .	61
4.2.1 Dead Click Algorithm . . . . .	61

5	CONCLUSION	63
	REFERENCES	67
	ACKNOWLEDGMENTS	71



# Listing of figures

2.1	Creating Feature Code. . . . .	10
2.2	Feature Importance. . . . .	11
2.3	Clustering of Feature Codes . . . . .	13
2.4	Clusters behavior in different page types and user sessions. . . . .	14
2.5	Cluster Feature Codes. . . . .	15
2.6	Distribution of Most Common Feature Code. . . . .	15
2.7	Top Feature Codes. . . . .	17
2.8	Backoffice Graph. . . . .	19
2.9	Resource Details. . . . .	20
2.10	Example of caption. . . . .	21
2.11	Feature Codes with 12 Hour Frequency. . . . .	22
2.12	Backoffice Graph. . . . .	24
2.13	Problematic Resource 1. . . . .	25
2.14	Problematic Resource 2. . . . .	25
2.15	Campaigns Usage. . . . .	28
2.16	Campaign Effect Score Code. . . . .	29
2.17	Campaign Effect Score. . . . .	30
2.18	Distribution of Most Common Feature Code. . . . .	30
2.19	Back-office Bounce Rates. . . . .	31
3.1	Back-office Bounce Rates. . . . .	34
3.2	Data Flow for Recommendations. . . . .	35
3.3	Feature Importance for Recommender System. . . . .	37
3.4	First Models Performances. . . . .	38
3.5	Creating User Ratings. . . . .	40
3.6	User Page Paths. . . . .	42
3.7	Feature Importance After Total Weight. . . . .	43
3.8	Model Performances After Total Weight. . . . .	45
3.9	User Cluster. . . . .	48
3.10	User Cluster After Total Weight. . . . .	49
3.11	Category path of the product . . . . .	51
3.12	ROC Curve. . . . .	53
3.13	Cluster Propensity Score Graph. . . . .	55
4.1	Rage Click Graph. . . . .	60

4.2 Dead Click Graph. . . . . 62

# Listing of tables

3.1	Performance Metrics of Different Models . . . . .	38
3.2	Performance Metrics for Different Models After Total Weigh . . . . .	45



# Listing of acronyms

<b>PCA</b> .....	Principal Component Analysis
<b>URL</b> .....	Uniform Resource Locator
<b>SVM</b> .....	Support Vector Machine
<b>KNN</b> .....	K-Nearest Neighbors
<b>OHE</b> .....	OneHotEncoder
<b>SVC</b> .....	Support Vector Classification
<b>ROC</b> .....	Receiver Operating Characteristic
<b>AUC</b> .....	Area Under the Curve
<b>EEG</b> .....	Electroencephalogram
<b>CSS</b> .....	Cascading Style Sheets
<b>AI</b> .....	Artificial Intelligence



# 1

## Introduction

E-commerce has emerged as a dominant force in the global marketplace in today's fast-paced, digitally connected world. Online retail's ease of use and accessibility have transformed consumer behavior and traditional business models. E-commerce operates in a variety of market segments and is conducted on computers, tablets, smartphones, and other intelligent devices[1]. This provides numerous benefits to consumers, businesses, and the economy of the countries. E-commerce expansion can benefit the domestic economy, as the coefficient of scale for online advertising has the highest value, accelerating investment in online advertising, expanding opportunities for growth, and increasing overall economic expansion[2]. Consumers can easily find what they are looking for, identify alternative options, and select the optimal product for themselves. Businesses that have already connected with a certain number of people can effectively expand their customer base and increase their product sales in a more profitable manner. Also, e-commerce enables new business actors to enter the market without any barriers because online platforms are easily accessible to everyone. With all the positive impacts of e-commerce on different sectors and different parties, managing the e-commerce site effectively requires hard work and lots of attributes. Despite the rapid growth and opportunities presented by e-commerce, there are major challenges that risk the financial health of companies operating in this industry. One such difficulty is possible lost revenue, which requires careful consideration and innovative solutions.

Since demand for seamless online experiences increases, so does the importance of efficient revenue recognition for e-commerce businesses. However, this expanding industry brings lots

of challenges with itself. Due to the fact that something of this kind contains so many features, it is also open to problems and improvements. Everyone who engages in e-commerce as a customer may encounter problems while visiting a website. This issue could be caused by the website or by the user. In these instances, these issues must be minimized to avoid revenue loss for the organization. Providing examples of these issues could help the reader comprehend the purpose of this paper. Commonly online, slow-loading pages can frustrate and discourage potential customers, resulting in abandoned shopping carts and revenue loss. A lot of research has been done in the past on the change in satisfaction levels depending on the delay time on the user's website. Hoxmeier and DiCesare's (2000) study found a significant connection between satisfaction and delay in information retrieval tasks. The highest satisfaction level was found in no delay, while 3-9 seconds satisfaction remained relatively stable. A 12-second delay resulted in a noticeable drop in satisfaction[3]. The issues could be the result of problematic web resources, such as broken links or ineffective checkout processes, which slow down the flow of transactions and the recognition of revenue. When a user is unable to view the homepage or the details about the item, it is a major issue because it damages customer trust. There are many cases in which users decide to make a purchase while visiting a website, even if they did not intend to buy the product or it was not necessary at the moment. When they encounter slow pages or problems during the checkout process, this is likely to be a deal-breaker for them. These numerous issues illustrate the complexity of revenue recognition in the e-commerce industry and the need to identify and address the underlying causes of lost revenue.

If you are an owner of an e-commerce business, you must have a solid awareness of the importance of the resources on your online platform. These resources include an extensive variety of data that defines the fundamental characteristics of your website. Particularly, they include image resources, CSS resources, and JavaScript resources, which collectively represent the functional and visual aspects of your digital storefront. After entering the website, an interaction begins and pages building elements such as images, banners, and text. This integration requires gathering and creating various resources, which support user interactions. However, challenges may emerge in the form of errors that can disrupt the user experience. These issues can arise from a variety of causes, including server irregularities, connectivity limitations, and coding errors, among others. These kinds of problems are very common in the e-commerce sector but the most important thing is how effectively the business can solve the problem. Finding these problems is the most problematic thing and tracking these issues is very hard to achieve. Our purpose in this paper is to create an algorithm to detect these problems and report them with detailed information.



Measuring website performance and content efficiency can be done by analyzing different aspects. One of the most common aspects in the e-commerce sector is Bounce Rate. Bounce Rate is an important metric for determining user engagement with a website. A lower Bounce Rate indicates increased user involvement and deeper content exploration, allowing organizations to improve visitor engagement. However, in this realm of user interactions, the term "bounce" refers to situations in which a user's journey begins and ends on a single webpage, with no further engagement. Campaigns are important in this because users who arrive via campaign links may leave quickly, possibly due to a lack of interest in the campaign content.

This interaction between campaigns and user behavior highlights the importance of a distinct analytical perspective. Importantly, campaign-related bounces do not necessarily indicate website issues, but rather reflect the complex patterns of user preferences and behaviors in the digital world. To address this, an algorithm was created to improve the bounce rate analysis. This algorithm focuses on the basic concept of bounces, including metrics such as total pages visited and interactions to provide a more comprehensive picture of user engagement and behavior. The outcomes of this algorithmic improvement are, providing clarity in distinguishing between insignificant bounces and those influenced by campaign-related anomalies. As we delve into the complexities of data analysis and assigning values to the "bounce" column, we come across a variety of scenarios that help us better understand user behavior. Finally, this analysis is based on a binary judgment of "True" or "False" in the "bounce" column. This examination is essential for making informed decisions about website performance and marketing strategies. We gain knowledge of marketing effectiveness and potential website issues by calculating the campaign effect score, making this algorithm a critical tool for organizations looking to optimize their online presence.

E-commerce business does not lose revenue only because of problems from the website. Since the improvement of technology and the impact of AI in business, lots of new and very effective approaches discovered. One of the most important ones is the Recommender System. Recommender System, developed in 1995, is a tool for interacting with complex information spaces, providing personalized views, and prioritizing items of interest. Methods of artificial intelligence are employed such as machine learning, data mining, and user modeling. These systems are essential to e-commerce sites like Amazon and Netflix, which conduct business online[4]. Companies without these technologies miss lots of revenue opportunities and in this sector, lots of big names as we mentioned, implement this system into their business and they experience lots of positive returns[5]. In this paper, we are analyzing user behavior and recommend a new product to a user that they might be interested in. Understanding user

behavior is crucial for businesses in the e-commerce industry, as it allows for accurate forecasting of purchase likelihood and enables businesses to optimize their marketing campaigns. We created an algorithm and studied diverse user interactions to provide businesses with valuable insights by studying and decoding interactions between products and customers. This helps identify possible purchase patterns and creates customized interactions to attract consumers and increase conversion rates. An extensive dataset derived from a company's digital domain serves as the foundation for an algorithm that analyzes user behavior, including demographics, product properties, interaction metrics, and session details. The algorithm's path begins with careful data preprocessing and feature engineering, which refines the dataset for detailed investigation. The algorithm then executes exploratory data analysis, revealing crucial trends and hidden clusters resulting from user attributes, and after that uses machine learning, navigating the complex of predictive modeling, taking into account both consumer behavior and product preferences. This algorithm goes beyond data manipulation and becomes an indicator of direction for companies seeking to establish deeper connections with their customers. It allows for understanding user intentions and personalizing experiences that resonate with user expectations.

We mentioned the importance of user behavior and its impact on the business. During the research, we realized we could use customer behavior to detect problems. Understanding user behavior is crucial for organizations to improve their online platforms and maximize income in the rapidly evolving world of e-commerce. The web allows direct communication between suppliers of goods and services and their clients. Combined with the capability to collect detailed data at the level of individual mouse clicks, this presents an enormous potential for customizing the web experience for clients. There has been a recent increase in the amount of research conducted on different aspects of the personalization issue. Various web-based businesses rely heavily on human participation to collect user profile data in the majority of their current personalization strategies[6]. Deep studies into user activity patterns are essential for identifying and fixing problems that could lead to lost revenue. E-commerce businesses can identify errors that may limit their income potential by identifying deviations, anomalies, and patterns within this behavioral data. Research highlights the importance of recognizing user annoyance and frustration as significant indicators of website difficulties that may contribute to revenue loss. Studies have shown that analyzing user behavior can help identify problems like "rage clicking" and "dead clicks," which can lead to broken links, unresponsive buttons, or confused interfaces. Rage clicking is a common issue online, where users repeatedly click on unreliable or poorly functioning features to achieve desired results. This behavior can be attributed to

technological issues, delayed loading times, complex navigation, and imprecise user interfaces. Rage clicking in online retail serves as a clear indicator of user dissatisfaction and highlights areas that need attention. Dead clicks are instances where users engage with a website or application by clicking on non-functional items, often leading to annoyance and misunderstanding. Addressing dead clicks can improve user experience and overall usability, allowing businesses to enhance navigation, customer satisfaction, and e-commerce platforms by reducing dead clicks and promoting meaningful interactions. We created algorithms to detect this click in the most accurate way and use these pieces of information to detect anomalies.

To summarize, e-commerce is a dynamic environment that provides significant opportunities for growth and innovation while also presenting challenges that require creative solutions. As this paper delves into the complex relationship of revenue recognition, user behavior, and technological advancements, it becomes clear that the success of the digital storefront is dependent on the integration of user experience optimization and technical excellence. Businesses can unlock the full potential of e-commerce by addressing issues that disrupt user satisfaction and identifying patterns that enable personalized interactions. The algorithms and methodologies outlined in this paper pave the way for a future in which revenue leakages are minimized, user experiences are improved, and the digital world becomes an effective marketplace for both consumers and businesses. As technology advances, this paper serves as guidance, providing strategies for navigating the broad world of e-commerce with clarity and success.



# 2

## Finding Causes of Slow Pages

In this paper, the primary goal is to investigate the issue of lost revenue recognition in e-commerce, particularly focusing on the identification of the reasons behind slow-loading pages and its implications on revenue recognition. The growing significance of e-commerce as a major revenue source for businesses has made it crucial to optimize website performance and improve user experience. Slow-loading pages can result in frustrated users, increased bounce rates, and potential lost revenue. To address this challenge, we aim to simplify the complex data and look in detail at the factors that cause slow-loading pages.

Our dataset consists of various variables, including categorical and numeric ones, making the analysis process challenging. To gain meaningful information, we employ data analysis techniques and explore the relationship between different variables. By identifying the root causes of slow-loading pages, we can create effective strategies to mitigate them and improve revenue recognition.

One of the key approaches we employ is to prevent unnecessary resource reloading. By tracking resource changes, we can determine if any modifications have occurred, leading to performance issues. To achieve this, we leverage clustering methods to identify problematic resources that exceed a certain threshold. These resources can be clustered based on various factors, such as page type, URL, resource size, and initiator type, among others.

For instance, we consider an e-commerce website with numerous different pages. By clustering resources based on their attributes, we can group similar resources depending on their occurrences on the same pages. This allows us to monitor and compare the performance of re-

sources within the same cluster easily. Consequently, we can identify any performance-related changes or issues affecting specific clusters. However, as resource characteristics change over time, we need to periodically reevaluate and update the clusters accordingly.

There can be a scenario where a particular product's URL or location is modified due to website updates or restructuring. In such cases, the resource should be relocated to a different cluster that aligns better with its new attributes. This dynamic clustering process helps us maintain accurate performance monitoring and detect changes in the performance of specific resources.

Nevertheless, handling a large number of cases presents a challenge in effectively clustering the resources. Identifying meaningful patterns from this enormous amount of data can be challenging due to the volume of resources. Addressing this challenge involves the selection of appropriate clustering algorithms, feature engineering, and continuous monitoring to ensure the clustering remains up-to-date.

By understanding the underlying causes of slow-loading pages and employing clustering techniques, valuable insights can be uncovered, and that will assist e-commerce businesses in optimizing their websites. The implications of this research can help businesses prioritize resource allocation and focus on performance enhancements, ultimately leading to better user experiences and increased revenue recognition in the highly competitive e-commerce landscape.

## 2.1 DIMENSIONALITY REDUCTION

Dimensionality reduction is an important method for addressing the difficulties caused by datasets with high dimensions. It reduces the degrees of freedom in results by representing them with a smaller, condensed set of variables, and also helps reduce the computational load in further steps[7]. These datasets, which are frequently characterized by a large number of features or attributes, can result in computational errors, an increase in complexity, and even problems with visualization and interpretation. Dimensionality reduction provides a method to simplify and improve the analysis of data while keeping its essential characteristics as a solution to these issues.

Fundamentally, dimensionality reduction involves transforming a dataset with numerous variables into a simplified representation that encapsulates the most important information. Dimensionality reduction can be achieved through a variety of methods, each with its own principles and techniques. These techniques can be applied to improve the efficiency of learning algorithms, thus improving the accuracy of the prediction of the different classifiers [8].

These methods can be generally classified as either linear or nonlinear. We need to imply dimensionality reduction to our features because our initial plan is to track the resources with useful information. Every resource has various attributes and information and we want to choose the most important ones to make further analysis to achieve high performance.

### 2.1.1 BINARY ENCODING

After the research on dimensionality reduction, we generated an algorithm that uses binary encoding to represent complex combinations of categorical variables in resource clustering. The algorithm's main purpose is to simplify the representation of selected variables while preserving their value of information. By creating a binary code for each combination of values in these columns, we enable efficient further analysis and modeling. By mapping categorical values to integers, we encoded them in a binary format. For example, if we have 8 categories, we would need 3 bits  $\log_2(8)$  to represent them. [9]. Each data point's feature values are then represented as a binary vector where each bit corresponds to a category.

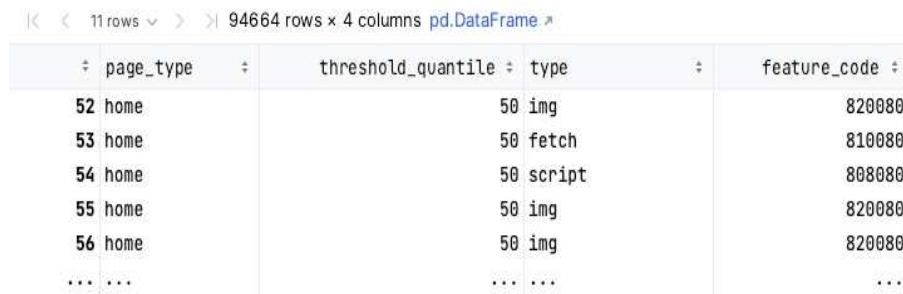
The first step of the algorithm involves using the "unique()" function in Python to generate a list of all unique combinations of values in the three selected columns. This function is commonly found in Python libraries like NumPy or pandas and is designed to extract unique values from an array or series. It ensures that each distinct combination is represented uniquely in the subsequent binary encoding process. By identifying and retaining only unique values while discarding duplicates, helps cover the full range of variations in these categorical variables.

In the next part, the algorithm examines the values of the selected columns in each row of the dataset. It searches each row for a match with one of the previously generated unique combinations. When a match is discovered, the corresponding bit in a binary code is set to 1 for that particular combination. This procedure is repeated for each row in the dataset, producing a unique binary code for each resource.

Binary encoding is a good way to represent complex combinations of categorical variables. Unlike traditional one-hot encoding, which requires a unique binary feature for each value combination, our algorithm generates a single binary code for each value combination. This method reduces the number of dimensions, making it less vulnerable to the curse of dimensionality and more computationally efficient.

We can use clustering algorithms to group similar resources by using binary codes as features. Because these binary representations capture relationships between categorical variables, we can identify patterns and similarities among resources with similar attribute combinations. In

domains such as e-commerce, converting categorical data into binary codes improves resource management, optimization, and decision-making.



	page_type	threshold_quantile	type	feature_code
52	home	50	img	820080
53	home	50	fetch	810080
54	home	50	script	808080
55	home	50	img	820080
56	home	50	img	820080
...	...	...	...	...

Figure 2.1: Creating Feature Code.

In the 2.1, we can get an idea of how the algorithm works. Three categorical features are used to create a unique feature code for each case. In the table, `threshold_quantile` and `page_type` values are the same but the `type` which refers to the type of the resource, is the value that has different values. When we look at the feature codes, the beginning of the codes is the same because the first two value is the same in every row. We can see the differences in the second digit of the feature code when the type changes. Also, when the type has the same values the feature code is the same. This was a small exercise to understand how the algorithm works. We can add more features and with that, our feature code will increase in digits but this is not affecting the algorithm in a negative way. The aim is to cover every possible case and combination.

### 2.1.2 FEATURE IMPORTANCE

Before proceeding with the binary encoding algorithm, it is necessary to evaluate the importance of features. Applying a Random Forest algorithm allows valuable information into how is the importance and contribution of various characteristics to the overall resource clustering process. In the case of data sets containing a larger amount of variables, the selection of feature subsets becomes essential. It eliminates minor variables and generates efficient and improved prediction performance on class variables, resulting in a more accurate and affordable understanding of the data. The random forest has become known as a highly effective and robust algorithm capable of handling feature selection problems even with a large number of variables[10]. By holding what features are the most informative, we can make informed decisions about which variables to choose for future algorithmic steps.



The dataset had to be modified before running the feature importance analysis. We convert categorical variables to integers because Random Forest only accepts numeric input. This transformation keeps the variables' categorical structure while allowing them to be included in the feature analysis process.

To assure compatibility with the Random Forest algorithm, the variable value ranges are also modified. Calculating the minimum and maximum values of each column enables the determination of optimal data ranges that cover the entire data spectrum. This phase reduces bias and guarantees that no feature gets lost due to various value scales.

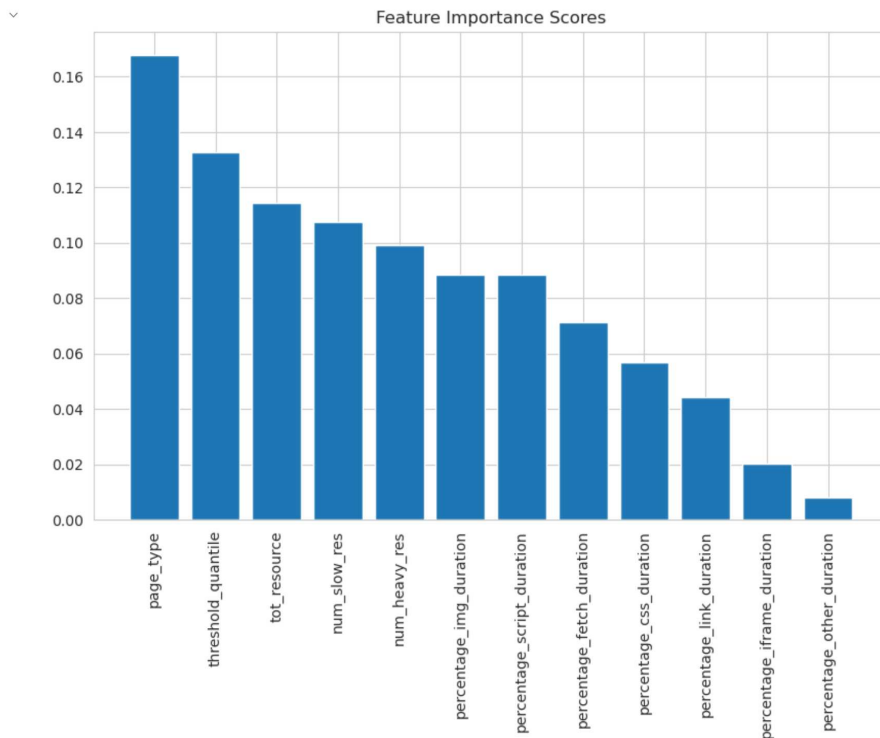


Figure 2.2: Feature Importance.

For categorical variables to be meaningfully represented in the feature importance analysis, they must be handled carefully during data preparation. The conversion of categorical variables to numeric formats maintains the accuracy of the original data while expanding the scope of the analysis.

Once the dataset has been appropriately modified, the Random Forest algorithm can be used to determine the significance of features. Random Forest performs well at determining the importance of each feature as a method of ensemble learning by creating multiple decision

trees and combining their predictions. Those with higher priority scores are considered more influential in the clustering procedure.

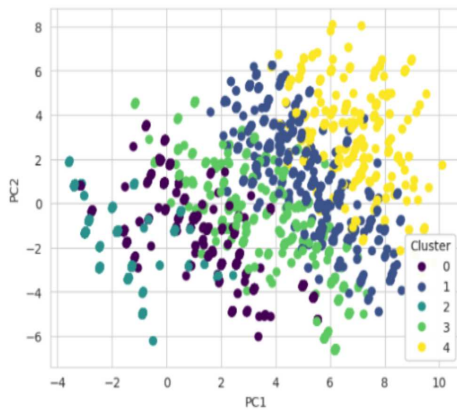
In 2.2, we can observe and select the most relevant variables for our following binary encoding algorithm based on the result we gain from the feature importance analysis. By identifying the most informative features, we may focus on those that significantly contribute to the overall resource clustering process, leading to more accurate and meaningful results.

To sum up, what we want to achieve in this part, before implementing the binary encoding algorithm, it is important for our research to conduct a feature importance analysis. By preparing the dataset, converting categorical variables, and modifying value ranges, we ensure the dataset's compatibility with the algorithm. These findings prepare the way for more efficient feature selection along with a more robust and reliable resource clustering procedure.

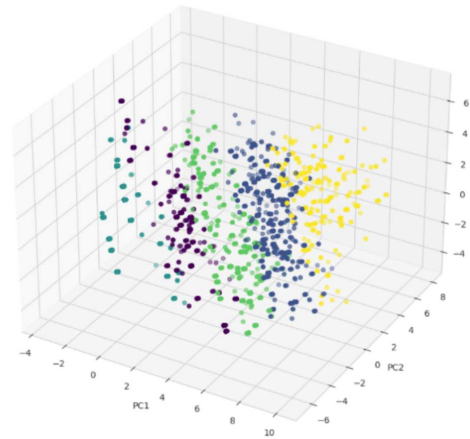
### 2.1.3 CLUSTERING RESOURCES

K-means clustering is a traditional unsupervised learning algorithm with a high degree of efficiency and simplicity. Due to its iterative nature, it is utilized in numerous fields. The algorithm employs distance as a measurement standard, generates  $k$  clusters in the data set, computes the average distance value, and computes the initial centroid [11]. In resource clustering, unique feature identifiers are assigned to each case as the basis for clustering similar resources. Through the algorithm for binary encoding, these feature codes contain complex combinations of categorical variables in a binary representation. By employing these feature codes for clustering, we can identify meaningful patterns and similarities among the cases, thus gaining a valuable understanding of the factors influencing their performance.

To achieve resource clustering based on the feature code, appropriate clustering algorithms are implemented. These algorithms evaluate the similarity between feature codes using distance metrics and grouping resources with similar binary patterns into clusters. As a result, resources with comparable combinations of categorical variable values are grouped together, assisting the identification of common characteristics among them. In this context, our primary objective was to assess the effectiveness of the feature code in creating meaningful clusters, with the aim of determining its suitability for further analysis.



(a) 2D Scatter Plot



(b) 3D Scatter Plot

**Figure 2.3:** Clustering of Feature Codes

Clustering based on feature codes allows us to effectively detect anomalies and outliers. Resources with different or unusual combinations of categorical variables can vary from the main clusters, indicating possible research opportunities. These outliers may represent specific cases of unusual user behaviors, which might require specialized attention and personalized techniques. Figure 2.3 shows the scatter plot using PCA. Combining PCA and K-means clustering is an effective strategy for improving the precision and efficiency of clustering in high-dimensional datasets. By first applying PCA, the dimensionality of the data is reduced while its essential variance is preserved, leading to improved interpretability and noise reduction. This simplified representation enables more precise K-means clustering, as the reduced dimensions reduce computational burdens and highlight meaningful patterns[12]. Plots show feature code can be a strong feature for clustering the resources. Also, this plot does not give useful information about what is the behavior of clusters. To get useful information, focusing on specific cases was needed. For instance, provide an analysis that tells us which clusters are most common and how frequently they appear within specific page types. In figure 2.4, we can see the behavior of the clusters in different page types. For this plot, a short amount of data to used to observe more clearly. Most of the cluster appears on the home page which was expected and for cluster '0' we can say it appears more in users' sessions. There are clear reasons why it is like this because home page is generally the first page where users interact and most of the resources are downloaded on this page. Another case is, on the product page which refers to the number '3' in the plot, and only cluster '2' appears on that page. It can be specific resources for the product page which might be the product images.

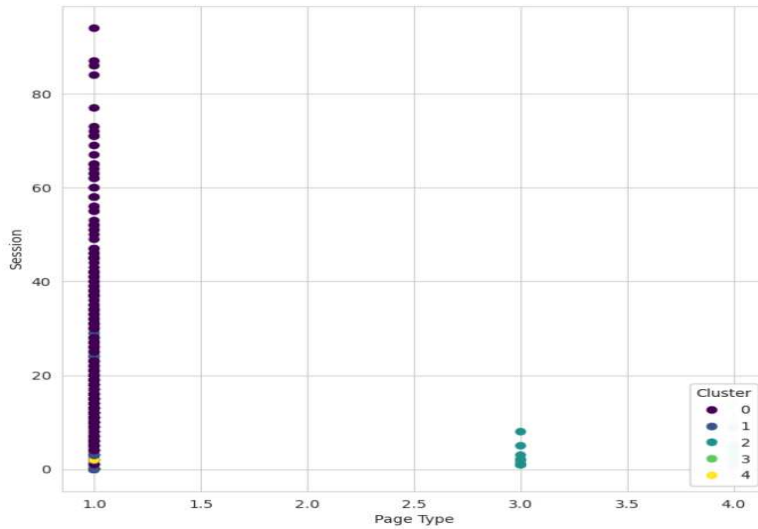


Figure 2.4: Clusters behavior in different page types and user sessions.

Using the unique feature codes assigned to each case in our clustering procedure allows useful analysis and decision-making. By grouping comparable resources together, it is possible to identify patterns, identify outliers, and comprehend the underlying factors influencing resource performance. Also, being able to see the resource’s behavior in different variables is valuable information for businesses. We can easily check the cluster’s behavior on different continuous and categorical variables to detect possible problems.

#### 2.1.4 DISTRIBUTION OF FEATURE CODES

In this part, analyzed made for the distribution of feature codes within clusters over the time period. The investigation of time provided significant information into the behavior of resources and shows how they have changed over time. By analyzing the distribution of the most common feature code and plotting its time-based trends, we were able to gain a deeper understanding of resource behavior, which in the end helped in the optimization of the corresponding web page resources.

After creating our feature codes and applying the clustering algorithm, our dataset was ready for further analysis. The procedure began by identifying the most frequent feature codes within the cluster. 2.5 shows feature codes in one cluster and how many times they appear in that cluster. With this plot, now we are able to determine how often feature codes appeared within the clusters. The next part is required to plot the distribution of the most common feature code over various time intervals. This temporal analysis helped us to identify patterns, trends, and

changes in the behavior of the most common feature code. Figure 2.6, provides an analysis of the evolution of the resource’s features, as represented by the feature code.

As we delve deeper into cluster-level observations, the importance of this analysis becomes even more apparent. By recording the occurrences of the most frequent feature code within each cluster, we were able to identify the cluster when this code was the most common. This detailed temporal analysis uncovered complex temporal behaviors and potential anomalies associated with the most frequent feature code within a particular cluster. These insights provide a complete image of the behavior of the cluster’s resources, allowing us to make informed decisions regarding optimization strategies, content changes, and user experience improvements.

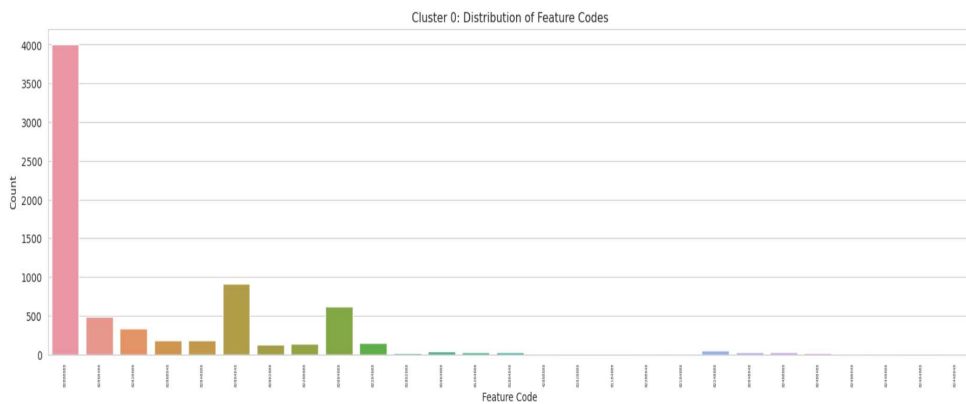


Figure 2.5: Cluster Feature Codes.

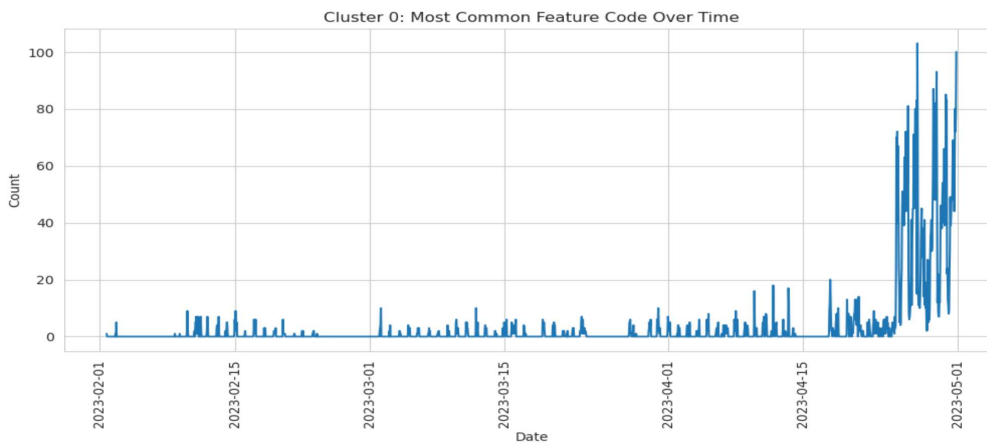


Figure 2.6: Distribution of Most Common Feature Code.

Outputs we got from the plots show we can easily check and control the feature codes over

time and we can translate our findings into practical knowledge for real-time problem detection. As each resource is associated with a unique feature code, any deviations from the expected distribution of feature codes can signal anomalies or performance problems. This allows us to easily detect problematic resources during specific time periods, pinpointing instances of slow-loading pages or other performance-related issues.

Anomalies or sudden changes in distribution patterns can serve as early indicators of emerging performance concerns. For instance, a notable shift in the frequency of a particular feature code could suggest a deterioration in the user experience, prompting proactive measures to address potential issues before they escalate. Consider a scenario where a specific feature code corresponds to resources that consistently load slowly over time. By closely monitoring the distribution of this feature code, we can quickly identify instances where its occurrence deviates from the norm. This immediate feedback empowers website administrators and developers to investigate the underlying causes and take action to optimize the resource.

#### 2.1.5 ANALYZING FEATURE CODES

The approach starts by applying the feature code algorithm to the 'anomaly page speed' data. This dataset includes multiple types of selected characteristics, such as page type, threshold quantile, total resources, number of slow resources, number of heavy resources, percentage of image duration, percentage of script duration, and percentage of fetch duration. The dataset is organized based on these fundamental features using clustering techniques, resulting in the assignment of unique feature codes to each individual case. This feature code assignment allows the informative analysis of the behavior of each case over time.

Important to the proposed algorithm is an extensive visualization approach that provides a precise representation of the periodic changes in the behavior of feature code. Using a particular amount of time of four days, the algorithm carefully captures the frequency distribution of feature codes by producing a graph. This graphical representation is an effective tool for understanding the complex patterns of feature code occurrences across the given time range.

The graph accurately shows the ups and downs of feature code frequencies, allowing the observer to identify trends, spikes, and variations in their occurrence over time. Specifically, the visualization highlights instances of unexpected changes regarding the counts of particular feature codes. These obvious differences from the established patterns serve as indicators, showing the presence of potential anomalies within the dataset.

The identification of these anomalies opens up possibilities for further investigation. Anoma-

lies in the behavior of feature code may indicate underlying problems that influence the corresponding web page resources. These variations may correspond to larger anomalies in the website's functionality or user interactions. By identifying these irregular occurrences, the algorithm provides an option to investigate potential correlations between feature code anomalies and website-wide anomalies.

Not only does visualization assist in the detection of anomalies, but it also functions as a start to deeper analyses. Once anomalies have been identified, they can be investigated further for potential connections with other metrics or events within the operational environment of the website. The methodology's ability to provide information that can be put into action is improved by the variety of components, which also makes it possible to develop accurate strategies for improving the performance of web pages.

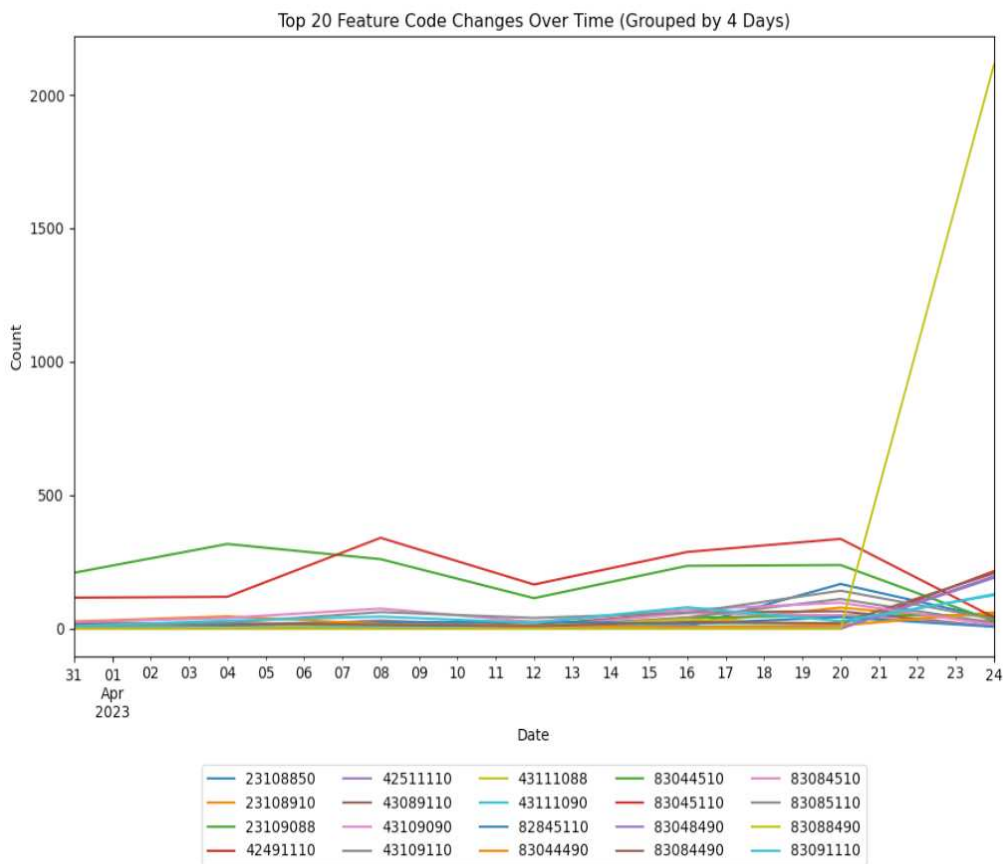


Figure 2.7: Top Feature Codes.

Upon close examination of the 2.7, we can see the behavior of the top 20 feature codes over

one month. The top 50 feature codes are decided depending on their occurrences over time. After analysing the graph a clear and sharp increase is noticeable between 20-24 April. This sudden and noticeable increase raises a red flag, indicating the presence of a significant anomaly or potential problem woven into the structure and functionality of the website. This observation led to the development of an algorithm that was developed with the intention of eliminating the complexities of this behavior in order to gain a more thorough understanding of it.

As a result, we developed an algorithm that establishes a precisely defined threshold that functions as a sensitive indicator for the emergence of particular feature codes. The basic idea relies on the constant monitoring of these feature codes, allowing us to identify immediately any significant deviations or anomalies in their occurrence patterns. The algorithm operates by applying the technique of feature code clustering to the dataset.

Once set up, the algorithm continuously compares the frequency of defined feature codes to the preset threshold. This monitoring enables it to detect even the last-minute or abrupt changes in the frequency of particular feature codes. Upon discovering a feature code that exceeds the predefined threshold, the algorithm immediately sets up an alert, similar to a digital flare signaling the possible emergence of a significant issue or anomaly. It is quite important for business owners to be able to observe a graph like this. Problems occur on the website environment very often and finding quick solutions is essential for companies. Some of these problems might be very obvious or easy to detect but there are lots of cases that are very hard to detect. These cases might be not very serious like preventing users from entering the website, but they can affect the performance of the website which can have a bigger effect in the future.

The system works to immediately inform stakeholders of any anomalous website performance landscape events. This early warning mechanism speeds up the investigation process significantly. It allows us to rapidly transfer our focus to the identified feature code and concentrate our analytical efforts on understanding the underlying causes and implications of its increased frequency. The process of feature code clustering, which is an integral part of the algorithm, improves its effectiveness. By employing this method, we are able to not only isolate the code for the notable feature but also gain an understanding of its contextual relationship with other features.



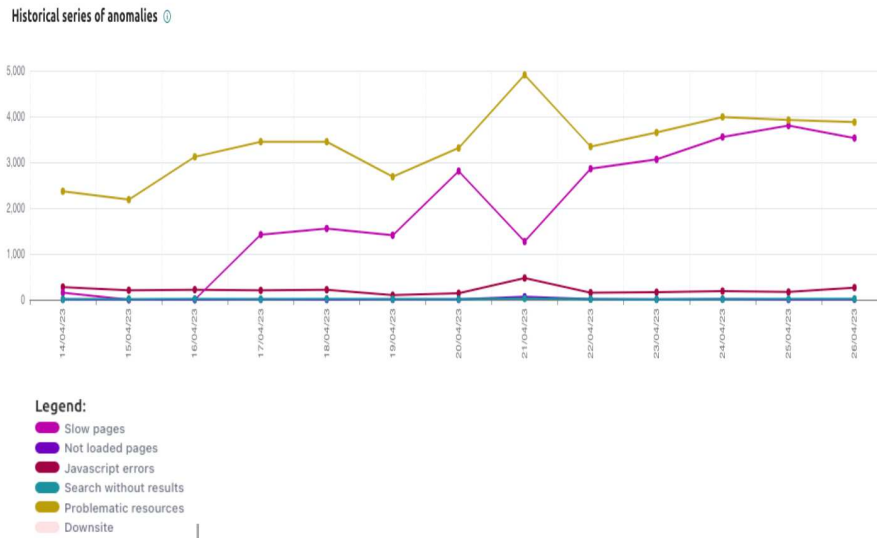


Figure 2.8: Backoffice Graph.

After detecting the feature codes with a sharp increase with the algorithm, now we need to confirm if the algorithm works and catches the problematic resources. 2.8 is from the company's back office. The company itself has an algorithm to count problematic resources, slow pages, and other types of pages. Their algorithm is not able to give detailed information about what are these resources but the algorithm we created will cover this big problem. Within the context of management oversight, the significance of the graph increases as it provides a detailed analysis of the temporal behaviors of numerous pages.

After the 20<sup>th</sup> of April, a significant change in the behavior of problematic resources was observed, which marked a turning point in this investigation. A clear increase can be observed in problematic resources after 20<sup>th</sup> April. Also, with the increase of problematic resources, the slow pages show the opposite behavior. It can be the harbinger of big problems in the website because if the slow pages decrease while problematic resources increase, that means the website might crash. If we look at the graph before 20<sup>th</sup> April both slow pages and problematic resources were increasing and showing similar behavior. After 20<sup>th</sup> April, both of them showed significant and opposite changes. That means to some point website had problems because of problematic resources which caused slow pages but after 20<sup>th</sup> websites were crushed and since the users were not able to use the website, it was normal to see slow pages decrease almost the bottom.

Turning to our algorithm performance, the algorithm can effectively catch unexpected resource changes in the selected time. The advantage here is, that with our algorithm, we are

able to give more information about this resource. This information will include a page of the resource, the type of the resource, and size of the resource, and more. In this way, the owner of the website can easily find the problematic resource and be able to solve it without affecting the customer experience on the website.

As we attempt to figure out the puzzle contained within this specific aspect of the code, a study of its features and implications is required. Assuming that it is unique, this feature code becomes the focus of an investigation to determine its underlying importance, function, and possible effect. By closely analyzing the feature code, we hope to unravel the complex pattern it creates. The attributes encoded in its binary structure are the key to understanding its unique function in the website dynamics. The process of emergence differences and afterward an increase are likely to help us to find a possible explanation for the observed change in problematic resources after April 20.

```
[('83108490', datetime.date(2023, 4, 24), 2113.0)]
```

**Figure 2.9:** Resource Details.

In figure 2.9, our algorithm captures and displays the distinctive feature code occurrences that are defined by unexpected and obvious changes in the image that is located above. With the selected frequency, in our case, it is 4 days, the algorithm takes the difference of counts of occur for each feature code within the selected frequency. After the algorithm compares this difference with the selected threshold and if the count of the feature code passes the threshold that means there is a problematic increase in the resource. Every single data point that makes up this graph is a feature code, and it is accompanied by the date that corresponds to it as well as the magnitude of the change that it shows. Specifically, the time period beginning on the 20<sup>th</sup> and ending on the 24<sup>th</sup> of April jumps out as being particularly notable. In this part, the aim is to get detailed information about this resource. After implementing this algorithm and getting the outputs of feature codes with sharp increase and ordering them, as we expected we got the feature code that increased after 20<sup>th</sup> April. With 2.9 we can see the feature code, date, and count. Within this time frame, there is a clear illustration of a sharp change in a certain feature code, which displays a remarkable rise of 2113 counts.

```

1 sharp_feature_codes = resources.loc[(resources['feature_code'] == sharply_increasing_feature_codes[0][0])]
2 sharp_feature_codes
Executed in 150ms, 31 May at 10:41:21

```

	page_type	percentage_img_duration	time_to_load	heavy_res_list
64501	1	3	117	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...
64503	1	3	860	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...
64508	1	3	102	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...
64510	1	3	262	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...
64512	1	3	654	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...
64522	1	3	33	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...
64525	1	3	401	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...
64538	1	3	40	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...
64541	1	3	844	{ "https://shop.parmalat.it/img/cms/Banner%20shopfitnessfitness...

Figure 2.10: Distribution of Most Common Feature Code.

The figure 2.10 displaying observed feature codes is provided so that complex observations can be obtained. The first observation shows an important theory, which states that the slow page in the issue is the home page. This can be determined from the fact that the page type is set to 1, which indicates that it is the home page. In addition, a careful investigation of the percentage picture duration attribute reveals that its values might fall anywhere within a range of 0 to 4. This effectively corresponds to the percentage interval that ranges from sixty to eighty-five of the resources contained inside the page. As mentioned in the previous section, we convert our numeric variables to categorical variables in order to more effectively implement the feature code algorithm. Because of that, the values we see in the table represent a range of values.

Upon a closer examination of the heavy resource list, an unexpected find becomes clear. The presence of banners on the home page is the reason for the development of so many resources in their heavy format. This addition seems to be the foundation of the problem that is occurring on the homepage.

A critical improvement that examines variances within the chosen frequency interval has been smoothly merged as part of a strategic step toward improving the existing algorithm. Particularly, when a frequency of four days is selected, the algorithm calculates differences in the occurrences of feature codes across each of these four-day segments, and as a result, it is an informative analysis of the produced differentials. The identification of significant shifts and evolving trends that are layered within each feature code is made possible with the help of this additional aspect. In addition to this, very important attention is given to feature codes that are making their first appearances. When a feature code appears for the very first time, beginning with a count of zero, it takes on a unique analytical alignment. This differentiation takes on importance, as it represents an emerging and unmatched occurrence that has the potential to expose significant abnormalities or serious situations.

The particular focus given to these developing feature codes provides the algorithm with

complete capacity. This allows the algorithm to perform an accurate and complete examination of potential problems associated with the domain of the website. The addition of these changes improves the algorithm, which is driven by its goal of improving its ability to detect anomalies and changes in the website's workings.

### 2.1.6 IMPLEMENTING ALGORITHM ON 'PAGE SPEED RESOURCE' DATASET

Further analysis was necessary in different datasets to see if we could get more inside information about the resource. The 'Page speed resource' dataset includes various information about the resource. In this case, the algorithm is deployed with a focus on the following features: 'page type', 'type', 'resource type', 'resource directory', 'decoded size', 'duration', 'resource name', 'external', and 'resource domain'. Each of these features contributes to a comprehensive analysis of the system's dynamics.

The resulting graph 2.11 serves as a visual representation of the interaction between these feature codes, which are tracked over a 12-hour time interval. The choice of a 12-hour frequency allows for an examination of the data, enabling the identification of trends, anomalies, or shifts occurring within relatively short time spans. While the dataset's size poses a challenge in obtaining a complete month's worth of data, the selected time window, spanning from April 20<sup>th</sup> to April 24<sup>th</sup>, provides a focused view of the system's behavior.

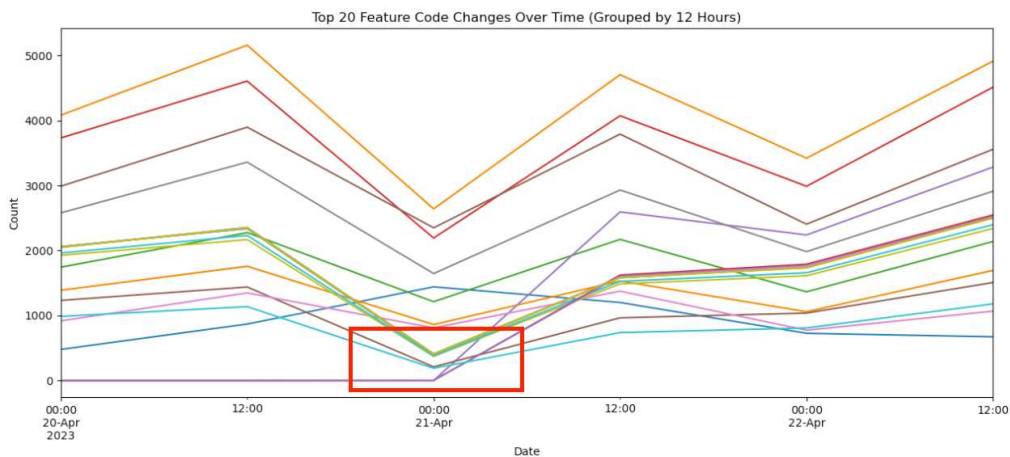


Figure 2.11: Feature Codes with 12 Hour Frequency.

By carefully examining the graph, observers are able to acquire significant insights into the patterns and fluctuations of the feature codes within the period of time that has been specified.

This understanding extends to having an idea of how the website or system operates, both in its typical state and in response to certain inputs. The frequency counts that were recorded at the 12-hour interval made it easier to spot notable occurrences as they occurred.

2.11, a perceptive analysis reveals dynamics among various feature codes. While many of these codes seem to exhibit similar trends, there are notable exceptions that caught our attention. Particularly on April 20<sup>th</sup>, there emerges a distinct pattern where two feature codes deviate significantly from the 'o'. In addition, within this time period, there is a clearly visible increase in the number of other feature codes that have been reported.

In addition, the fact that we ran our algorithm shows that performing in order was required whenever we came across a scenario like this. Because we are required to pay a heightened level of attention to the case in the event that new feature codes develop. As we have stated previously, other feature codes demonstrated the same behavior; however, these two newly developed feature codes have begun to demonstrate the same behavior and have passed other feature codes. Because these feature codes reflect the issue resource, we also need to take into consideration the fact that certain feature codes have the potential to have an effect on other feature codes. When one resource experiences issues, it is possible for such issues to spread to the other resources. When we open the website, for instance, we anticipate seeing the home page together with all of the graphics and other content. If one of your resources is not functioning properly, it can have a knock-on effect on the others, which could lead to a more significant issue. Here, we try to think of everything that could go wrong and keep an eye out for any potential issues that can arise.

In order to identify unusual occurrences in feature codes, the algorithm was modified to include a dynamic threshold mechanism. This strategy recognizes the variety of behaviors exhibited by feature codes and makes it possible to tune thresholds to the characteristics of each individual code. The goal is to precisely find feature codes that indicate strange behavior or sudden occurrences, comparable to the anomalies. The dynamic threshold performs the role of a sharp observer by differentiating between the typical operation of the resources and the existence of probable anomalies that call for more detailed investigation. This adaptable framework has potential because, it allows website managers to actively monitor a wide range of feature codes, allowing them to immediately discover and investigate unanticipated changes. Because of this increased attention, reaction times have been shortened and investigations have been more narrowly focused, which has eventually improved the website's stability and performance.

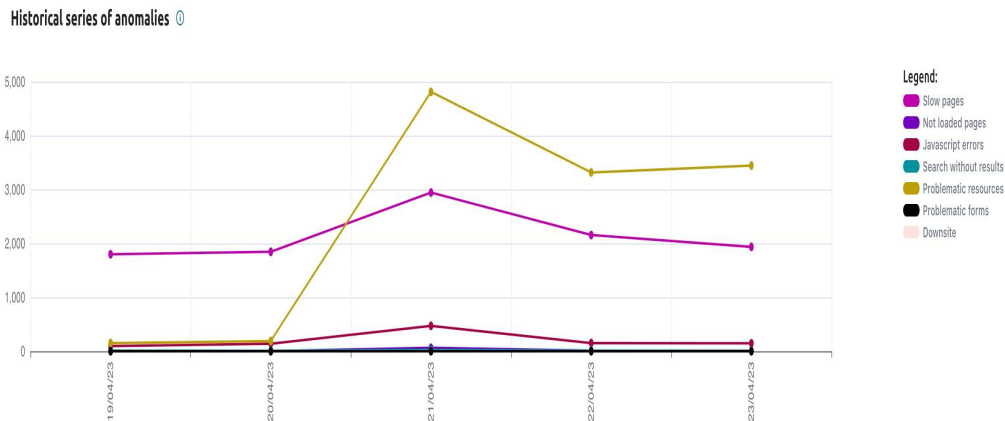


Figure 2.12: Backoffice Graph.

2.12 shows analysis and the thorough examination of problematic resources in the back office. As a result of this examination, a striking correlation has been uncovered, which highlights a key relationship between the reported spike in feature codes on April 20<sup>th</sup> and a corresponding change in the behavior of problematic resources. This connection is essential for understanding why the observed spike occurred.

This correlation is far from being a coincidence; rather, it serves as physical evidence of the effectiveness of the algorithm that was deployed. The ability of the algorithm to rapidly identify and record changes in the behavior of website visitors stands out as a crucial factor in the development of this interesting relationship. The algorithm to flag these changes, which can be seen by the notable increase in feature codes, resonates with the parallel change in the performance of problematic resources.

The fact that the algorithm is synchronously aligned with the dynamics of the real world shows the effectiveness of the tool as an anomaly detection method. Its reliability is demonstrated by the fact that it is able to provide insights that correlate with the behavior that is observed in the back office. Because of its expertise in anomaly detection, it is able to identify problems in a timely and accurate manner, which contributes to the website's stability and reliability. The company's ability to quickly identify problematic resources and architectural abnormalities on the website is essential to the maintenance of the company's competitive position in the market. This practical approach to website monitoring and anomaly management not only finds anomalies in a timely manner but also keeps a high level of accuracy throughout the process.

The detailed figures presented 2.13, 2.14 provides an extensive overview of the feature codes

that were discovered in the analysis. Our earlier conclusions are strengthened and expanded upon by the discovery, upon closer examination of insights that were hidden within these specifics.

page_type	type	resource_type	resource_directory	decoded_size	duration	resource_name	
453421	home	link	css	/themes/classic-rocket/assets/cache	6	175.100006	theme-ae1a54574.css
453422	home	link	css	/themes/classic-rocket/assets/cache	6	175.100006	theme-ae1a54574.css
453494	home	link	css	/themes/classic-rocket/assets/cache	6	163.899994	theme-ae1a54574.css
453496	home	link	css	/themes/classic-rocket/assets/cache	6	163.899994	theme-ae1a54574.css
453529	home	link	css	/themes/classic-rocket/assets/cache	6	635.299988	theme-ae1a54574.css
...	...	...	...	...	...	...	...
884885	home	link	css	/themes/classic-rocket/assets/cache	6	139.100006	theme-ae1a54574.css
885049	home	link	css	/themes/classic-rocket/assets/cache	6	164.000000	theme-ae1a54574.css
885054	home	link	css	/themes/classic-rocket/assets/cache	6	164.000000	theme-ae1a54574.css
885062	home	link	css	/themes/classic-rocket/assets/cache	6	164.000000	theme-ae1a54574.css

Figure 2.13: Problematic Resource 1.

page_type	type	resource_type	resource_directory	decoded_size	duration	resource_name	
396267	home	img	png	//img/cms/	8	110.000000	Banner%20shopfitnessfitness%20(2).png
396271	home	img	png	//img/cms/	8	110.000000	Banner%20shopfitnessfitness%20(2).png
473227	home	img	png	//img/cms/	8	8275.000000	Banner%20shopfitnessfitness%20(2).png
473241	home	img	png	//img/cms/	8	8275.000000	Banner%20shopfitnessfitness%20(2).png
473341	home	img	png	//img/cms/	8	154.100006	Banner%20shopfitnessfitness%20(2).png
...	...	...	...	...	...	...	...
884833	home	img	png	//img/cms/	8	1716.300049	Banner%20shopfitnessfitness%20(2).png
884848	home	img	png	//img/cms/	8	3117.800049	Banner%20shopfitnessfitness%20(2).png
884877	home	img	png	//img/cms/	8	3117.800049	Banner%20shopfitnessfitness%20(2).png
885005	home	img	png	//img/cms/	8	2776.100098	Banner%20shopfitnessfitness%20(2).png

Figure 2.14: Problematic Resource 2.

As one explores deeper into the complexities, a discovery is made; one of the feature codes has a strong connection with the "img" resource type. This conclusion fits in perfectly with our original hypothesis, which centered on the fact that the company had strategically placed a new banner on its homepage. The fact that there has been an obvious rise in the number of instances of this resource type, which was revealed by the analysis of the feature code, is a clear approval of this theory. The stronger connection between this change and the observed changes in website dynamics is demonstrated by the rising number of "img" resource-type activations in the feature code cluster.

Additionally, an effective rapport can be seen between the "css" resource type and an additional feature code. This correlation, in turn, directs attention to an important aspect: the strong presence of CSS components or stylesheets that are inherently tied to the main page. Due to the focus that has been placed on this particular type of resource, the likelihood of specific design or layout adjustments, or even style improvements, being the driving cause behind the increased occurrences has been significantly increased. This gives weight to the hypothesis that the homepage has been given various CSS complexities, which in turn gives rise to the repeated instances of this feature code.

The information that was obtained from this analysis provided website administrators with a practical road map to follow in order to effectively address and maximize the resources that were being investigated. Since the data that we are working with comes from the real world, we are able to verify this issue with the company and share the results with them. After sharing the information, the responses we received to our questions confirmed that the algorithm was performing as intended and identifying the genuine issues. The issue was caused when the company added a new banner to its homepage, which resulted in a major problem with the website, a site crash, and users being unable to access the website. Mostly due to the fact that this resource also affects other resources. After removing the resource, the problem was fixed; however, determining the cause of the issue at the time might be time-consuming, but by using the algorithm that we created, we are able to detect the issue early on and reveal exactly where the issue is located.

### 2.1.7 BOUNCE-CAMPAIGN ALGORITHM

Web analytics programs help organizations monitor visitor behavior and gather data on website engagement. The Bounce Rate is a performance indicator that measures user engagement with a web page. A low Bounce Rate indicates more users are engaged, exploring content, and clicking deeper into the site. By increasing user engagement, the Bounce Rate decreases, allowing organizations to better understand and retain visitors [13]. The data from the company provides information into the complex world of user interactions on the website, where the idea of "bounces" is front and center. In this discussion, the term "bounce" refers to the situation in which a user's journey on a website starts and stops inside the bounds of a single page, without any further engagement on the user's part. Even though this kind of behavior can be caused by a number of different circumstances, such as a slow-loading page or strange events on the page itself, there is one aspect in particular that calls for careful study, and that is the influence of campaigns.

Campaigns have their own distinct effect on user behavior, which can occasionally take the form of bounces that require particular focus. In the event that consumers arrive at a page by clicking on a link associated with a campaign, the reason for their quick exit could be due to a lack of interest in the content of the campaign itself. The interplay between a campaign's goals and the preferences of its users is complex and calls for a separate analytical perspective to be applied.

The fact that these bounces are connected to the dynamics of the campaign does not necessar-



ily imply that there are issues in the design or performance of the website. Instead, they reflect the complex details of human preferences and behaviors, providing insights into the dense web of interaction that is created by digital interactions. An algorithm that is designed to improve the bounce measure is taking shape as part of a strategic solution that is being developed to ensure the reliability of the analysis. This algorithm expands its scope beyond the simple idea of a bounce by using additional measures such as the overall number of pages visited and the total number of interactions. This combination of indicators makes it possible to conduct an evaluation that is more complete in terms of user engagement and behavior, producing a more complete picture of the user's digital journey.

The results of this algorithmic improvement are really significant. In the analysis of bounce rates, an unusual clarity develops, accurately separating circumstances in which bounces are not of any relevance to website performance or anomalies. The strength of the algorithm resides in its capacity to extract the substance of user behavior, excluding instances in which bounce rates are more sensitive to campaign-driven anomalies than to fundamental website flaws.

It's critical to pay close attention to the more complicated aspects of data analysis, and this is especially true when it comes to assigning values to a column labeled "bounce". In order to successfully navigate this complex landscape, it is important to give careful thought to many scenarios, each of which paves the way for a more complex comprehension of how users behave. At the core of this analysis is a judgment that can only take one of two possible forms: either "True" or "False" should be written in the "bounce" column. The difficulties of this determination become apparent in a number of different contexts.

#### HOME PAGE ENTRIES AND EXITS

In the situation where both the entry page and the exit page match the home page, the default assumption is that a bounce has occurred. The situation, however, becomes more complicated when users navigate through the website in stages, beginning and ending with the home page. In order to address this issue, an identifiable statistic has emerged, and it is referred to as the "total page viewed." If this metric is higher than two, it suggests that there was persistent involvement that went beyond simple entry and exit, which disproves the bounce assumption.

#### DIFFERENT PAGES FOR ENTRY AND EXIT

When the entry and exit pages are different from the home page and do not overlap a more complex evaluation is required. This is the case when there is no overlap between the entry

and exit pages. A dual assessment is required for this purpose, taking into account both the "total page viewed" and the "total interaction" variables. This combined examination enables a full assessment of user interaction, which facilitates a more accurate classification of a case as a bounce True or otherwise False, which is made easier by the fact that the case can be classified as either of them.

Following the presentation of this improved process, the following step includes putting it into practice in some way. The "bounce" column has been brought up to date, which allows the feature code algorithm to take the spotlight. The scope of the investigation has been restricted even further by focusing on the data that was gathered between January 1 and March 1. As the 2.15 reveals crucial patterns and variations in the behavior of feature code. The graph in this representation highlights a moment that is of the highest significance; specifically, the undeniable spike of two feature codes. These feature codes include information about the user's device, the user's browser, if there is a campaign existing in the session, and the bounce information. In that way, we can do an analysis of which campaigns cause the most bounce.

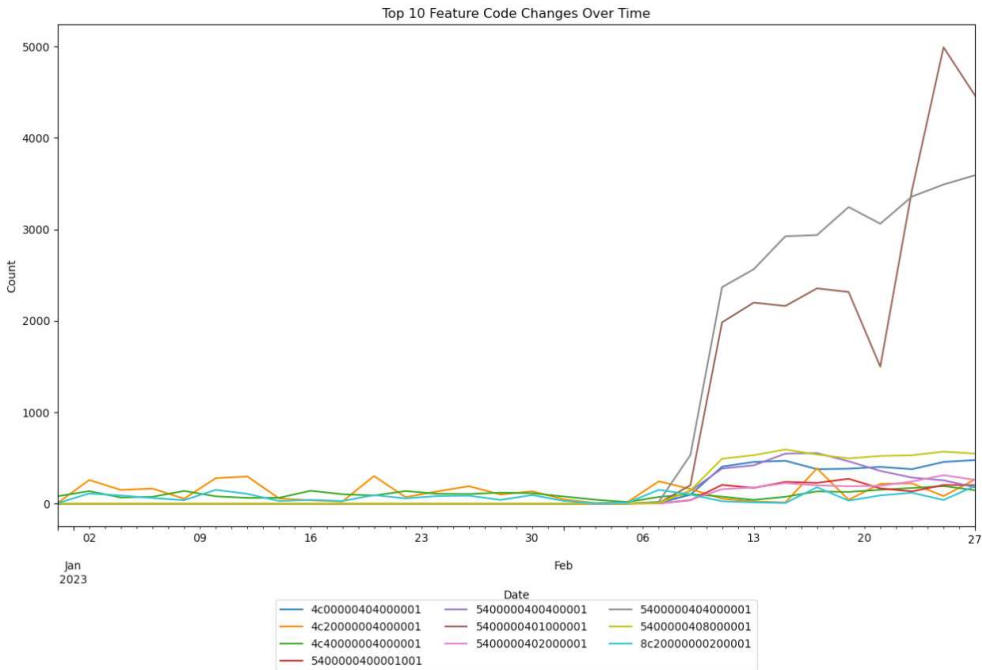


Figure 2.15: Campaigns Usage.

```

def calculate_campaign_effect_score(data, time_frequency):

    grouped_data = data.groupby(pd.Grouper(key='date', freq=time_frequency))

    campaign_effect_scores = []
    dates = []

    for time_period, group in grouped_data:
        num_sessions = len(group)
        num_true_bounces = group['bounce'].sum()
        num_false_bounces = num_sessions - num_true_bounces

        if num_sessions == 0:
            campaign_effect_percentage ==0
        else:
            campaign_effect_percentage = (num_true_bounces - num_false_bounces)
/ num_sessions * 100

        campaign_effect_scores.append(campaign_effect_percentage)
        dates.append(time_period)

    return dates, campaign_effect_scores

```

**Figure 2.16:** Campaign Effect Score Code.

The upper function 2.16, is carefully built to compute the campaign effect score within the context of a given dataset while taking into account a specified time frequency for the purpose of data grouping. The accuracy of the function's operation is dependent on two critical parameters, primarily the data and the time-frequency. The first one refers to the dataset that is being analyzed, while the other provides the interval that should be used for categorizing the temporal data.

The function described here aims to assess the impact of marketing campaigns on website user behavior. It starts by generating arrays of campaign-effect scores and dates. It examines time segments iteratively, calculating session counts (visits) within each. A critical calculation differentiates true bounces (quick exits) from false ones. To avoid division errors, the function handles cases with zero session counts by resetting the campaign effect percentage to zero. Finally, the campaign effect percentage is calculated by subtracting false bounces from true bounces, dividing by total sessions, and converting to a percentage. This percentage quantifies the campaign's impact on user bounce rates and serves as a useful measure of its impact across time segments.

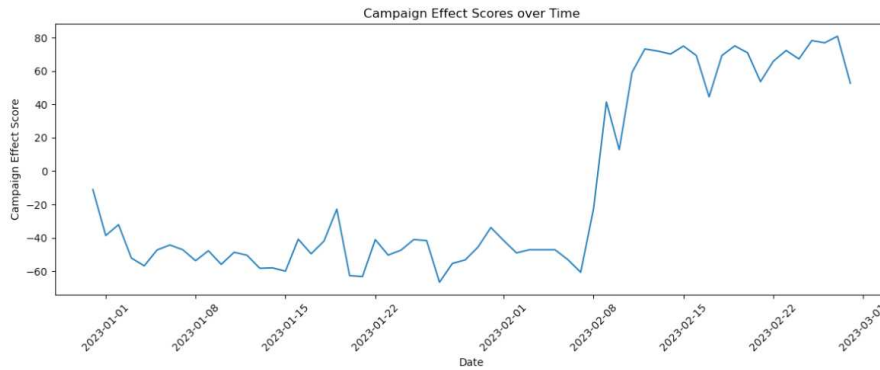


Figure 2.17: Campaign Effect Score.

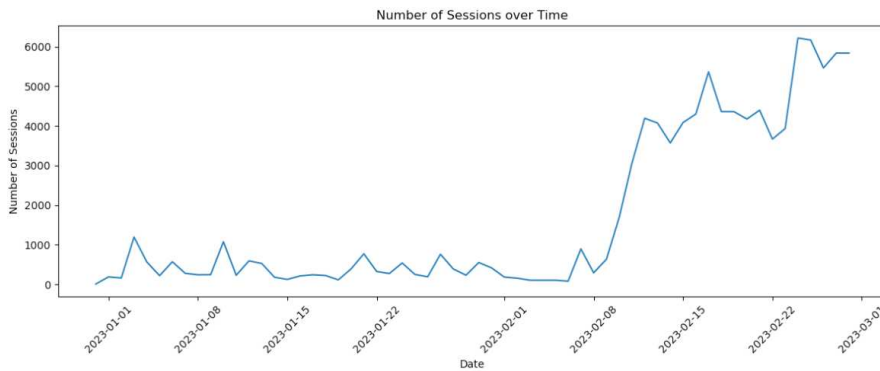


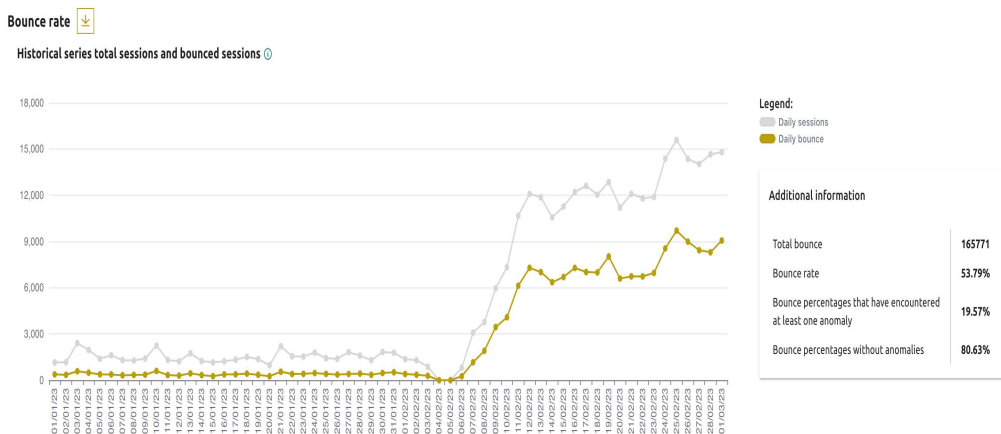
Figure 2.18: Distribution of Most Common Feature Code.

A graph of user sessions can be found in 2.18, which is presented alongside the score for the effectiveness of the campaign 2.17. 2.18 display the total number of sessions that occurred over a specified time period and were set up to occur at a particular frequency. This graph functions as a visual companion that offers an understanding of the user engagement dynamics, and it does so in combination with the campaign effect scores.

When these connected graphs are examined in great detail and compared to one another, symmetry can be noticed in the patterns that emerge. When the development of the campaign effect scores is reflected in the graph that keeps track of the total number of sessions, a visually practical plot is created. Prior to the important date of February 7<sup>th</sup>, both plots exhibit consistent trends that are defined by barely noticeable undulations. The current day represents a period of stability since it marks a moment where the fluctuating levels of user activity and the effectiveness of campaigns are connected and show the same behavior. This phase can be identified by the ups and downs in user activity as well as the effectiveness of the campaign.

On the other hand, the plot that takes place after February 7<sup>th</sup> marks a turning point in the flow of events, as an obvious rise is noted in both the campaign effect score and the session count. This rise implies that people are engaging with the content more frequently after this date. This should sound the alert since it suggests that there is a major connection between the effectiveness of the campaign and the level of user involvement.

The fact that campaign effect scores and session counts have both been on the rise at the same time is undeniable evidence of the power of smart marketing. These advertisements not only serve to attract individuals but also help in holding their interest, which ultimately results in longer interactions and lower bounce rates. In addition to this, it is necessary that we highlight the significance of the campaign effect score when it comes to identifying the issue with the website. Bounce rate is one of the factors that is used in the calculation of the campaign effect score. We have already discussed the algorithm that is used. Therefore, if there are not a lot of sessions on the website and the campaign impact score is high, we can assume that the bounces are related to the campaigns. On the other hand, if the campaign effect score is low and there are a lot of sessions on the website, this may be an important warning.



**Figure 2.19:** Back-office Bounce Rates.

The closing graph 2.19, which is prominently shown, reveals an intricate pattern of daily sessions and daily bounce rates, therefore carefully charting the ups and downs of user interactions inside the company’s digital environment. When you compare this graph to the one that showed the campaign effect score before, a connection becomes more apparent. A visible parallelism emerges, connecting the two graphical stories and highlighting an essential aspect that

the algorithm that underlies the campaign effect score analysis maintains consistent correctness and reliability.

The accuracy of the algorithm is strongly demonstrated by the fact that this alignment is reliable. The convincing validation that the algorithm is able to mirror real-world dynamics is provided by the similarity between the graphs of the back office and the graphs of the campaign effect scores. It highlights the algorithm's basic ability to understand and resemble the delicate interaction of user engagement and bounce rates, encapsulating them within the calculated campaign effect scores.

Using the algorithms can transform it into an efficient decision-making tool. When administrators are equipped with this analytical knowledge, they are able to begin a process of up-to-date development of strategies, campaign initiative leadership, and user experience influence.

# 3

## Recommender System

### 3.1 EVALUATING OPTIMAL FUNNELS WITH USER CLUSTERING AND PURCHASE PROPENSITY EVALUATION

An understanding of user behavior has emerged as an important asset, essential for businesses that want to succeed in the constantly evolving field of e-commerce. This understanding is essential for a number of reasons. Because of the intense level of competition in this industry, being able to accurately forecast the likelihood of a user making a purchase has taken on a greater level of importance. As an outcome of this, companies have a responsibility to arrange their marketing campaigns with the modified precision necessary to effectively engage customers in ways that are compatible with the customers' desires and preferences. The use of complex algorithms that have been included in the foundation of data analytics has emerged as a powerful tool for dealing with this difficult environment.[14]

The Recommender System filters personalized information based on user preferences from large data sets. To provide effective recommendations, accurate user interest and taste construction are necessary, and accurate feedback on the recommendations is crucial.[15] The practice of studying diverse user interactions is at the center of this research. Businesses stand to get access to a variety of insights by studying and decoding the interactions between their products and customers. When users are grouped together according to the patterns of their behaviors, a map of user preferences is revealed. This provides a perspective from which to identify possi-

ble patterns of purchase. This time-consuming categorization serves as the foundation around which customized interactions are built, with every interaction being crafted to attract consumers and increase conversion rates. Conversions, in which a user becomes a customer, are the focus of online marketing. Conversion can vary from business to business, with e-commerce conversions involving purchases at the end of a session. The term "conversion" originates from the concept of a conversion funnel, which shows the path users take before converting [14]. A value event is any occurrence that moves a user down the value event. A user watching a film on Netflix, for example, is a value event that begins broad and then narrows. A large number of site visitors visit, add items to their wish lists, share products, and some purchase items, and in the end funnel narrows.

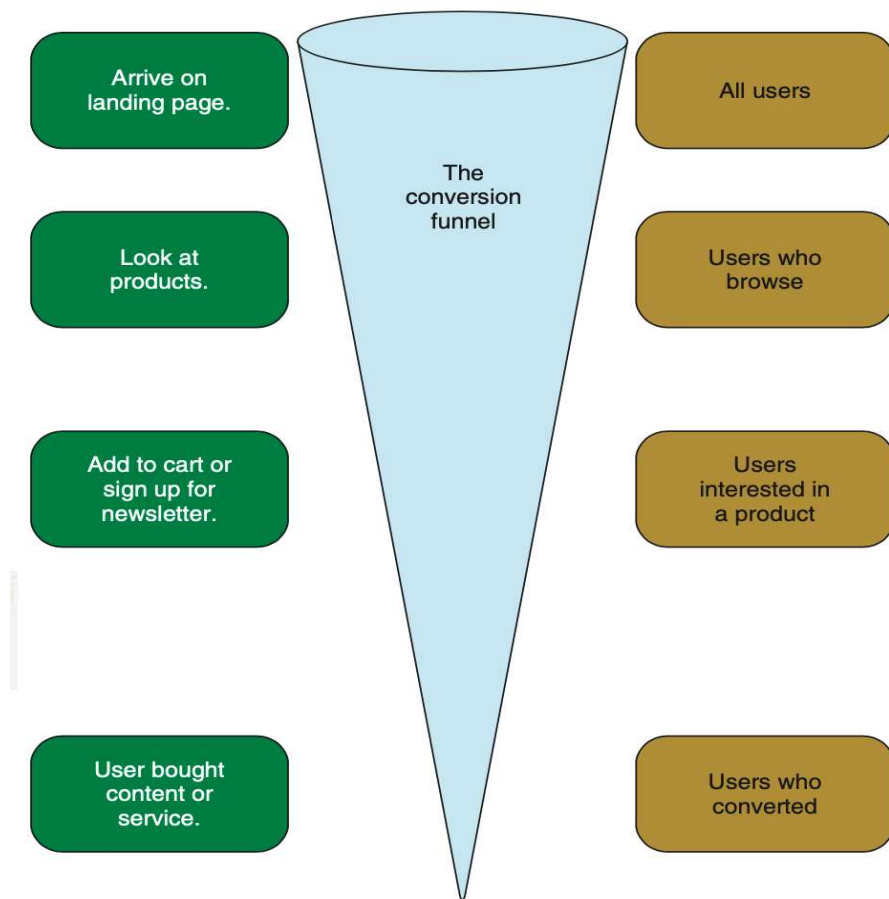


Figure 3.1: Conversion Funnel.

This dataset is a repository of information that spans a spectrum that includes user demographics, product properties, interaction metrics, and session details. This rich data is what



feeds the analytical ability of the algorithm. It is a good source to help us to understand user behavior and predict their tendency to make purchases. The algorithm's demanding path begins with careful data preprocessing and feature engineering. This is just the beginning of the algorithm. This step of preparation establishes the framework for following analyses, ensuring that the dataset is refined to a clean form and ready for investigation at this point in the process. A project that goes beyond simple statistics and provides a complete understanding of user behavior can be achieved through the application of an algorithm that, after the basis is set up, executes exploratory data analysis. The insights that were obtained during this phase expose trends and make evident the hidden clusters that are the result of user attributes.

The algorithm follows a path through machine learning, which is a setting in which predictions can be discovered. It navigates the complexities of predictive modeling effectively, taking into account both the big picture of consumer behavior and the details of product preferences.

It goes well beyond the realm of simply manipulating data and transforms into an indicator of direction for companies who are looking to establish deeper connections with the people they serve. It gives the capacity to understand user intentions, personalize experiences, and design marketing tactics that resonate harmoniously with user expectations, all of which can be accomplished through the usage of this. In the end, this algorithm acts not only as a tool but more as a visionary companion, guiding organizations toward a future that will be characterized by increased consumer engagement and better conversions.

2.19 shows the process of data flow from evidence to recommendations involves collecting information about website usage, gaining an understanding of user preferences, and applying this information to train the recommender system for the purposes of making recommendations.

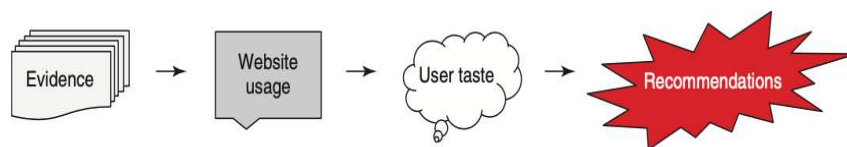


Figure 3.2: Data Flow for Recommendations.

In order for the Recommender System to recommend items that are useful to a specific user, it is often necessary to comprehend the user and his or her interactions with the system. Usually, these interactions appear as both explicit and implicit input from customers, which provides

the key indicators for modeling users' preferences for items and the essential information for personalizing recommendations.[16] We collect information from five distinct sources as the foundation of this project. The resulting data will illustrate how people interact with the website overall. We are using information taken from the "cart-analysis-product," "cart-evolution," "stats-cart-analysis-general," and "stats-user-behavior-general" parts of a database that has been properly arranged. The items, their prices, and the quantities are all included in the information contained in this dataset. There is information on the user's cart, including the total price of the cart, the number of products currently in the user's cart, and the type of cart the user has, which indicates whether or not the user has purchased the product in question.

The information that we have about user activity is of the highest priority. We have information on their interactions, including the type of device they used, the amount of time they spent, the pages they visited, and the number of pages they visited. Each of these pieces of data focuses on a different aspect of the overall online experience, such as examining shopping carts, the characteristics of products, the evolution of shopping carts through time, and the activities of users. When we put it together, these different aspects create a comprehensive picture for us of how users interact with the website.

Since we are merging these different datasets on a common column which is `session_id`. The `session_id` is the primary key for our datasets and shows different users under different `session_id` numbers. When we merged these datasets, the data frame we created brought us various and valuable information from different datasets for each session. Being able to see these values under one data frame is going to be very effective for our analysis. The combination of these several types of data helps us understand the story more. It enables us to see a great deal more detail and comprehend how various elements are related to one another. We are able to discover things such as when users perform actions that are similar to one another or when they change their behavior together.

In consideration of the fact that we offer a wide variety of different characteristics, it was important, before moving on to other activities, to evaluate the significance of each feature. An example feature importance graph 3.4 has been constructed after doing an analysis of the relevance of the features using a random forest classifier. At the top of the list is the target variable known as "cart type," which is of the greatest significance because of its direct connection to the outcome that was anticipated. In the case of the other factors, they do have an impact on the forecasts but their significance is significantly less prominent in comparison to that of the primary objective variable.

In order to further improve the effectiveness of the model and produce insights that are more

robust, an important quest has emerged, and that quest is to identify a single attribute that is of major importance. Such a characteristic, when provided with considerable importance, has the potential to significantly improve the performance of the model, providing a source of insights that may be used to drive future research.

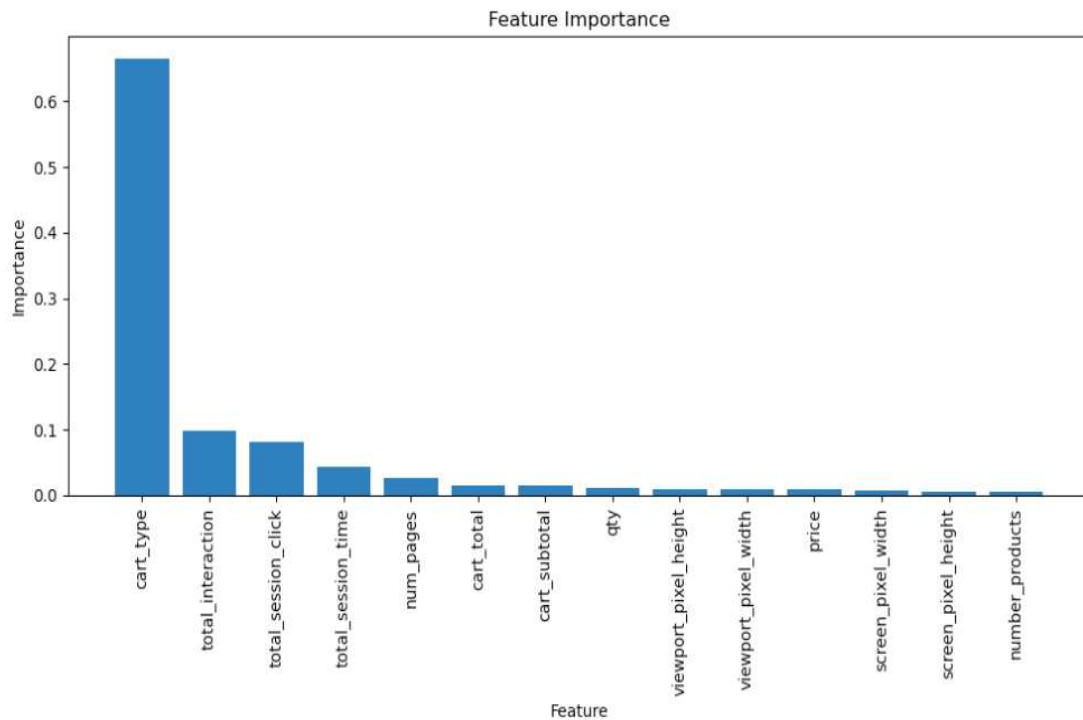


Figure 3.3: Feature Importance for Recommender System.

The addition of a single high-impact feature can be game-changing, despite the fact that the current features work together to put predictive power on the system. This feature not only improves the forecast accuracy but also provides the direction future investigations should take. The pursuit of this unique feature needs close exposure to the problem domain, an insightful comprehension of the dataset's complex context, and a sensitivity to the underlying connections that can be found within the data. This goal may include digging deeper into the dataset, inventing new features, or coordinating a combination of existing qualities to a newer, robust competitor for predictive power.

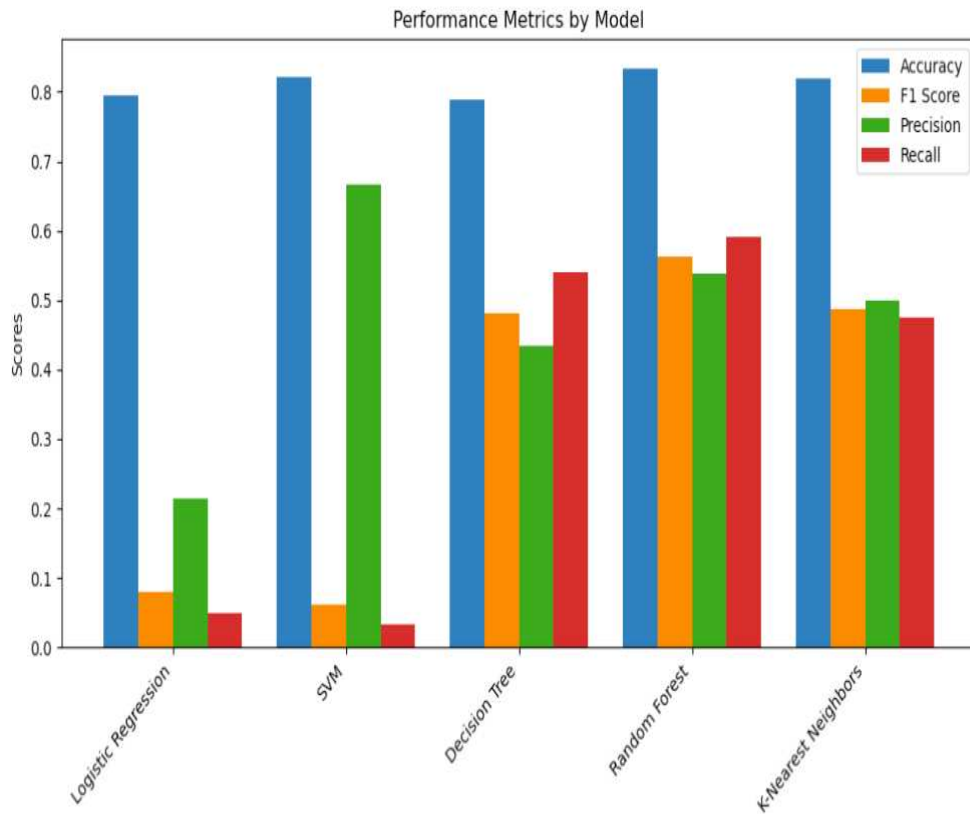


Figure 3.4: First Models Performances.

Model	Accuracy	F1 Score	Precision	Recall
Logistic Regression	0.7953	0.0800	0.2143	0.0492
SVM	0.8220	0.0625	0.6667	0.0328
Decision Tree	0.7893	0.4818	0.4342	0.5410
Random Forest	0.8338	0.5625	0.5373	0.5902
K-Nearest Neighbors	0.8190	0.4874	0.5000	0.4754

Table 3.1: Performance Metrics of Different Models

The evaluation of the predictive models provides details into their performance as well as the possibility of accurately identifying users who have a high likelihood of making a purchase. An examination of the most important metrics draws attention to the advantages and disadvantages of each approach. Here, our target variable is 'cart type' which type is boolean. 1 is the corresponding user makes the purchase and 0 means the user leaves the site without any

purchase. As our predictor features, we use the first seven features in the ranking of feature importance.

In this experiment, we conduct a comparative analysis of five machine learning models. K-Nearest Neighbors, Logistic Regression, Support Vector Machine, Decision Tree, and Random Forest. This analysis intends to evaluate the performance of these models on a specific dataset and show their respective strengths and abilities. Each model was selected based on its distinct qualities and suitability for various scenarios. With its interpretability and effectiveness in linear relationships, Logistic Regression is ideally suited for situations where understanding the impact of features is essential[17]. SVM, on the other hand, is adaptable and excels when there is a distinct class separation or when the data is high dimensional[18]. Decision Trees, which are adept at capturing complex relationships and non-linear boundaries, excel in scenarios where feature importance is of the highest priority[19]. Random Forest, a collection of Decision Trees, provides robust performance across datasets, allowing for improved generalization through aggregation. Due to its ease of use and capacity to handle non-linear boundaries. For the machine learning evaluation, metrics such as F1 Score, Precision, and Recall are essential for measuring the performance of classification models. These metrics provide information into a model's ability to achieve a balance between minimizing false positives and ensuring it identifies all positive instances, which makes them especially valuable when working with unbalanced datasets. F1 Score, a harmonic mean of precision and recall, effectively balances false positives and false negatives, which makes it especially useful for unbalanced datasets. Precision, the ratio of true positive predictions to all positive predictions, indicates the model's ability to minimize false positives. Recall, which quantifies the ratio of true positive predictions to all actual positive cases, demonstrates the model's ability to identify all positive instances. Collectively, these metrics provide an understanding of each model's predictive capability.[20]

The Logistic Regression model has reasonable accuracy but a low F1 Score of 0.0492, indicating it struggles with a trade-off between true positives and false negatives. The Support Vector Machine model, with a precision of 66.67, is accurate in predicting user purchases but has a low recall of 0.0328, indicating it might miss a significant portion of actual purchases. In contrast, the Decision Tree model's F1 Score is 0.4818, suggesting a better balance between precision and recall. It has a recall of 0.5410, successfully capturing many actual purchases. The Random Forest model performs well with a high accuracy of 0.8338 and an F1 Score of 0.5625, showing a balanced trade-off between a precision of 0.5373 and a recall of 0.5902. The KNN model achieves an F1 Score of 0.4874, indicating a balanced precision of 0.5000 and recall of 0.4754 and effectively predicting purchases while maintaining fair coverage.

Among the models that were tested, the Random Forest model was shown to have the highest accuracy and balanced precision-recall trade-off. As a result, it appears to be a particularly good contender for identifying users who have a high propensity for making purchases. In addition, it is essential to recognize that these measures may not be adequate for performing an accurate forecast of the user's tendency to make purchases. It's possible that the present features used in the models don't capture the entire complexity of user behavior and preferences, both of which are essential elements in determining how well a model can forecast consumer spending. It is important to explore the production of new, relevant characteristics that can provide a more comprehensive representation of user interactions, preferences, and intentions in order to improve the predictive capabilities of the models. This will allow the models to better respond to new information.

### 3.2 CREATING RATINGS WITH USER INTERACTION

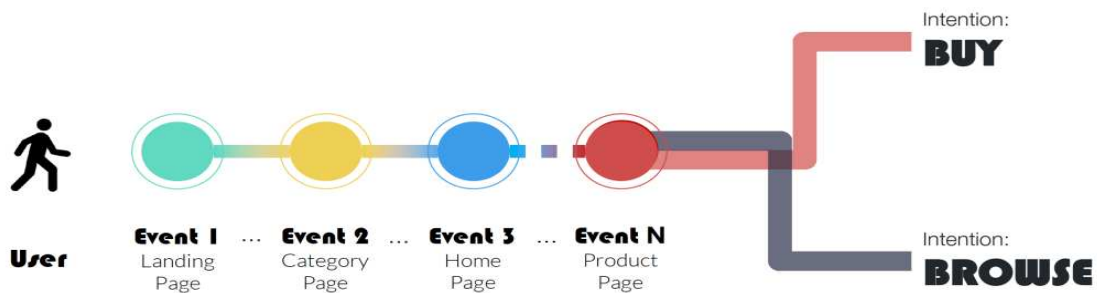


Figure 3.5: Creating User Ratings.

In a Recommender System, user ratings are valuable data to have and give lots of information about user preferences. We can see examples of rating on Amazon or Glassdoor sites where people rate products or their workplaces. Ratings combine important aspects which are user, content, and user's sentiment to the content[14]. In the lack of clear consumer ratings for individual products, we need to find a way to create something else with the variables we have. This approach requires the development of a dynamic rating structure that is founded on the digital footprints of individual users. This method takes advantage of the complex network of user behavior on the website, creating a grading system that is based on the activities and interactions of users.

The idea of propensity scores, which reflect the statistical possibility of users carrying out particular activities or displaying particular behaviors, is essential to this method. We use this score to evaluate user involvement and measure their platform interaction intensity. In order to bring this idea into reality, we need to assign weights to various user actions. These user actions will each be given a distinct meaning that is symbolic of their value. For example, when a user navigates to a page that is dedicated to a single product, whether it's a comprehensive review of its details or the straightforward act of adding an item to the shopping basket, the weight of this action resonates more powerfully, overtaking other, more general interactions such as casual browsing or moving across sites. This weight determines the user's purpose by providing a window into the user's level of interest or engagement with the particular product in question.

Making this involves more than just page views and mouse clicks alone. It includes a list of interactions, such as session duration, engagement with specific features, and a click pattern, and each of them contributes to the depth of understanding of user intention. A dynamic rating system is formed as a result of this complex web of user behaviors, which includes many different aspects and facets. It creates a complex mosaic, which acts as an alternative for explicit ratings and is created by the fluctuation of user behavior.

In addition, taking into account the correlation with cart type may result in a significant improvement in the performance of the models. Cart type may reveal insights into the shopping behavior of users, such as whether they are browsing casually or actively considering purchases. One example of this would be determining if a user is browsing casually or actively considering purchases. The models have the ability to identify complex patterns that contribute to more accurate predictions if the cart type variable is included as a feature in the analysis. Incorporating characteristics that are irrelevant or biased could lead to overfitting or incorrect outcomes. However, the selection of additional features should be motivated by a deep understanding of user behavior and domain knowledge.

### 3.2.1 CREATING THE VARIABLE

page_type_list	num_pages	unique_page_type_list
{category, checkout, category}	3	{category, checkout}
{home, home, product, category, category, category, category, ...}	28	{category, home, product, checkout}
{home, product, home, product, home, product, product, product, home, cat...}	82	{category, home, product, checkout}
{category, category, home, checkout, product, product, category}	7	{category, home, product, checkout}
{product, product, checkout, product, category, product, checkout, prod...}	13	{category, product, checkout}
{home, category}	2	{category, home}
{home, category, product, product, product, product, product, category, ...}	55	{checkout_success, category, home, product, checkout}
{product, product}	2	{product}
{product, product, product, checkout, product, home}	6	{home, product, checkout}

Figure 3.6: User Page Paths.

In the provided 3.6, the main focus is on giving different levels of importance to the actions users take while they're visiting various pages during their time on the website. Finding out how useful or significant each action is in terms of determining whether or not a user is likely to make a purchase is the objective of this research.

We determine a particular list that we refer to as the "unique-page-type-list" in order to ensure that we do not count the same page more than once and to maintain order in the process. This list contains all of the unique kinds of pages that a user visits. We know how essential each activity is, we can determine it by using a unique dictionary that we call "value-mapping." Each variety of pages, such as "checkout" or "checkout-success" is assigned a value that reflects the significance of the page type. Because of these values, we are able to determine which behaviors are most likely to result in a purchase. We have 6 types of different pages within the website and each one of them has its own value of importance. For example, the highest value is checkout-success, because that means the user made the purchase. The algorithm accepts a list of page kinds as input. After that, it examines our "value-mapping" dictionary to see if the page type is present there. If this is the case, the value is added to a running total that we refer to as the "Page Interaction Weight".

After that, we calculate the "Total Weight" by first dividing the total number of pages by the square root of that number. The explanation for this is that users may revisit pages more than once. When users go shopping on the website, for instance, they navigate to a number of different pages that display various products. After this, we updated the overall weight by integrating some key features, and those features will include the total session time, the total interaction, and the total session click. The sum of the user's active and inactive time on the website is referred to as the total session time. After multiplying the total number of interactions and clicks



during the session, we split the total session time because there are some instances in which the user might walk away from the computer and return after two hours. The means by which we may understand the scope of this engagement, and as a result, we have divided the overall interaction with the user.

$$\text{Total Weight} = \frac{\text{Page Interaction Weight}}{\sqrt{\text{Number of Pages}}} \quad (3.1)$$

$$\text{Total Weight} = \frac{\text{Page Interaction Weight}}{\sqrt{\text{Number of Pages}}} + \frac{\text{Total Session Time}}{\text{Total Interaction} \times \text{Total Session Click}} \quad (3.2)$$

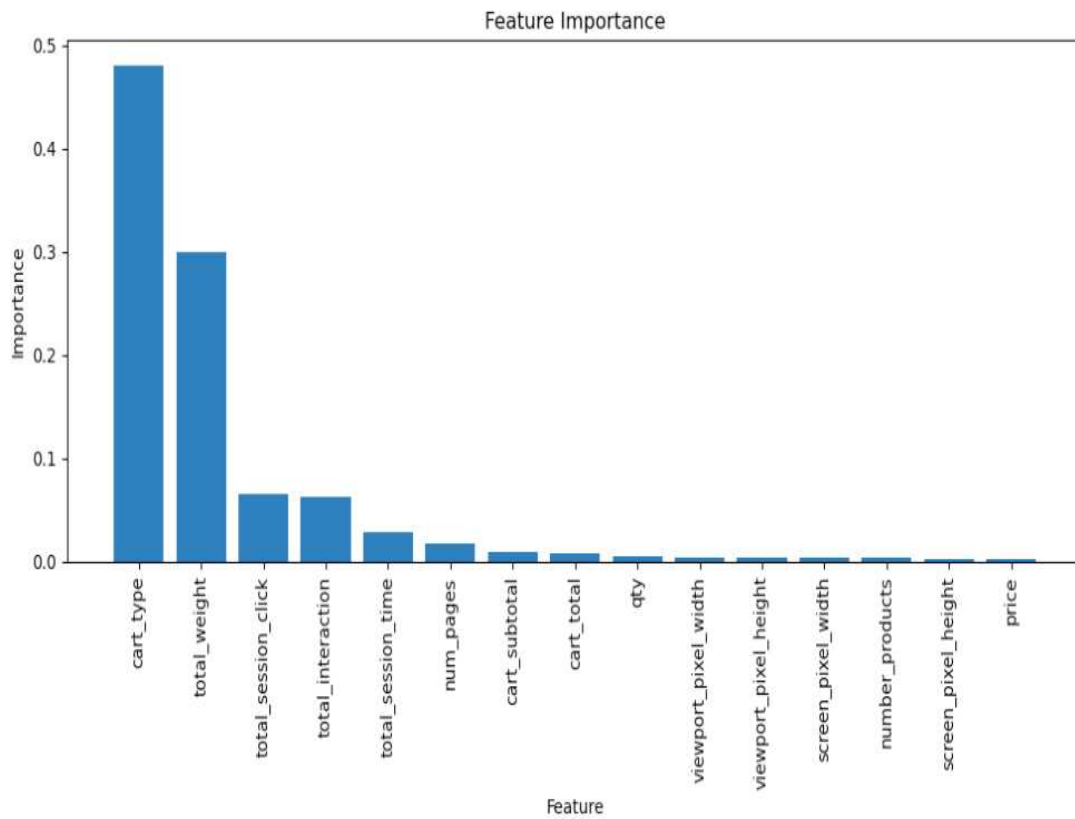


Figure 3.7: Feature Importance After Total Weight.

After implementing the method, we now have a unique value that is referred to as the 'total weight' for each user. This figure serves as a kind of summary of all the significant actions that a user performs while they are on the website. These actions include viewing products, adding them to the shopping cart, and completing the checkout process. This 'total weight' is similar to a score that demonstrates the extent to which the user is actively engaging with the website.

Something interesting pops out at us now whenever we examine a graph that illustrates the ways in which various factors are connected to one another. According to the 3.7, there is a significant correlation between this 'total-weight' score and whether or not consumers end up making a purchase. This value tells us whether or not someone is likely to become a customer of ours in the future.

This value can be understood better if we think of it in the following way; whenever a user performs significant things while they are on the website, such as spending more time looking at products, coming near to making a purchase, or completing processes such as checking out, their "total-weight" increases. As a result of this, the likelihood of them making a purchase increases in parallel with the 'total weight' of the recommendation. Therefore, when we talk about users who make purchases, we are talking about users who are truly interested and involved in the activity. They are not simply skimming over the content; rather, they are investing time and effort in looking at the site and interacting with it.

This 'total weight' is more than simply a figure on a chart. It's an effective tool that we may employ in a variety of different ways. It can be put to use in models that forecast the future or make suggestions about various matters. For instance, we may use it to determine the likelihood that a person will purchase a certain item. Or, if we want to provide better product suggestions to a customer based on what they might be interested in purchasing, we can use this 'total-weight' metric to do so. Because of this aspect, we are in a position to make decisions that are more intelligent. When we have a better understanding of who is most likely to make a purchase, we are in a better position to devise strategies for targeted marketing and design experiences that are tailored to the preferences and habits of potential customers. In the end, everything revolves around making things better for users and contributing to the success of businesses.

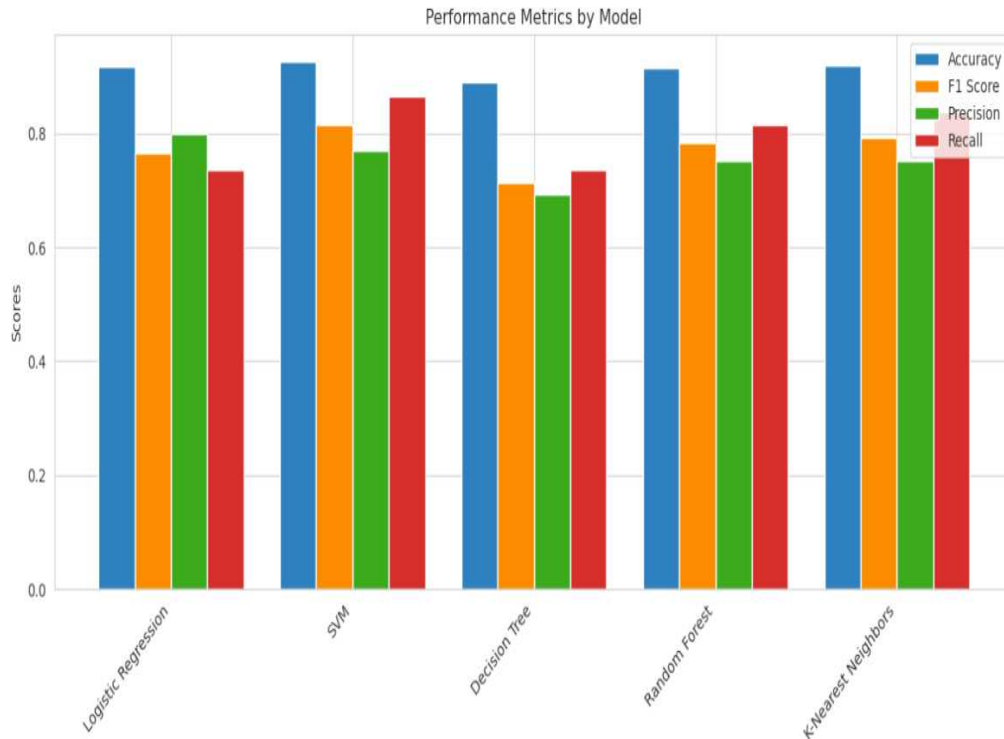


Figure 3.8: Model Performances After Total Weight.

Model	Accuracy	F1 Score	Precision	Recall
Logistic Regression	0.9274	0.8125	0.7839	0.8432
SVM	0.9384	0.8184	0.7767	0.8649
Decision Tree	0.9073	0.7592	0.7360	0.7838
Random Forest	0.9315	0.8191	0.8063	0.8324
K-Nearest Neighbors	0.9234	0.7989	0.7824	0.8162

Table 3.2: Performance Metrics for Different Models After Total Weigh

The inclusion of the 'total weight' variable has helped attempts to improve predictive models and gain a better understanding of user behavior. This new statistic, which captures the essence of user involvement across our website, has improved our models. As a result, we now have a better ability to identify potential customers.

By adding the 'total weight' feature, an important development has changed both the accuracy of our predictive models and the practicality they provide. We recalculated the perfor-

mance of our models using this complex metric system and found that they had significantly improved across all aspects.

The Logistic Regression model, which had an F1 Score of 0.0800 prior to the addition of 'total weight,' increased to 0.8125 with the variable. This improvement is not only a visual one; rather, it indicates an expanded capability to strike a balance between true positives and false negatives, illustrating a model that not only forecasts accurately but also captures potential customers in a more efficient manner. In the same way, the F1 Score of the SVM model jumped from 0.0625 to 0.8184 while simultaneously achieving a new level of predictive power. This was complemented by notable improvements in both precision and recall.

The Decision Tree model improved from a balanced F1 Score of 0.4818 to 0.7592, demonstrating that the 'total weight' feature added depth to its prediction skills. This progress was demonstrated by the model's F1 Score, which increased significantly. The Random Forest model, on the other hand, demonstrates the transforming power of the 'total weight' variable. Its F1 Score increased from 0.5625 to 0.8191, highlighting a newly discovered precision-recall balance that is critical for collecting potential consumers while avoiding false negatives. The K-Nearest Neighbors model embraced its balanced nature even more, increasing its F1 Score from 0.4874 to 0.7989. This result demonstrates the effect of the 'total weight' addition once more.

The introduction of the 'total weight' variable marked the beginning of an advanced period of predictive modeling capability. It has been demonstrated to be an essential component in strengthening the precision-recall trade-offs and enhancing the overall predictive power of our models. Because we are now equipped with the 'total weight' insight, we are in a position to identify users who have a higher level of involvement. This will accelerate us toward making decisions that are more informed, developing targeted marketing tactics, and providing users with better experiences.

### 3.2.2 UNVEILING HIDDEN PATTERNS: CLUSTERING ANALYSIS WITH K-MEANS

Clustering analysis is a powerful method that we have adopted in our continuous search for a deeper understanding of user behavior and to search for hidden insights within our dataset. In both of these activities, we are looking for previously unseen insights. We look for patterns by applying the K-Means algorithm to the data that we have carefully preprocessed.

We sorted our dataset into two main categories, namely category columns and numerical columns so that we could create the framework for our clustering. Our categories columns include data such as the user agent device and the user agent browser info, which together provide useful knowledge about user platforms and surfing preferences. Our numerical columns, which contain indicators such as cart subtotal, number of pages, and total session clicks, simultaneously capture both the qualitative and quantitative essence of user engagement and interactions.

For classification, regression, and clustering algorithms, data in numeric format usually provides better outcomes. However, many problems in machine learning contain categorical or nominal features in addition to numeric features. One-hot Encoding is a popular method for converting categorical features to numerical features in traditional data mining tasks[21]. The encoding of our categorical data with the OHE was an essential step in the progression of the process. This step included the smooth transformation of text characteristics into a format that the clustering algorithm could understand. We were able to obtain relevant information from categorical variables. At the same time, we normalized our numerical features by using the StandardScaler. This allowed us to eliminate unjustified scale influence on the clustering results while also ensuring a fair distribution of importance across the various metrics.

After carefully organizing and preprocessing our data, we apply the K-Means method. Aim was the to put people who are similar together based on the characteristics and actions they share by configuring the algorithm to generate five clusters. The results of our clustering study reveal a complicated collection of user clusters, each of which can be identified by a distinct set of interactions, preferences, and behaviors. As we continue to explore the outlines of these clusters, we are beginning to create an accurate depiction of how users interact with our platform. Information about user preferences, browsing behaviors, and the level of engagement intensity come to the forefront, which enables us to design strategies and experiences to appeal to these various user categories.

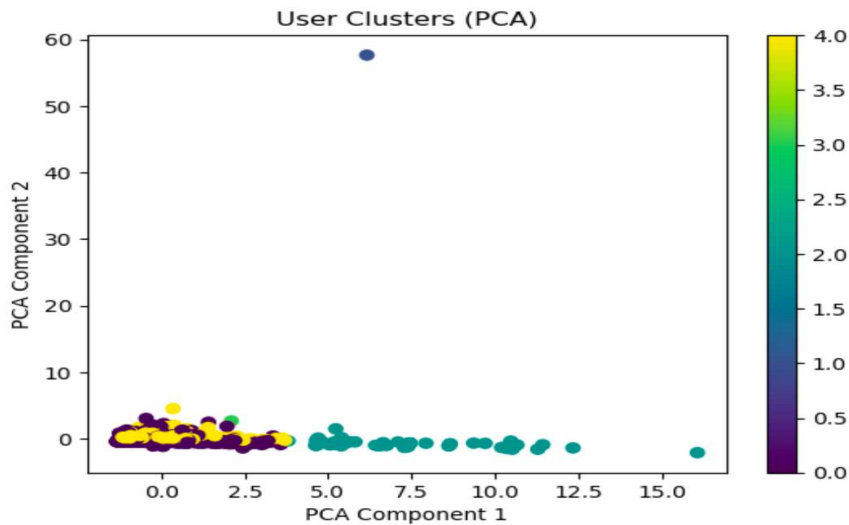


Figure 3.9: User Cluster.

The visualization of the scatter plots 3.9 for clustering, which is done using both numerical and categorical variables, shows the complex patterns that are contained within the data. The usage of categorical factors for the purpose of clustering, such as the browsers and devices used by users, did produce some useful insights; but, the clustering’s overall effectiveness was limited. The significant separation of clusters was not as pronounced as expected, maybe because the chosen number of attributes did not have adequate discriminating ability.

The clustering result displayed unsatisfactory performance in the original scatter plot. The scatter plot included numerical factors such as the number of pages, total interactions, cart total, total session time, total session clicks, product price, and quantity. Because the clusters did not have distinct boundaries and were overlapping, it was difficult to figure out any meaningful patterns from the data.

However, a significant shift occurred after the 'total weight' information was included for the first time. The scatter plot 3.10 displayed clearer and more distinguishable clusters after the addition of this feature. The 'total weight' successfully improved the discriminatory abilities of the clustering algorithm, which resulted in clusters that are not only more distinct from one another but also more meaningful in the distinctions between them.

K-means clustering analysis using scatter plots, two important metrics are used to evaluate the quality of the clusters formed which are inertia and silhouette score. Inertia reflects the tightness of data points around their cluster centroids, with lower values indicating defined clusters on scatter plots. Silhouette score considers both the connection within clusters and

separation from neighboring clusters, generating scores between -1 and 1. Higher silhouette scores correspond to better-defined and separated clusters on scatter plots. These metrics aid in visually interpreting cluster quality and assessing how effectively the K-means algorithm groups data points in scatter plot visualizations[22]. In order to get an accurate measure of how well the clustering worked, we computed two important metrics: the inertia and the silhouette score. The preliminary clustering, which was carried out without the use of the 'total weight,' produced a first cluster with an Inertia value of 7952.82 and a Silhouette Score of 0.2490. Even though they were suggestive, these metrics highlighted the urgent need for improvements in cluster separation.

The 'total weight' feature resulted in a visible improvement in the overall performance of the cluster once it was put into action. The Silhouette Score jumped to 0.7360, suggesting that the clusters are now significantly more distinct from one another. Concurrently, the inertia experienced a dramatic decrease, reaching a value of 262.799, further proving the efficiency of the feature in producing distinct and solid clusters.

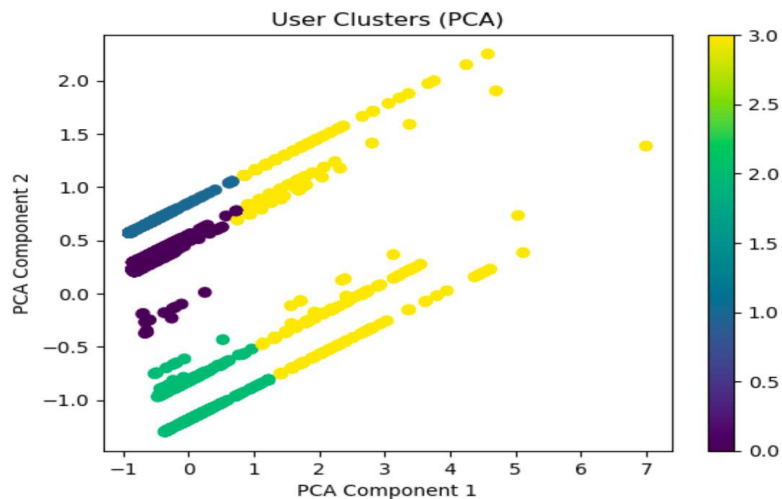


Figure 3.10: User Cluster After Total Weight.

### 3.2.3 CALCULATING THE CUSTOMER PROPENSITY OF PURCHASE

This is a type of statistical method that is frequently used in a variety of industries, including marketing and studies involving observations. Given specific attributes or qualities of individuals or entities, we are able to have a better understanding of the possibility or probability of a certain event occurring with the assistance of propensity scores.

In this particular situation, the event that catches our attention is whether or not a user will end up buying a specific product. This is the question that drives our focus. The percentage value that is assigned to a user's propensity score reflects the likelihood that the user will make this purchase given the user's characteristics. This score is assigned to each user. Having determined the total weight of the user in the prior chapter, we will now make use of that information to compute the user's propensity score.

We are making use of a technique of machine learning known as a Support Vector Classifier in order to compute these scores. This approach is frequently applied to a variety of classification problems. It does this by locating a decision boundary, also known as a hyperplane, that most effectively divides the various classes contained inside our dataset. The decision function is the most important part of the classification process used by the SVC. This method generates an output score based on a set of input features—in our case, features relating to a user—that it gets as input. The expected class is determined based on both the pattern and size of this score. The operation of the decision function involves measuring the distance between a data point and the decision border. When the distance between the two points increases, the model's level of confidence in its own prediction also increases. We can imagine a line being drawn through the middle of a scatter plot to separate two different groupings of dots. The more distant a point is from this line, the more positive we are that it belongs to one of these categories.[23][24][25]

The decision function of the SVC is responsible for assigning a score to each individual user. This score is intended to show where the user stands in relation to the decision boundary. In a way, it can be understood as a measurement of the model's confidence in the user's intention to buy the product. After that, we turn the output of the decision function into a score that represents its likelihood. This score is determined by applying a mathematical operation known as the logistic function, which is also referred to as the sigmoid function. This transformation is essential for binary classification tasks, as it allows the output to be interpreted as probabilities or likelihoods of a particular class. The characteristic S-shaped curve of the sigmoid function ensures that larger decision values are mapped closer to 1 (higher probability) and smaller decision values are mapped closer to 0 (lower probability). This transformation aligns the model's output with a probabilistic interpretation, making it suitable for tasks such as estimating the probability of user interactions in a recommendation system based on the users' past behavior and engagement.[26]



The formula we are using,

$$\text{Propensity Score} = \frac{1}{1 + e^{-\text{decision\_function\_output}}}$$

ensures that the score is always between 0 and 1. A score closer to 1 indicates a higher likelihood of the user purchasing the product, while a score closer to 0 suggests a lower likelihood.

### 3.2.4 PRODUCT CATEGORY

If only one feature is included in the calculation of the customer's propensity score, the results will not be effective or accurate. Because in our most recent algorithm, we are generating the propensity score of the client by just using the total weight, it is quite difficult to determine the customer's true behavior from this score. Consider, for example, the kind of products in which they might also be interested. There are multiple products that fall into the same category or are somehow connected to one another. Imagine a customer who buys a TV and also has an interest in the audio system or DVD player that the company also sells. Because of this, when we are determining the user propensity score for various items, we need to make sure that we take this into consideration as well. Consequently, given the lack of unique customer details included in my dataset, it was important to make improvements to the clustering of users. What we have so far is the entire weight of the client, which demonstrates the degree to which the consumer is integrated intensely into the Website. The total weight is an important characteristic of our algorithm; however, in order to specify the user clustering, we needed to introduce additional variables for our particular scenario. In our particular situation, we work with both customer and product data simultaneously. In our situation, the ability to group users according to the products in which they are most interested would be an extremely helpful feature to have.



Home > All > Telephony & photography > Mobile phone > Smartphones  
> APPLE - iPhone 14 Pro Max 256GB - Operating system: Apple iOS-Screen size

Figure 3.11: Category path of the product.

The reality that our dataset does not have this kind of information was the source of the issue that we were having in the current case. In order to find a solution to this issue, it was required to conduct an exhaustive search for the dataset. After performing the study, important information was discovered. There is a URL column in the dataset, and the value for that column is a link to the product page. By clicking on this link, we will be able to see the website page for the product page and get a sense of the design of the site 3.11. There is information regarding the product's journey through the various categories on the website. On this particular website, there are three different levels of category organization. We are able to view the category path that the product takes due to the reference figure category. In the next part of this report, we will refer to this level of organization as the general category, the category, and the subcategory. Following the application of this reasoning, the dataset would have gained useful details not only about the products but also about the customers. These three features will be available to us for use in clustering data and accurately determining the user's propensity score.

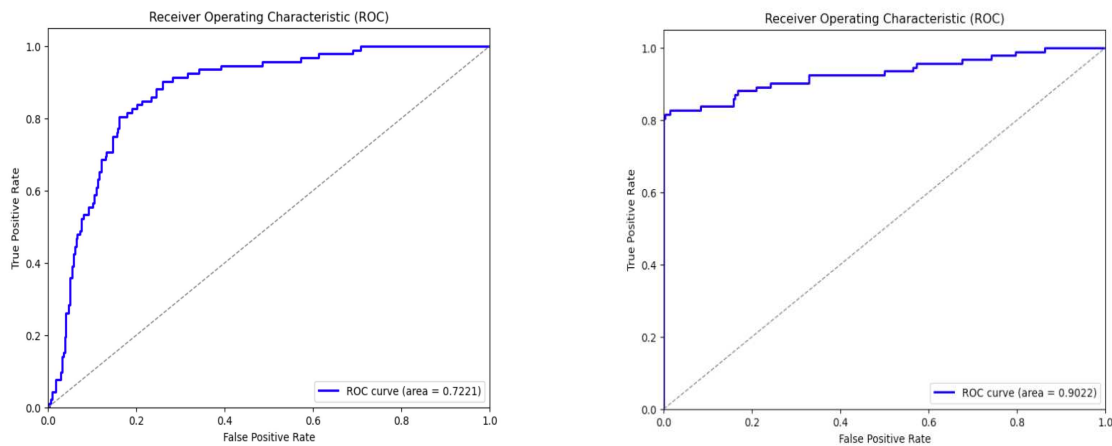
### 3.2.5 IMPLICATION OF PRODUCT CATEGORY

In this method, we have developed a recommendation system that makes use of clustering, propensity scores, and other techniques from the area of machine learning in order to give customers customized product recommendations based on the interactions and preferences they have shown. After adding the product category information to our project, finding the best way to add it to the equation was very important. Since we have three levels of category, we can use these layers to define more specific clusters and we can increase the accuracy of calculating the propensity score of the user. Our three-level product category is a general category, a category, and a subcategory.3.11, our general category here is telephone and photography, the category is the mobile phone, and finally, the smartphone is the subcategory. Under the subcategories, we can find the specific product we are looking for and since every product has this path, we can give weights to these levels. When we want to calculate the user propensity score for specific products, we can use the user's purchase history. If their current purchase has a similar path they are likely to buy the other products under the same category also. In our algorithm, we gave more weight to the subcategory and after category and gave a small weight to the general category. Because the subcategory shows more specific information and this is what we need to use more.

Using machine learning, we were able to gain insight into the predictive potential of the program. The OHE method was utilized again in order to encode categorical features, while

the StandardScaler function that is included in SciKit-Learn was utilized in order to standardize numerical features[23]. The utilization of these methods guarantees that the data is organized in the most effective manner possible for the training of the model. After getting the propensity values from using the decision function using our categorical and numerical values, we can add the category values to the propensity score when we are calculating the propensity score for a specific product or for general categories.

In order to measure the efficacy of our algorithm, we also carried out an extensive set of tests. We determined the ROC AUC score, also known as the Receiver Operating Characteristic Area Under the Curve, by doing cross-validation with a 5-fold split. This score is a reliable metric for binary classification tasks. The model's capacity to accurately and reliably predict user interactions was demonstrated by the obtained mean ROC AUC value of 0.9250, with a minimal standard deviation of 0.0209. The robustness of the algorithm was further validated by independent testing performed on a different test set, which produced a test set ROC AUC value of 0.9022. The results of these validation tests demonstrated that the algorithm is capable of producing accurate predictions.[27][28]



(a) ROC Curve Without Total Weight.

(b) ROC Curve With Total Weight.

Figure 3.12: ROC Curve.

## CLUSTERING ANALYSIS

Categorization is made for users according to the similarities in their interactions by applying K-means clustering to the preprocessed data and analyzing the results. This clustering is an important step that must be taken before we can adjust our recommendations to particular user clusters and improve the level of personalization that our solution provides.

The propensity scoring algorithm is the foundation of our entire system of recommendations. Propensity scores are calculated to determine the possibility of a user interacting with a product. These scores were first calculated by employing the decision function of an SVM model. In addition, in order to improve the level of personalization, we implemented modifications to the propensity score.

These modifications are the effect of giving serious consideration to the significance of sub-categories, categories, and general categories in forecasting the behavior of users. We gave each of these considerations a weight, and then we calculated changes to represent how important each one is to the overall process of making recommendations. Our algorithm had been programmed to generate recommendations that fit closely with user preferences once the clustering and propensity score changes were implemented. Users within each cluster were given recommendations that connected directly with their interests, which resulted in an increased possibility of engagement with the platform. The order of priority of the items was determined by making changes to the propensity scores. This was done to ensure that the recommendations not only take into account the preferences of the users but also take into account the unique interaction patterns of each user.

#### CLUSTER-SPECIFIC THRESHOLDS AND RECOMMENDATIONS

The implementation of cluster-specific thresholds represents the most significant development we have made to our algorithm. To determine the threshold for each cluster's average propensity score, we compute it. Products that have average propensity ratings that are higher than this threshold are evaluated to be strong potential candidates for recommendation within that cluster. In order to get a better understanding of how this process works, we develop an insightful bar chart 3.13 that compares the typical propensity scores of different goods to the cluster threshold. This visualization offers a simple and straightforward explanation of the products that are most closely linked with the interests of each group. 3.13 we can see the product codes and propensity score for cluster 2. A clear observation that we can make is that all of the product codes start with 2. Which means 'washing and cleaning' as a general category. We can see the code changes because there are categories and subcategories under this general category. Our algorithm is able to produce recommendations for users that actually connect with them since it is equipped with cluster-specific thresholds and changes to the propensity score. We ensure that consumers receive suggestions that are not only relevant but also consistent with their preferences by taking into account the distinct behaviors that are associated with each cluster.





# 4

## Detecting Error with User Behaviour

The approach of analyzing user click behavior has been used in a variety of fields to address specific challenges and improve understanding. The first paper, in the field of web design and user experience optimization, introduces a method based on the hidden semi-Markov model for accurately identifying user click patterns. This approach is useful for web operators and designers who are dealing with the complexity of modern websites, as it helps them understand user interactions and customize their designs accordingly. This methodology could be applied to other user interaction scenarios, such as mobile apps or software interfaces, where determining user engagement patterns is critical.[29]

The second paper in the field of cognitive research and human-computer interaction uses an innovative physiological perspective to analyze web user behavior. The study attempts to uncover the relationship between physiological cues and user click intentions by combining physiological data such as pupil dilation and EEG responses with web log data. While the preliminary results show the potential of such an approach, they also highlight the importance of high-quality measurement. This methodology may inspire similar studies in areas other than web usage, such as assessing user responses to various types of media content or interactive environments.[30]

The third paper presents an extensive exploration of user click behavior models in the context of information retrieval and search engine optimization. To predict user clicks on search engine result pages, these models take into account a variety of factors such as document titles, URLs, and snippets, as well as user session history. The paper contributes to improving search result

presentation and user satisfaction by evaluating the effectiveness of different click models in predicting actual user behavior. The adaptability of this approach could go beyond search engines, finding applications in e-commerce platforms, or content recommendation systems to improve user engagement and conversion rates.[31]

Understanding user behavior has emerged as an important component for organizations that want to improve their online platforms and maximize income in the fast-evolving world of e-commerce, where every click and interaction holds the potential for revenue. The digital marketplaces of today are so much more than simple transactional platforms; rather, they have developed into complicated ecosystems that provide customers with personalized experiences, material that is constantly updated, and navigation that is simple. With the information, doing deep studies into the patterns of user activity is now an absolute requirement for locating and fixing problems that, if not addressed, could result in lost revenue.

The term "user behavior analysis" refers to the process of methodically collecting, measuring, and then interpreting data regarding how users interact with a website. Businesses can acquire insights into how visitors engage with the web pages they provide. This allows businesses to identify problems, optimize processes, and eventually improve the overall user experience. E-commerce businesses are able to identify essential errors that may be limiting their income potential by seeing deviations, anomalies, and patterns within this behavioral data. These errors may be preventing businesses from reaching their full revenue potential.

The analysis of user behavior as a means of determining the degree of severity of an error is an important line of research within this field of study. The importance of recognizing user annoyance and frustration as significant indicators of website difficulties that may contribute to revenue loss is being highlighted by an increasing number of research that highlights the need to understand this occurrence. Examples of "rage clicking," in which users repeatedly click on a single item out of annoyance, or "dead clicks," in which interactions generate no reaction, have been related to underlying problems such as broken links, unresponsive buttons, or confused interfaces.

Studies carried out in the field of research have shown that analyzing user behavior can be helpful in recognizing problems like these. According to the findings of a study, a sizeable portion of the shopping carts that were not completed was due to the dissatisfaction of the user, which was brought on by unclear link paths and unresponsive design components. This research illustrates the direct correlation between anomalies in user behavior and lost revenue.



## 4.1 RAGE CLICK

Rage clicking is a feature that can be found within the field of user behavior analysis. This type of clicking refers to the frequent and rapid clicking of a particular feature or area on a website, which is typically exhibited by a user who is frustrated or upset. In the context of online retail, "rage clicking" can be an indicator of underlying problems that decrease the quality of the user experience and may even result in lost income. The reasons for rage-clicking could differ, and they can be broken down into many basic variables that cause frustration in users.

In the world of e-commerce, elements that are unreliable or don't work properly are a common source of rage-clicking. Users can turn to clicking the same button, link, or interactive feature multiple times in an effort to get the desired result when they come across a feature, button, or link that does not produce the expected response. This behavior starts from the expectation of a specific activity, such as adding an item to the shopping cart or advancing to the checkout page, only to be faced with disappointment owing to technological issues or delayed loading times. For example, adding an item to the shopping basket.

Another source of annoyance is navigation that is difficult to understand and user interfaces that are imprecise. Users of e-commerce systems that are characterized by complex menu structures, confusing labeling, or complicated pathways may indulge in rage-clicking in an attempt to discover particular products or information. A lack of a clear layout can result in confusion and prevent users from quickly achieving their goals, which can ultimately lead to irritation that presents itself as repetitive clicking behavior.

In addition, inconsistencies in inventory and availability can serve as a cause for rage-clicking. When users come across things that are listed as being in stock but they are unable to purchase them when they attempt to do so, their irritation may exhibit itself in the form of several clicks on the unreachable item or the error message that is linked with it. These kinds of situations can result not just in a poor user experience but also in lost sales opportunities, which has the potential to have an effect on the business's total revenue.

### 4.1.1 RAGE CLICK ALGORITHM

Within a given dataset of user interactions on an e-commerce website, the purpose of the algorithm is to search for instances of possible rage-clicking and identify them. The procedure consists of a few essential stages. In the beginning, we needed to make a boolean mask that we will refer to as the change mask. This mask signals any changes that occur in the page type, ele-

ment title, or 'session-id', which will indicate transitions between various interaction sessions. This helped to separate the interactions into their own individual sessions. The next step is to collect the change mask and assign a sequence number to each session. This effectively groups interactions that occur during the same session together.

Following that, we track the number of clicks that have occurred in a row inside each session and sequence in order to calculate the continuous count. After this, we calculate the amount of time that has passed between two successive clicks that occurred during the same session and sequence. We quantify the amount of time that has passed between these clicks by taking the timestamp of the first click in the group and subtracting it from the timestamps of succeeding clicks.

We established the criterion for rage clicks by utilizing a boolean mask to determine what defines a potential rage click. This criterion highlights sequences of clicks that fulfill certain characteristics, which are as follows: there must be at least 6 consecutive clicks within a 10-second interval or there must be at least 3 consecutive clicks within a 3-second window. Last but not least, we add a new column to the data frame called rage click and give the value 1 to rows in the data frame which satisfies the criteria for rage clicking. This allows us to identify possible occurrences of rage-clicking in the data. Figure 4.1, shows rage clicks within the selected time frame. Since the occurrence of rage clicks is not very common, we can conclude from the graph.

This algorithm is aimed at discovering patterns of rapid and repetitive clicking behavior that may signal user irritation, which may potentially lead to lost income on the e-commerce platform. This frustration may be caused by the platform's inability to meet the user's expectations. However, it is crucial to evaluate the accuracy of the algorithm and adjust it as necessary to guarantee that it is efficient in recognizing true rage-clicking behavior.

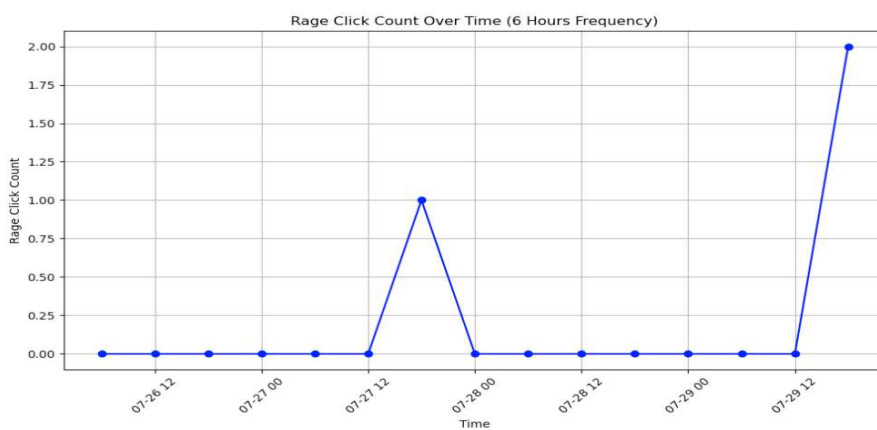


Figure 4.1: Rage Click Graph.

## 4.2 DEAD CLICK

Dead Clicks refer to the situations in which users engage with a website or application by clicking on items that do not have any kind of functional response. Dead clicks are an important occurrence. The user frequently experiences annoyance and misunderstanding as a result of these exchanges. In the context of online shopping, the term "dead clicks" can be an indicator of underlying usability flaws that prevent the interactions of users from occurring in an ongoing manner. These kinds of interactions may take place as a result of elements that cannot be clicked on, signs that are misleading, or buttons that do not carry out the operation that was intended for them. Users who come across Dead Clicks may have particular results in mind, such as product details or navigation to a different page, but instead, they are confronted with non-responsiveness when they attempt to access those features. This kind of experience might leave users feeling unsatisfied and give them an unfavorable impression of the operation of the website. Eliminating non-functional components and maintaining clear visual cues may direct users toward meaningful interactions and minimize behaviors motivated by irritation, making it essential to address the issue of dead clicks in order to improve the user experience and overall usability of a product. Businesses have the ability to set the path for easier navigation, increased customer happiness, and a more effective e-commerce platform by reducing the number of dead clicks on their websites.

### 4.2.1 DEAD CLICK ALGORITHM

Within a given dataset of user interactions, the presented technique is intended to detect and mark occurrences of "dead clicks," also known as inactive clicks. A "dead click" is an interaction that occurs when a user clicks on a component of a website or application that does not result in any meaningful action or navigation. In the initial step of the process, the algorithm generates a mask by comparing the session IDs in successive rows in order to identify session changes. This mask is put to use to single out the rows that mark the beginning of a new session. After that, the algorithm builds a dead click mask by taking into consideration the transition between the current row and the row that comes after it within a session. If the current row has a value that is not null in the 'element link' column, indicating a clicked element, and the next row has a null value in the 'element link' column, indicating that there is no next element, and if the following row has a null value in the 'next event first id' column, showing that there is no subsequent event, then the criterion for a "Dead Click" has been met. The 'dead click' column

in the dataset is then updated by the algorithm in accordance with the new information. In order to process the very last row of the very last session, the algorithm examines whether or not it satisfies the definition of a "Dead Click," and if it does, it labels it as such. Overall, this algorithm contributes to a more accurate analysis of user interactions and has the potential to assist in the improvement of user experience and interface design by systematically identifying instances of "Dead Clicks" based on specific conditions involving clicked elements, session changes, and the presence of future occurrences. We can see from the figure 4.2, that dead clicks within the selected time frame in the website.

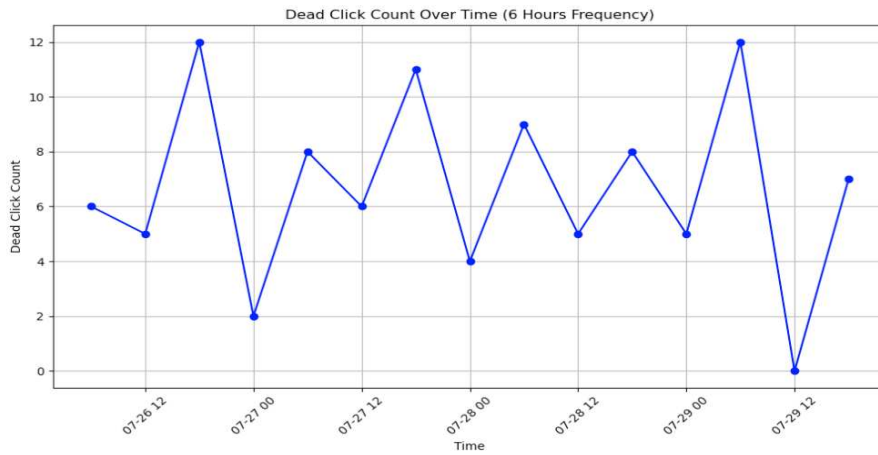


Figure 4.2: Dead Click Graph.

# 5

## Conclusion

Our first purpose in this paper was to find the cause of slow pages and track them. Dimensionality reduction was needed in that part because we have various attributes from resources, and it was very challenging to use them and create a meaningful output. As a solution to this problem, we created a feature code algorithm using binary encoding. In that way, we have feature codes for each specific case in the resources. The biggest advantage of this algorithm is that we are able to carry resource information with it. Before creating this code, we implemented various steps to achieve the best results. First, we applied the feature importance method to choose the most important features to create the feature codes. After we analyzed the distribution of these feature codes, the output we received made us realize that we could use these feature codes to detect if there was problematic behavior from the resource. By defining the threshold, we were able to detect the unexpected behavior of a resource at the defined frequency. The outputs of this algorithm were very satisfactory, the algorithm was able to catch every problematic resource occurrence on the web page.

The next focus was defining bounces accurately before making further analysis. Bounces are an important aspect of the e-commerce environment because, with the information on bounces, we can come to a conclusion about the performance of a website's content. If there are many bounces on the web page, it can be a sign that users are not interested in the content of the website, or there can be some other problem within the website. Every e-commerce company uses campaigns with different time frames for different products. These campaigns generally do not focus on every type of user; they mainly focus on specific types of users. Be-

cause of that, if there is a bounce after viewing the campaign page, we cannot count it as a bounce. Because it does not mean users are not interested in the web page or there is a problem with the page. Maybe they are just not interested in that specific campaign. After this analysis, we define an algorithm to detect campaign bounces and define the bounces accurately. After defining these true bounces, an algorithm is generated for calculating the campaign effect score. In this way, we can compare the campaign effect score values with the number of sessions at the same time. If there is an inverse ratio and if there are not many sessions on the website and the campaign impact score is high, we can assume that the bounces are related to the campaigns. On the other hand, if the campaign effect score is low and there are many sessions on the website, this might be an important warning. Using the algorithms can turn it into an effective decision-making approach. When administrators are prepared with this analytical understanding, they have the ability to go on a journey of informed strategy development, leading campaign initiatives, influencing user experiences, and driving through website optimization.

The next strategy was to calculate the purchase propensity of the users in general or for a specific product. Most of the companies lost a lot of revenue because they were not able to suggest to users possible products they could purchase depending on their user and purchase behavior on the website. In this section, we created a new feature named total weight that represents the intensity of the user's interaction on the website. This total weight became highly correlated with cart type. Cart type has binary values that indicate whether the user makes the purchase or not. Before creating this value, we did not have strong features and our model performances were very low. By creating total weight, we improved our model performances significantly. After this part, we calculate the user propensity by using the SVC decision function by choosing total weight as a selected feature. After calculating the propensity score, we used product categories with giving weight. We created three levels of product categories and gave weight depending on the similarities of the product categories. If there are users who purchase under similar categories, they are likely to be under the same cluster. Our propensity-based product recommendation algorithm represents a powerful tool for businesses seeking to optimize user engagement and sales. By integrating clustering, propensity scores, and machine learning, we have developed a robust system that provides strong recommendations, improving the user experience and driving conversions.

Finally, our last approach was detecting problems with user behavior. User behavior analysis is the systematic process of collecting, measuring, and interpreting data on how users interact with websites. This practice empowers businesses to gain insights into user engagement, detect underlying problems, streamline processes, and ultimately improve the overall user experience.

For e-commerce enterprises, such analyses reveal critical errors that might be curbing revenue potential. By spotting deviations, anomalies, and patterns in behavioral data, companies can address issues like "rage clicking" or "dead clicking," which point to broken links, unresponsive elements, or confusing interfaces. Two algorithms, the "Rage Click Algorithm" and the "Dead Click Algorithm," exemplify how systematic analysis of user behavior can unearth valuable insights. The "Rage Click Algorithm" identifies patterns of rapid and repetitive clicking behavior, potentially indicative of user irritation. By setting specific criteria for identifying "rage clicks," this algorithm assists in pinpointing user frustrations and provides a foundation for improving the platform's performance and meeting user expectations. Similarly, the "Dead Click Algorithm" focuses on identifying "dead clicks," interactions that offer no meaningful results. This algorithm aids in fine-tuning user experience and interface design by systematically detecting instances of "Dead Clicks."

This study delves into the complex challenges that e-commerce businesses face in efficiently recognizing revenue and maintaining financial health. We uncover the significant impact of slow-loading pages, problematic web resources, and errors on business models by analyzing real-world data and user behavior patterns. We address these challenges and unlock the full potential of the digital marketplace through thorough analysis and innovative algorithmic solutions.

Our research introduces novel algorithms that not only detect but also offer practical, data-driven solutions to issues in the online commerce environment. These algorithms identify problems in real time and provide insights into how to improve marketing tactics, improve user interactions, and increase overall customer satisfaction. These tools, which use machine learning, user behavior analysis, and predictive modeling, enable businesses to navigate the unpredictable e-commerce sector and achieve long-term growth.

Given the rapid evolution of the e-commerce industry, versatility is essential. Our research provides e-commerce companies with a road map for overcoming challenges, optimizing platforms, and increasing financial resilience. Despite shifting trends and technological advancements, decision-makers can use these insights to maintain innovation and customer satisfaction. Fundamentally, this research highlights the relationship between technological innovation and customer-centric strategies. Data analytics, machine learning, and user behavior insights can help e-commerce businesses grow. Improving revenue recognition, improving user experiences, and maintaining growth are all dependent on attention to detail, adaptability, and a commitment to data-driven improvement. E-commerce companies can strengthen their global market presence and thrive in the connected world of digital commerce by embracing these efforts.





# References

- [1] S. Parashar, "E-commerce: A big approach for business," *International Journal of Technology and Business Management*, vol. 12, no. 3, pp. 192–205, July–September 2022.
- [2] S. meng Liu, "An empirical study on e-commerce's effects on economic growth," *2012 First National Conference for Engineering Sciences (FNCES 2012)*, pp. 0081–0084, 2012. [Online]. Available: <https://api.semanticscholar.org/CorpusID:9235695>
- [3] J. Hoxmeier, C. Dicesare, and Manager, "System response time and user satisfaction: An experimental study of browser-based applications," *Proceedings of the Association of Information Systems Americas Conference*, 01 2000.
- [4] R. Burke, A. Felfernig, and M. H. Göker, "Recommender systems: An overview," *AI Mag.*, vol. 32, pp. 13–18, 2011. [Online]. Available: <https://api.semanticscholar.org/CorpusID:5935762>
- [5] M. Vlachos, V. G. Vassiliadis, R. Heckel, and A. Labbi, "Toward interpretable predictive models in b2b recommender systems," *IBM J. Res. Dev.*, vol. 60, pp. 11:1–11:12, 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:11533684>
- [6] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on web usage mining," *Commun. ACM*, vol. 43, pp. 142–151, 2000. [Online]. Available: <https://api.semanticscholar.org/CorpusID:165932>
- [7] S. Kaski and J. Peltonen, "Dimensionality reduction for data visualization [applications corner]," *IEEE Signal Processing Magazine*, vol. 28, pp. 100–104, 2011. [Online]. Available: <https://api.semanticscholar.org/CorpusID:15117438>
- [8] P. Jindal and D. Kumar, "A review on dimensionality reduction techniques," *International Journal of Computer Applications*, vol. 173, pp. 42–46, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:54007187>

- [9] C. Seger, “An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing,” 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:250534659>
- [10] J. Jaiswal and R. Samikannu, “Application of random forest algorithm on feature subset selection and classification and regression,” 2017 *World Congress on Computing and Communication Technologies (WCCCT)*, pp. 65–68, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:24747351>
- [11] B. Chong, “K-means clustering algorithm: a brief review,” *Academic Journal of Computing & Information Science*, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:244185242>
- [12] C. Ding and X. He, “K-means clustering via principal component analysis,” *Proceedings of the twenty-first international conference on Machine learning*, 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:11356277>
- [13] M. Vendivel, “Virtual rebel website: A strategy to increase user engagement through bounce rate analysis,” 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:27430415>
- [14] K. Falk, *Practical Recommender Systems*. Manning Publications, 2019.
- [15] M. Lerato, O. A. Esan, A.-D. Ebuloluwa, S. M. Ngwira, and T. Zuva, “A survey of recommender system feedback techniques, comparison and evaluation metrics,” 2015 *International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–4, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:15962035>
- [16] G. Jawaheer, P. Weller, and P. Kostkova, “Modeling user preferences in recommender systems,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 4, pp. 1 – 26, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:18603824>
- [17] J. C. Stoltzfus, “Logistic regression: a brief primer.” *Academic emergency medicine: official journal of the Society for Academic Emergency Medicine*, vol. 18 10, pp. 1099–104, 2011. [Online]. Available: <https://api.semanticscholar.org/CorpusID:33452324>
- [18] R. Hu, X. Ian Zhu, Y. Zhu, and J. Gan, “Robust svm with adaptive graph learning,” *World Wide Web*, vol. 23, pp. 1945 – 1968, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:255135635>

- [19] S. J. Kazemitabar, A. A. Amini, A. Bloniarz, and A. Talwalkar, "Variable importance using decision trees," in *NIPS*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:28116257>
- [20] A. B. Yedidia, "Against the f-score," 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:209900222>
- [21] I. ul Haq, I. Gondal, P. Vamplew, and S. Brown, "Categorical features transformation with compact one-hot encoder for fraud detection in distributed environment," in *Australasian Data Mining Conference*, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:67751877>
- [22] K. R. Shahapure and C. K. Nicholas, "Cluster quality analysis using silhouette score," *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 747–748, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:227122930>
- [23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [24] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," vol. 2, 2011. [Online]. Available: <https://doi.org/10.1145/1961189.1961199>
- [25] A. Basudhar and S. Missoum, "Adaptive explicit decision functions for probabilistic design and optimization using support vector machines," *Computers & Structures*, vol. 86, pp. 1904–1917, 2008. [Online]. Available: <https://api.semanticscholar.org/CorpusID:16071085>
- [26] R. M. Neal, "Pattern recognition and machine learning, by christopher m. bishop," *Technometrics*, vol. 49, 2007. [Online]. Available: <https://api.semanticscholar.org/CorpusID:124716054>
- [27] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, pp. 1145–1159, 1997. [Online]. Available: <https://api.semanticscholar.org/CorpusID:13806304>

- [28] Y. Jung, “Multiple predicting k-fold cross-validation for model selection,” *Journal of Nonparametric Statistics*, vol. 30, pp. 197 – 215, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:125195896>
- [29] C. Xu, C. Du, G. Zhao, and S. Yu, “A novel model for user clicks identification based on hidden semi-markov,” *J. Netw. Comput. Appl.*, vol. 36, pp. 791–798, 2013. [Online]. Available: <https://api.semanticscholar.org/CorpusID:37756027>
- [30] G. Slanzi, J. A. Balazs, and J. D. Velásquez, “Combining eye tracking, pupil dilation and eeg analysis for predicting web users click intention,” *Inf. Fusion*, vol. 35, pp. 51–57, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:42409694>
- [31] M. Zengin and B. Carterette, “User click detection in ideal sessions,” *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:18161726>

# Acknowledgments

I would like to express my deepest gratitude to my supervisor, Tomaso Erseghe, for his invaluable guidance and unwavering positive attitude throughout my thesis journey.

Heartfelt appreciation goes to my dad, whose unwavering support and boundless love have made me feel that I am never alone in this journey. Your endless care has truly made me the luckiest son, and I carry immense pride in being your child.

To my brother, you are my eternal source of inspiration. Your role in my life has been paramount, leading me to where I stand today and providing a constant reminder of what I can achieve. Your influence will resonate with me forever.

A special tribute is reserved for my mom, my guiding light and best friend throughout my life. Your teachings have instilled in me the finest qualities, and your soul is the most beautiful thing I have ever seen and will stay like this forever.

To my beloved Zeynep, your presence has been a constant source of strength and encouragement not just during my academic pursuit but throughout the 26 years of our shared journey.

My flatmates, go on this journey alongside me. Bilge, you were my main drive to come to this master, your support and presence have been my pillars of strength. Umut, I appreciate you always taking the time to watch funny videos with me. I am happy to have you as the person who made me laugh the most.

Nilsu and Rocio, my incredible friends, you have been my rocks throughout this master's. The memories we have together were the real joy of this journey and as we step into the future, I can not wait for the new adventures we will share. Nilsu, I will see you in the corner, and Rocio, I will carry you whenever you need me.

Endrit, even when I was doubting myself, you were always there to prove me wrong. I am so grateful that I had the opportunity to get to know you through this master's and that you have become my second brother with whom I can share my ups and downs.

At this moment, I extend my gratitude to everyone who has contributed to my growth. Each of you holds a unique place in my heart, and I am humbled by the connections we share.

Thank you.