

Shopping Intensity: Unveiling Consumer Segments Through Clickstream Data

*A project report submitted to
Jawaharlal Nehru Technological University Kakinada, in the partial
Fulfillment for the Award of degree of
BACHELOR OF TECHNOLOGY
IN*

COMPUTER SCIENCE AND ENGINEERING

Submitted by

K. SUBBARAO	21491A05R3
A. SURYA	21491A05U6
G. ANVITHA	21491A05S4
G. CHARITHA	21491A05T7
K. SHANMUKHA	21491A05U4
K. KHADAR	21491A05U5
SK. IMRAN	21491A05W8

Under the esteemed guidance of

Mr.K. Kishore Babu, MCA, M.Tech., (Ph.D)

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

QIS COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

An ISO 9001:2015 Certified institution, approved by AICTE & Reaccredited by NBA, NAAC 'A+' Grade

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

VENGAMUKKAPALEM, ONGOLE – 523 272, A.P

April 2025

QIS COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

An ISO 9001:2015 Certified institution, approved by AICTE & Reaccredited by NBA, NAAC 'A+' Grade

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

VENGAMUKKAPALEM, ONGOLE:-523272, A.P

April 2025



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the *technical report* entitled **"Shopping Intensity: Unveiling Consumer Segments Through Clickstream Data"** is a Bonafide work of the following final B.Tech students in the partial fulfillment of the requirement for the award of the degree of bachelor of technology in COMPUTER SCIENCE AND ENGINEERING for the academic year 2024-2025.

K. SUBBARAO	21491A05R3
A. SURYA	21491A05U6
G. ANVITHA	21491A05S4
G. CHARITHA	21491A05T7
K. SHANMUKHA	21491A05U4
K. KHADAR	21491A05U5
SK. IMRAN	21491A05W8

Signature of the guide
Mr.K. Kishore Babu, MCA, M.Tech., (Ph.D)

*Assistant Professor,
Department of CSE (AIML)*

Signature of Head of Department
Dr. D. Bujji Babu, M. Tech., Ph.D.,

*Professor and HOD.,
Department of CSE*

Signature of External Examiner

ACKNOWLEDGEMENT

We thank the almighty for giving us the courage and perseverance in completing the project. It is an acknowledgement for all those people for all those people who have given us their heartfelt cooperation in making in making the major project a grand success.

We would like to place on record our deep sense of gratitude to the Honorable Executive Chairman **Dr. N.S. Kalyan Chakravarthy**, Honorable Executive Vice Chairman **Dr. N. Sri Gayatri Devi** and Principal **Dr. Y.V. Hanumantha Rao** for providing the necessary facilities to carry out the project work.

We express our gratitude to the Head of the Department of CSE, **Dr. D. Bujji Babu, M. Tech, Ph. D**, QIS College of Engineering & Technology, Ongole for his constant supervision, guidance and co-operation throughout the project.

We would like to express our thankfulness to our project guide **Mr.K. Kishore Babu, MCA, M.Tech.,(Ph.D)**, Assistant Professor – Department of CSE(AIML), QIS College of Engineering & Technology, Ongole for his constant motivation and valuable help throughout the project work.

We would like to express our thankfulness to CSCDE & DPSR for their constant motivation and valuable help throughout the project.

Finally, we would like to thank our Parents, Family and Friends for their cooperation in completing this project.

TEAM MEMBERS

K. SUBBARAO	21491A05R3
A. SURYA	21491A05U6
G. ANVITHA	21491A05S4
G. CHARITHA	21491A05T7
K. SHANMUKHA	21491A05U4
K. KHADAR	21491A05U5
SK. IMRAN	21491A05W8

DECLARTION

We hereby declare that the project work entitled “**Shopping Intensity: Unveiling Consumer Segments Through Clickstream Data**” done under the guidance **Mr. K. Kishore Babu. MCA, M.Tech., (Ph.D)**, Assistant Professor – **Department of CSE (AIML)**, is being submitted to the “Department of Computer Science & Engineering (AIML)”, QIS College of Engineering & Technology, Ongole is of our own and has not been submitted to any other University or Educational Institution for any degree.

TEAM MEMBERS

K. SUBBARAO	21491A05R3
A. SURYA	21491A05U6
G. ANVITHA	21491A05S4
G. CHARITHA	21491A05T7
K. SHANMUKHA	21491A05U4
K. KHADAR	21491A05U5
SK. IMRAN	21491A05W8

ABSTRACT

Understanding consumers is a top priority for businesses that want to maximize and generate sales. we analyze purchase intensity through clickstream analysis to determine clear segments of consumers based on browser behavior. Apply k-means to classify users using wise attributes such as browsing, session length, and purchasing courses. This study provides insights on many types of buyers, ranging from rare browsers to high quality buyers, allowing marketing to adapt and improve customer loyalty. Through data-controlled segmentation, research shows the underlying dynamics of online shopping, providing insights that can be implemented into e-commerce websites. The results are based on data to improve target marketing and customer maintenance by contributing to the deeper understanding of interaction with consumers in Internet transactions. The visualization of the consumer segments can be very crucial for businesses and to increase their revenue through certain users.

Keyword: Clickstream analysis, AI, ML, Consumer segmentation, E-commerce research, Online shopping personalization techniques.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF TABLES	v
	LIST OF FIGURES	vi
	LIST OF SYMBOLS AND ABBREVIATIONS	vii
1	INTRODUCTION	1-3
	1.1 Motivation	1-2
	1.2 Objectives	2-3
	1.3 Organization of Thesis	3
2	LITERATURE SURVEY	4-6
	2.1 Related Work	4-5
	2.2 Research Gap	6
	2.3 Conclusion	6
3	SYSTEM ANALYSIS	7-11
	3.1 Existing System	7-8
	3.2 Proposed System	9-10
	3.3 Software Specification	10
	3.4 Hardware Specification	11
4	ML TECHNIQUES	12-24
5	SYSTEM DESIGN	25-32
	5.1 System Architecture	25-26
	5.2 Module Description	26- 27

	5.3 UML Diagrams	28-32
	5.3.1 Use Case Diagram	28
	5.3.2 Class Diagram	28-29
	5.3.3 Sequence Diagram	29
	5.3.4 Collaboration Diagram	29-30
	5.3.5 Deployment Diagram	30
	5.3.6 Activity Diagram	30-31
	5.3.7 Component Diagram	31
	5.3.8 ER Diagram	31-32
	5.3.9 DFD Diagram	32
6	PROPOSED METHODOLOGY	33-34
	6.1 System	33
	6.1.2 Data Base	33-34
	6.1.3 Create Data Set	34
	6.1.4 Pre-Processing	34
	6.1.5 Training	34
	6.1.6 Prediction	34
7	IMPLEMENTATION	35-38
	7.1 Algorithm	35-36
	7.2 Implementation of the Model	36-38
8	SOFTWARE LIBRARIES	39-45
	8.1 Python	39-43
	8.2 Operating System	43

	8.3 Pandas	43-44
	8.4 Streamlit	44
	8.5 Plotly	44-45
	8.6 Scikit-learn	45
	8.7 Matplotlib	45
9	SYSTEM STUDY AND TESTING	46-50
	9.1 Feasibility Study	46-47
	9.1.1 Economical Feasibility	46
	9.1.2 Technical Feasibility	46
	9.1.3 Social Feasibility	47
	9.2 System Testing	47-50
	9.2.1 Unit Testing	47
	9.2.2 Integration Testing	47-48
	9.2.3 Acceptance Testing	48
	9.2.4 Functional Testing	48-49
	9.2.5 White Box Testing	49
	9.2.6 Black Box Testing	49-50
10	SOURCE CODE	51-54
11	OUTPUT SCREENSHOTS	55-56
12	RESULTS AND DISCUSSION	57-59
13	CONCLUSION AND FUTURE WORK	60
	13.1 Conclusion	60
	13.2 Future Work	60
	REFERENCES	61-62

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
9.1	Testcases	50

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
5.1	System Architecture	25
5.3.1	Use Case Diagram	28
5.3.2	Class Diagram	28
5.3.3	Sequence Diagram	29
5.3.4	Collaboration Diagram	29
5.3.5	Deployment Diagram	30
5.3.6	Activity Diagram	30
5.3.7	Component Diagram	31
5.3.8	ER Diagram	31
5.3.9	DFD Diagram	32
7.1	Project Flow	33
11.1	Upload page	55
11.2	Output page	55
12.1.1	Conversion Funnel	57
12.1.2	Navigation Analysis	57
12.1.3	Session details	58
12.1.4	Session Depth Distribution	59
12.1.5	Click patterns	59

LIST OF SYMBOLS AND ABBREVIATIONS

ML	Machine Learning
AI	Artificial Intelligence
SVM	Support Vector Machine
DBSCAN	Density-Based spatial clustering of applications with noise

CHAPTER-1

INTRODUCTION

In the digital age, understanding consumer behavior is crucial for businesses to stay competitive. Clickstream data recording the sequence of clicks made by users on a website provides a wealth of information about consumer interactions. This data can reveal insights into browsing patterns, purchase decisions, and user preferences. By segmenting consumers based on this data, businesses can tailor their marketing strategies and improve user experience.

This study aims to analyze clickstream data to uncover distinct consumer segments, understand their behaviors, and propose a system that enhances existing methods of consumer analysis. The goal is to address the limitations of current systems and provide a framework for more effective consumer segmentation and personalized marketing.

With the vast growth of e-commerce, understanding these behavioral patterns has become more critical than ever. Consumers exhibit a wide range of shopping behaviors, from casual browsing to intense shopping sprees. The ability to segment these behaviors can offer businesses the opportunity to tailor marketing strategies, improve user experience, and ultimately increase conversion rates.

However, the challenge lies in making sense of the extensive and often complex clickstream data. While data collection is straightforward, the real value comes from analyzing this data to identify distinct consumer segments. These segments can range from shoppers who spend extensive time comparing products to those who make quick purchasing decisions. Without a structured approach to identifying these segments, businesses risk missing valuable opportunities to engage with their consumers effectively.

1.1 Motivation:

Monitoring and analyzing clickstream data from different clients offers valuable insights and serves multiple motivations for businesses, marketers, and developers. The main reasons are:

1. **Understanding User Behavior:** Clickstream data tracks the sequence of actions users take—such as pages visited, links clicked, and time spent on each step. This helps reveal how clients interact with a website, app, or platform, identifying patterns, preferences, and pain points.

2. **Improving User Experience:** By analyzing where users drop off, linger, or encounter friction, businesses can optimize navigation, streamline interfaces, and enhance functionality to make the experience more intuitive and engaging.
3. **Personalization:** Tracking individual click patterns allows companies to tailor content, recommendations, or promotions to specific clients. For example, e-commerce sites can suggest products based on browsing history, increasing relevance and satisfaction.
4. **Boosting Conversion Rates:** Clickstream analysis highlights what drives users toward desired actions (e.g., purchases, sign-ups) and what deters them. This enables targeted improvements to funnels, calls-to-action, or checkout processes to maximize conversions.
5. **Marketing Optimization:** Understanding which links, ads, or campaigns drive traffic and engagement helps refine marketing strategies. It also reveals which channels or messages resonate most with different client segments.
6. **Performance Monitoring:** Clickstream data can expose technical issues—like slow-loading pages or broken links—allowing teams to address problems that might otherwise frustrate users and harm retention.
7. **Business Intelligence and Decision-Making:** Aggregated data provides a big-picture view of trends, such as peak usage times or popular features. This informs resource allocation, product development, and strategic planning.
8. **Competitive Advantage:** Companies that leverage clickstream insights can stay ahead by quickly adapting to client needs and preferences, outpacing competitors who rely on less granular feedback.

1.2 Objective

The objective of this project is to analyze clickstream data to gain a deeper understanding of user behavior and segment users based on their browsing patterns and interactions. Clickstream data, which captures user activities such as page views, clicks, and conversions, provides valuable insights into how users navigate a website. By leveraging machine learning techniques like K-Means and DBSCAN clustering, the project identifies distinct user groups, including high-engagement users, casual browsers, cart abandoners, and potential buyers. These clusters help in understanding how different types of users interact with the website and what factors influence their decision-making.

To achieve this, key metrics such as session duration, total page views, bounce rate, click patterns, and conversion rates are extracted and analyzed. These features allow the model to group users based on similarities in their browsing behavior. K-Means clustering helps in segmenting users into predefined groups based on numerical patterns, while DBSCAN is useful for identifying anomalies such as bots or irregular browsing behavior. The insights gained from this segmentation process enable businesses to optimize their marketing strategies, personalize user experiences, improve customer retention, and enhance website performance. For example, identifying frequent visitors who do not complete purchases can help businesses implement targeted promotions or retargeting strategies. Similarly, recognizing high-value customers allows for better loyalty programs and personalized recommendations.

Ultimately, this project provides a data-driven approach to improving website efficiency and engagement by helping businesses understand their audience, refine user experience strategies, and drive higher conversions. By applying machine learning techniques to real-time user interactions, organizations can make informed decisions that enhance customer satisfaction and boost overall business performance.

1.3 Organization of Thesis

In the rapidly evolving digital marketplace, understanding user behavior is crucial for improving customer engagement and increasing conversion rates. However, businesses often struggle to analyze vast amounts of clickstream data to identify meaningful patterns in user interactions. Traditional analytics methods fail to effectively segment users based on their browsing behavior, leading to inefficient marketing strategies and poor user experience.

This project aims to address this challenge by leveraging machine learning techniques, specifically K-Means and DBSCAN clustering, to segment users based on their session duration, page views, bounce rate, and conversion actions. By identifying distinct user groups such as high-engagement users, casual browsers, and cart abandoners, businesses can make data-driven decisions to enhance personalization, optimize website design, and improve customer retention. The goal is to develop an automated, scalable solution that provides actionable insights, enabling businesses to maximize their online performance and drive higher conversions.

CHAPTER-2

LITERATURE SURVEY

2.1 Related work

Manav Gumber, Apoorv Jain, A. L. Amutha (2021). " Predicting Customer Behaviour by Analysing Clickstream Data"

This paper explores methodologies for predicting costs associated with specific behaviours by analysing peak stream data. It emphasizes the importance of real-time data analytics in identifying patterns that lead to increased expenses, enabling organizations to take proactive cost-management measures.

By monitoring peak periods in data streams, the study demonstrates how organizations can optimize operations and reduce unnecessary expenditures. It highlights the potential of leveraging peak stream data analysis as a strategic tool to enhance efficiency and profitability.

Zhanming Wen , Weizhen Lin , Hongwei Liu (2023). "Machine-Learning-Based Approach for Anonymous Online Customer Purchase Intentions Using Clickstream Data "

This paper presents a machine learning-based approach to predict anonymous online customer purchase intentions using clickstream data. By incorporating multi-behavioural trendiness (MBT) and product popularity (POP) metrics, the study enhances prediction accuracy, demonstrating effectiveness on a dataset with over 3 million clicks.

The findings highlight that integrating MBT and POP improves predictive performance and reduces the time required for reliable forecasts. This approach offers valuable insights for e-commerce platforms to optimize customer engagement and boost conversion rates.

Melina Zavali (2021). “Shopping Hard or Hardly Shopping: Revealing Consumer Segments Using Clickstream Data”

This paper investigates consumer segmentation by analyzing clickstream data from a UK-based fast-fashion retailer's e-commerce site. Utilizing the partitioning around medoids algorithm on samples of 10,000 unique consumer visits, the study identifies six distinct consumer segments. Notably, the largest segment, termed "mobile window shoppers," generates the lowest revenue, while one of the smallest segments, "visitors with a purpose," contributes the highest revenue.

The findings highlight the potential of clickstream analysis in uncovering unique consumer segments and linking them to revenue generation. This approach offers valuable insights for marketing strategies, enabling more tailored targeting of customer segments to enhance profitability. The study underscores the importance of leveraging big data analytics in the apparel retailing industry to inform marketing decisions and optimize operations.

M. Anitha, K. Baby Ramya, MD. Zeenath (2024). " Predicting Online Shopping Behaviour Through Clickstream Analysis "

This paper explores the use of machine learning to predict online helping behaviour through clickstream data analysis. By examining user session timestamps, visit durations, and specific interactions, the study identifies patterns that indicate helping tendencies in online platforms.

The research employs classification algorithms to build predictive models, enabling platforms to enhance user experiences and optimize engagement strategies. The findings offer valuable insights for designing more effective online support systems.

Hang Lee (2024). "Interest-Based E-Commerce and Users' Purchase Intention on Social Network Platforms"

The paper discusses how social media platforms are changing the way people shop online. Instead of users searching for products, social media platforms use algorithms to suggest items based on their interests, leading to spontaneous purchases. Social interactions, such as reviews and recommendations, also influence buying decisions.

It also highlights Xiaohongshu, a platform that combines social networking with shopping, attracting users who value lifestyle over just price. The study emphasizes how influencer marketing and personalized content help build trust and encourage purchases, shaping the future of online shopping.

Silvia Cachero-Martínez , Rodolfo Vazquez-Casielles (2021). : "Building consumer loyalty through e-shopping experiences: The mediating role of emotions "

This paper explores how e-Shopping experiences influence consumer loyalty, highlighting the role of emotions. It finds that factors like website design, ease of use, and personalization create positive emotions, leading to trust and satisfaction, which strengthen customer loyalty.

The study suggests that businesses can enhance loyalty by improving user experience, ensuring secure transactions, and personalizing interactions. By fostering positive emotions, e-commerce platforms can boost customer retention and long-term engagement.

2.2 Research gap

Limited Consumer Segmentation: Existing systems for analyzing clickstream data often provide basic insights but lack the ability to effectively segment consumers based on their behaviors and preferences.

Inadequate Personalization: Current methods do not offer personalized marketing strategies tailored to distinct consumer segments, leading to missed opportunities for businesses to engage effectively with their customers.

Complex Data Analysis: The challenge of analyzing extensive and complex clickstream data remains, with many existing tools failing to provide actionable insights or a deeper understanding of consumer behavior.

Need for Advanced Techniques: There is a need for more sophisticated analytical approaches that can handle large-scale data and provide granular insights into consumer interactions, which are crucial for enhancing user experience and increasing conversion rates.

Lack of Real-Time Processing: Most segmentation models rely on batch processing, making it difficult to respond dynamically to user behavior. Implementing real-time data streaming could significantly enhance responsiveness.

2.3 Conclusion

The literature review highlights that leveraging clickstream data for consumer segmentation provides significant advantages for e-commerce businesses. Traditional and advanced clustering methods, combined with machine learning and deep learning techniques, serve as powerful tools for analyzing and predicting consumer behavior. Future research should focus on developing more robust and scalable approaches, integrating real-time processing for dynamic user tracking, and enhancing privacy-preserving techniques to maintain data security. Addressing these challenges will ensure that clickstream-based consumer segmentation continues to evolve as an essential tool for personalized marketing, customer retention, and business growth in an increasingly digital marketplace.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

This is because clickstream data are a form of big data, and thus, it is characterized by large volumes, which are difficult to process. As shown in the ASOS example, however, if the appropriate data analytics methods are employed, clickstream data can derive valuable insights to support marketing activities. As demonstrated by ASOS, one potentially useful application of clickstream data in marketing research and practice is consumer segmentation. Consumer segmentation is defined as the division of consumers into groups of buyers who share distinct characteristics and behaviors that might require separate products or marketing mixes. Recognizing consumer heterogeneity, much has been written about consumer segmentation in the offline environment, there is, however, a handful of research on consumer segmentation according to consumers' online behavior. Existing studies on consumer online segmentation are limited in terms of insights.

Disadvantages

Existing systems for monitoring and analyzing clickstream data—essentially the digital footprints users leave as they navigate websites or apps—come with several disadvantages. These systems are widely used in fields like e-commerce, marketing, and user experience design, but they're not without flaws. The major disadvantages are:

1. Data Overload and Noise:

Clickstream data can generate massive volumes of information, often capturing every mouse movement, click, or scroll. Many systems struggle to filter out irrelevant actions—like accidental clicks or bot traffic—leading to cluttered datasets that obscure meaningful patterns. Processing this firehose of data requires significant computational resources, and without proper refinement, insights can get buried in the noise.

2. Privacy and Compliance Challenge:

These systems often collect detailed user behavior, raising red flags under regulations like GDPR or CCPA. Tracking cookies, IP addresses, or session IDs can inadvertently scoop up personal data, and many older systems weren't built with consent management or anonymization in mind. Non-compliance risks fines, while overzealous data collection can erode user trust.

3. Limited Real-Time Analysis:

Traditional clickstream tools—like Google Analytics or legacy enterprise software—often rely on batch processing, meaning data is analyzed after the fact rather than live. This delay can be a problem for businesses needing instant insights, like spotting a sudden drop in conversions or reacting to user frustration mid-session.

4. Incomplete Contextual Understanding:

Clickstream data tells you **what** users did (e.g., clicked a button, left a page), but not **why**. Existing systems often lack integration with qualitative data—like user intent, emotions, or external factors (e.g., a trending news event driving traffic). Without this, you're stuck with a partial picture, making it hard to act on findings confidently.

5. Scalability Issues:

As traffic grows, some systems choke. Legacy tools or those with rigid architectures might slow down, crash, or require costly upgrades to handle millions of simultaneous users. This is especially true for platforms with basic cloud integration or outdated databases not optimized for high-velocity data streams.

6. Integration Gaps:

Many clickstream solutions don't play nicely with other data sources—think CRM systems, social media metrics, or offline sales data. Siloed analytics limit the ability to see the full customer journey, forcing teams to manually stitch together insights, which is time-consuming and error-prone.

7. Accuracy and Attribution Problems:

Cross-device tracking (e.g., a user switching from phone to laptop) or multi-touch attribution (e.g., which ad really drove the sale?) often trips up these systems. Cookies get blocked, sessions expire, and users clear their caches, leading to fragmented or duplicate data that skews results.

3.2 Proposed System

The proposed system for revealing consumer segments using clickstream data involves a multi-step process that leverages advanced data analytics and machine learning techniques to understand and categorize consumer behavior on e-commerce platforms. The system begins with the collection of clickstream data, which includes detailed logs of user interactions, such as page views, clicks, product searches, and purchase actions. This raw data is then preprocessed to remove noise and standardize the format, ensuring consistency and accuracy. Following preprocessing, the data undergoes feature extraction where meaningful attributes are identified. These features might include session duration, frequency of visits, types of products viewed, and patterns of navigation through the website. Next, clustering algorithms such as K-means, DBSCAN, or hierarchical clustering are applied to group users into distinct segments based on their behavioral patterns. These segments can range from 'browsers' who frequently visit but rarely purchase, to 'bargain hunters' who extensively compare prices before making a purchase, to 'loyal customers' who regularly buy specific brands or types of products. To enhance the segmentation process, machine learning models are employed to analyze and predict user behavior. For instance, using supervised learning techniques, the system can classify new users into predefined segments based on their initial interactions with the platform. Additionally, advanced models such as neural networks can uncover complex patterns and insights that traditional methods might miss. The resulting consumer segments are then validated and refined using metrics such as silhouette scores and Davies-Bouldin index to ensure they are meaningful and actionable.

The final results include visualizations of user clusters, insights into behavioral trends, and recommendations for personalized marketing strategies. These insights empower businesses to optimize their website structure, improve targeted advertising, and enhance customer engagement, ultimately leading to higher conversion rates and improved user experience.

Advantages :

Automated User Segmentation – The model automatically groups users based on behavior patterns, eliminating the need for manual analysis.

Better Customer Insights – By identifying high-engagement users, casual browsers, and cart abandoners, businesses can understand user preferences and improve targeting strategies.

Improved Marketing Strategies – Personalized marketing campaigns can be designed for

different user segments, increasing conversion rates and customer retention.

Real-Time Analysis – The model can process new clickstream data dynamically, allowing businesses to respond quickly to changing user behavior.

Fraud Detection & Bot Filtering – DBSCAN helps detect *anomalous activities*, such as bot-generated traffic or unusual browsing patterns, enhancing security.

Optimized Website Experience – Insights from clustering can guide businesses in refining website design, improving navigation, and reducing bounce rates.

Scalability & Adaptability – The model can handle large datasets and adapt to different industries, making it suitable for e-commerce, digital marketing, and content platforms.

3.3 Software Requirement

Software is a collection of instructions, procedures, and documentation that performs different tasks on a computer system. we can say also Computer Software is a programming code executed on a computer processor. The code can be machine-level code or code written for an operating system. Examples of software are MS- Word, Excel, PowerPoint, Google Chrome, Photoshop, MySQL, etc. System Software is a component of Computer Software that directly operates with Computer Hardware which has the work to control the Computer's Internal Functioning and also takes responsibility for controlling Hardware Devices such as Printers, Storage Devices, etc. Types of System Software include Operating systems, Language processors, and Device Drivers. Application Software are the software that works the basic operations of the computer. It performs a specific task for users. Application Software basically includes Word Processors, Spreadsheets, etc. Types of Application software include General Purpose Software, Customized Software, etc.

Operating System	: Windows 7/8/10
Server Side Script	: CSS, HTML
Programming Language	: Python
Libraries	: Numpy, Pandas, plotly, scikit-learn, Streamlit
IDE/Workbench	: PyCharm

3.4 Hardware Requirement

Hardware refers to the physical components of a computer. Computer Hardware is any part of the computer that we can touch these parts. These are the primary electronic devices used to build up the computer. Examples of hardware in a computer are the Processor, Memory Devices, Monitor, Printer, Keyboard, Mouse, and Central Processing Unit.

Processor	- I3/Intel Processor
RAM	- 8GB (min)
Hard Disk	- 128 GB
Key Board	- Standard Windows Keyboard

CHAPTER 4

ML TECHNIQUES

What is Machine Learning :

Machine Learning, as the name says, is all about machines learning automatically without being explicitly programmed or learning without any direct human intervention. This machine learning process starts with feeding them good quality data and then training the machines by building various machine learning models using the data and different algorithms. The choice of algorithms depends on what type of data we have and what kind of task we are trying to automate.

As for the formal definition of Machine Learning, we can say that a Machine Learning algorithm learns from experience E with respect to some type of task T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E . For example, If a Machine Learning algorithm is used to play chess. Then the experience E is playing many games of chess, the task T is playing chess with many players, and the performance measure P is the probability that the algorithm will win in the game of chess.

Types of Machine Learning :

Let's see the different types of Machine Learning now:

1. Supervised Machine Learning

Imagine a teacher supervising a class. The teacher already knows the correct answers but the learning process doesn't stop until the students learn the answers as well. This is the essence of Supervised Machine Learning Algorithms. Here, the algorithm learns from a training dataset and makes predictions that are compared with the actual output values. If the predictions are not correct, then the algorithm is modified until it is satisfactory. This learning process continues until the algorithm achieves the required level of performance. Then it can provide the desired output values for any new inputs.

2. Unsupervised Machine Learning

In this case, there is no teacher for the class and the students are left to learn for themselves! So for Unsupervised Machine Learning Algorithms, there is no specific answer to be learned

and there is no teacher. In this way, the algorithm doesn't figure out any output for input but it explores the data. The algorithm is left unsupervised to find the underlying structure in the data in order to learn more and more about the data itself.

3. Semi-Supervised Machine Learning

The students learn both from their teacher and by themselves in Semi-Supervised Machine Learning. And you can guess that from the name itself! This is a combination of Supervised and Unsupervised Machine Learning that uses a little amount of labeled data like Supervised Machine Learning and a larger amount of unlabeled data like Unsupervised Machine Learning to train the algorithms. First, the labeled data is used to partially train the Machine Learning Algorithm, and then this partially trained model is used to pseudo-label the rest of the unlabeled data. Finally, the Machine Learning Algorithm is fully trained using a combination of labeled and pseudo-labeled data.

4. Reinforcement Machine Learning

Well, here are the hypothetical students who learn from their own mistakes over time (that's like life!). So the Reinforcement Machine Learning Algorithms learn optimal actions through trial and error. This means that the algorithm decides the next action by learning behaviors that are based on its current state and that will maximize the reward in the future. This is done using reward feedback that allows the Reinforcement Algorithm to learn which are the best behaviors that lead to maximum reward. This reward feedback is known as a reinforcement signal.

Let's look at some of the popular Machine Learning algorithms that are based on specific types of Machine Learning.

Supervised Machine Learning includes Regression and Classification algorithms. Some of the more popular algorithms in these categories are:

1. Linear Regression Algorithm

The Linear Regression Algorithm provides the relation between an independent and a dependent variable. It demonstrates the impact on the dependent variable when the independent variable is changed in any way. So the independent variable is called the explanatory variable and the dependent variable is called the factor of interest. An example of the Linear Regression Algorithm usage is to analyze the property prices in the area according to the size of the property, number of rooms, etc.

2. Logistic Regression Algorithm

The Logistic Regression Algorithm deals in discrete values whereas the Linear Regression Algorithm handles predictions in continuous values. This means that Logistic Regression is a better option for binary classification. An event in Logistic Regression is classified as 1 if it occurs and it is classified as 0 otherwise. Hence, the probability of a particular event occurrence is predicted based on the given predictor variables. An example of the Logistic Regression Algorithm usage is in medicine to predict if a person has malignant breast cancer tumors or not based on the size of the tumors.

3. Naive Bayes Classifier Algorithm

Naive Bayes Classifier Algorithm is used to classify data texts such as a web page, a document, an email, among other things. This algorithm is based on the Bayes Theorem of Probability and it allocates the element value to a population from one of the categories that are available. An example of the Naive Bayes Classifier Algorithm usage is for Email Spam Filtering. Gmail uses this algorithm to classify an email as Spam or Not Spam.

Unsupervised Machine Learning mainly includes Clustering algorithms. Some of the more popular algorithms in this category are:

1. K Means Clustering Algorithm

Let's imagine that you want to search the name "Harry" on Wikipedia. Now, "Harry" can refer to Harry Potter, Prince Harry of England, or any other popular Harry on Wikipedia! So Wikipedia groups the web pages that talk about the same ideas using the K Means Clustering Algorithm (since it is a popular algorithm for cluster analysis). K Means Clustering Algorithm in general uses K number of clusters to operate on a given data set. In this manner, the output contains K clusters with the input data partitioned among the clusters.

2. Apriori Algorithm

The Apriori Algorithm uses the if-then format to create association rules. This means that if a certain event 1 occurs, then there is a high probability that a certain event 2 also occurs. For example: IF someone buys a car, THEN there is a high chance they buy car insurance as well. The Apriori Algorithm generates this association rule by observing the number of people who bought car insurance after buying a car. For example, Google auto-complete uses the Apriori Algorithm. When a word is typed in Google, the Apriori Algorithm looks for the associated words that are usually typed after that word and displays the possibilities.

Advantages of Machine Learning :

1. Automation

Machine Learning is one of the driving forces behind automation, and it is cutting down time and human workload. Automation can now be seen everywhere, and the complex algorithm does the hard work for the user. Automation is more reliable, efficient, and quick. With the help of machine learning, now advanced computers are being designed. Now this advanced computer can handle several machine-learning models and complex algorithms. However, automation is spreading faster in the industry but, a lot of research and innovation are required in this field.

2. Scope of Improvement

Machine Learning is a field where things keep evolving. It gives many opportunities for improvement and can become the leading technology in the future. A lot of research and innovation is happening in this technology, which helps improve software and hardware.

3. Enhanced Experience in Online Shopping and Quality Education

Machine Learning is going to be used in the education sector extensively, and it will be going to enhance the quality of education and student experience. It has emerged in China; machine learning has improved student focus. In the e-commerce field, Machine Learning studies your search feed and give suggestion based on them. Depending upon search and browsing history, it pushes targeted advertisements and notifications to users.

4. Wide Range of Applicability

This technology has a very wide range of applications. Machine learning plays a role in almost every field, like hospitality, ed-tech, medicine, science, banking, and business. It creates more opportunity.

Challenges in Machine Learning :

1. Poor Quality of Data

Data plays a significant role in the machine learning process. One of the significant issues that machine learning professionals face is the absence of good quality data. Unclean and noisy data can make the whole process extremely exhausting. We don't want our algorithm to make inaccurate or faulty predictions. Hence the quality of data is essential to enhance the output. Therefore, we need to ensure that the process of data preprocessing which includes removing outliers, filtering missing values, and removing unwanted features, is done with the utmost level of perfection.

2. Underfitting of Training Data

This process occurs when data is unable to establish an accurate relationship between input and output variables. It simply means trying to fit in undersized jeans. It signifies the data is too simple to establish a precise relationship. To overcome this issue:

- *Maximize the training time*
- *Enhance the complexity of the model*
- *Add more features to the data*
- *Reduce regular parameters*
- *Increasing the training time of model*

3. Overfitting of Training Data

Overfitting refers to a machine learning model trained with a massive amount of data that negatively affect its performance. It is like trying to fit in Oversized jeans. Unfortunately, this is one of the significant issues faced by machine learning professionals. This means that the algorithm is trained with noisy and biased data, which will affect its overall performance. Let's understand this with the help of an example. Let's consider a model trained to differentiate between a cat, a rabbit, a dog, and a tiger. The training data contains 1000 cats,

1000 dogs, 1000 tigers, and 4000 Rabbits. Then there is a considerable probability that it will identify the cat as a rabbit. In this example, we had a vast amount of data, but it was biased; hence the prediction was negatively affected.

We can tackle this issue by:

- *Analyzing the data with the utmost level of perfection*
- *Use data augmentation technique*
- *Remove outliers in the training set*
- *Select a model with lesser features*

4. Machine Learning is a Complex Process

The machine learning industry is young and is continuously changing. Rapid hit and trial experiments are being carried on. The process is transforming, and hence there are high chances of error which makes the learning complex. It includes analyzing the data, removing data bias, training data, applying complex mathematical calculations, and a lot more. Hence it is a really complicated process which is another big challenge for Machine learning professionals.

5. Lack of Training Data

The most important task you need to do in the machine learning process is to train the data to achieve an accurate output. Less amount training data will produce inaccurate or too biased predictions. Let us understand this with the help of an example. Consider a machine learning algorithm similar to training a child. One day you decided to explain to a child how to distinguish between an apple and a watermelon. You will take an apple and a watermelon and show him the difference between both based on their color, shape, and taste. In this way, soon, he will attain perfection in differentiating between the two. But on the other hand, a machine-learning algorithm needs a lot of data to distinguish. For complex problems, it may even require millions of data to be trained. Therefore we need to ensure that Machine learning algorithms are trained with sufficient amounts of data.

6. Slow Implementation

This is one of the common issues faced by machine learning professionals. The machine learning models are highly efficient in providing accurate results, but it takes a tremendou

amount of time. Slow programs, data overload, and excessive requirements usually take a lot of time to provide accurate results. Further, it requires constant monitoring and maintenance to deliver the best output.

7. Imperfections in the Algorithm When Data Grows

So you have found quality data, trained it amazingly, and the predictions are really concise and accurate. Yay, you have learned how to create a machine learning algorithm!! But wait, there is a twist; the model may become useless in the future as data grows. The best model of the present may become inaccurate in the coming Future and require further rearrangement. So you need regular monitoring and maintenance to keep the algorithm working. This is one of the most exhausting issues faced by machine learning professionals.

Applications of Machine Learning:

1. Image Recognition

Image Recognition is one of the reasons behind the boom one could have experienced in the field of Deep Learning. The task which started from classification between cats and dog images has now evolved up to the level of Face Recognition and real-world use cases based on that like employee attendance tracking.

Also, image recognition has helped revolutionized the healthcare industry by employing smart systems in disease recognition and diagnosis methodologies.

2. Speech Recognition

Speech Recognition based smart systems like Alexa and Siri have certainly come across and used to communicate with them. In the backend, these systems are based basically on Speech Recognition systems. These systems are designed such that they can convert voice instructions into text.

One more application of the Speech recognition that we can encounter in our day-to-day life is that of performing Google searches just by speaking to it.

3. Recommender Systems

As our world has digitalized more and more approximately every tech giants try to provide customized services to its users. This application is possible just because of the recommended systems which can analyze a user's preferences and search history and based on that they can

recommend content or services to them. An example of these services is very common for example youtube. It recommends new videos and content based on the user's past search patterns. Netflix recommends movies and series based on the interest provided by users when someone creates an account for the very first time.

4.Fraud Detection

In today's world, most things have been digitalized varying from buying toothbrushes or making transactions of millions of dollars everything is accessible and easy to use. But with this process of digitization cases of fraudulent transactions and fraudulent activities have increased. Identifying them is not that easy but machine learning systems are very efficient in these tasks.

Due to these applications only whenever the system detects red flags in a user's activity than a suitable notification be provided to the administrator so, that these cases can be monitored properly for any spam or fraud activities.

5.Self Driving Cars

It would have been assumed that there is certainly some ghost who is driving a car if we ever saw a car being driven without a driver but all thanks to machine learning and deep learning that in today's world, this is possible and not a story from some fictional book. Even though the algorithms and tech stack behind these technologies are highly advanced but at the core it is machine learning which has made these applications possible.

The most common example of this use case is that of the Tesla cars which are well-tested and proven for autonomous driving.

6.Medical Diagnosis

If you are a machine learning practitioner or even if you are a student then you must have heard about projects like breast cancer Classification, Parkinson's Disease Classification, Pneumonia detection, and many more health-related tasks which are performed by machine learning models with more than 90% of accuracy. Not even in the field of disease diagnosis in human beings but they work perfectly fine for plant disease-related tasks whether it is to predict the type of disease it is or to detect whether some disease is going to occur in the future.

7. Stock Market Trading

Stock Market has remained a hot topic among working professionals and even students because if you have sufficient knowledge of the markets and the forces which drives them then you can make fortune in this domain. Attempts have been made to create intelligent systems which can predict future price trends and market value as well.

This can be considered as one of the applications of time series forecasting because stock price data is nothing but sequential data in which the time at which data has been taken is of utmost importance.

DBSCAN:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering algorithm in data mining and machine learning. Unlike centroid-based clustering methods like K-Means, DBSCAN doesn't require the user to specify the number of clusters beforehand. Instead, it identifies clusters based on the density of data points in a region. It's particularly useful for discovering clusters of arbitrary shapes and handling datasets with noise (outliers).

Key Concepts:

- 1. Core Point:** A point is considered a core point if it has at least a minimum number of points (MinPts) within a specified distance (ϵ , epsilon) around it, including itself. These points are at the heart of clusters.
- 2. Border Point:** A point that is within the ϵ distance of a core point but doesn't have enough neighbors (MinPts) to be a core point itself. These points lie on the edges of clusters.
- 3. Noise Point:** A point that is neither a core point nor a border point. These are outliers that don't belong to any cluster.
- 4. Epsilon (ϵ):** A distance parameter that defines the radius of the neighborhood around a point. It determines how close points must be to be considered part of the same cluster.
- 5. MinPts:** The minimum number of points required (including the point itself) within the ϵ -neighborhood to classify a point as a core point.
- 6. Directly Density-Reachable:** A point q is directly density-reachable from point p if q is within ϵ distance of p and p is a core point.

7.Density-Reachable: A point is density-reachable from another point if there's a chain of core points connecting them, where each consecutive point is directly density-reachable.

8.Density-Connected: Two points are density-connected if there exists a core point from which both are density-reachable.

DBSCAN processes a dataset of points (typically in a 2D or higher-dimensional space) and assigns them to clusters or marks them as noise. Here's the step-by-step process:

1. Input: The dataset (a set of points), the distance parameter ϵ , and the minimum number of points MinPts .

2. Initialization: Start with an unvisited point in the dataset and mark it as visited.

3. Core Point Check:

- Compute the number of points within the ϵ -neighborhood of the current point.
- If the number of neighbors is $\geq \text{MinPts}$, the point is a core point, and a new cluster is started.

4. Cluster Expansion:

- For a core point, add all points in its ϵ -neighborhood to the current cluster.
- Recursively check each of these points. If any are also core points (i.e., they have at least MinPts neighbors within ϵ), add their ϵ -neighborhoods to the cluster as well.
- This process continues until no more points can be added to the cluster (i.e., all density-reachable points are included).

5. Move to Next Point:

- If the current point is not a core point (i.e., it has fewer than MinPts neighbors), it's temporarily labeled as noise. (It might later be included as a border point if it's in the ϵ -neighborhood of a core point.)
- Move to the next unvisited point in the dataset and repeat steps 3–4.

6. Termination: The algorithm stops when all points in the dataset have been visited and assigned to a cluster or labeled as noise.

Advantages:

1. No Need to Specify Number of Clusters: Unlike K-Means, DBSCAN determines the number of clusters automatically based on data density.
2. Handles Arbitrary Shapes: It can find clusters of any shape, not just spherical ones.

3. Robust to Noise: Outliers are naturally identified as noise points.
4. Parameter-Driven: The algorithm's behavior can be tuned with ϵ and 'MinPts'.

Disadvantages:

1. Sensitive to Parameters: Choosing appropriate values for ϵ and 'MinPts' can be challenging and often requires domain knowledge or trial and error.
2. Struggles with Varying Density: If clusters have significantly different densities, DBSCAN may fail to identify them correctly with a single ϵ value.
3. Scalability: The basic implementation has a time complexity of $O(n^2)$ due to the need to compute distances between all pairs of points, though this can be improved to $O(n \log n)$ with spatial indexing (e.g., KD-trees) for low-dimensional data.
4. Curse of Dimensionality: In high-dimensional spaces, distance metrics become less meaningful, and DBSCAN's performance degrades unless dimensionality reduction is applied first.

Applications

1. Anomaly Detection: Identifying outliers (noise points) in datasets.
2. Geospatial Analysis: Clustering geographic locations (e.g., crime hotspots).
3. Image Segmentation: Grouping pixels based on color or intensity.
4. Biology: Clustering gene expression data.

DBSCAN is a powerful and flexible algorithm when you need to find clusters without knowing their number or shape in advance. Its ability to handle noise makes it particularly valuable in real-world datasets.

Artificial Intelligence

Artificial Intelligence (AI) refers to the development of computer systems of performing tasks that require human intelligence. AI aids, in processing amounts of data identifying patterns and making decisions based on the collected information. This can be achieved through techniques like Machine Learning, Natural Language Processing, Computer Vision and Robotics. AI encompasses a range of abilities including learning, reasoning, perception, problem solving, data analysis and language comprehension. The ultimate goal of AI is to create machines that can emulate capabilities and carry out diverse tasks, with enhanced efficiency and precision.

The field of AI holds potential to revolutionize aspects of our daily lives. Today, the amount of data in the world is so humongous that humans fall short of absorbing, interpreting, and making decisions of the entire data. This complex decision-making requires higher cognitive skills than human beings. This is why we're trying to build machines better than us, in these task. Another major characteristic that AI machines possess but we don't is repetitive learning. Let consider an example of how Artificial Intelligence is important to us. Data that is fed into the machines could be real-life incidents. How people interact, behave and react etc. So, in other words, machines learn to think like humans, by observing and learning from humans. That's precisely what is called Machine Learning which is a subfield of AI. Humans are observed to find repetitive tasks highly boring. Accuracy is another factor in which we humans lack. Machines have extremely high accuracy in the tasks that they perform. Machines can also take risks instead of human beings.

Applications of Artificial Intelligence

1. E-Commerce:

AI enhances user engagement by providing personalized recommendations based on search history and preferences.

AI chatbots offer instant customer support, reducing complaints and queries.

2. Healthcare:

AI aids in advanced analysis, visualization, and predictions.

It assists in diagnosing diseases, predicting patient outcomes, and recommending treatment plans.

AI-powered medical imaging helps detect anomalies and assists radiologists.

3. Natural Language Processing (NLP):

AI enables language translation, sentiment analysis, chatbots, and voice assistants.

It enhances communication and understanding between humans and machines.

4. **Personalization:**

AI tailors content, recommendations, and advertisements to individual preferences.

Examples include personalized streaming recommendations, e-commerce product suggestions, and social media content.

5. **Robotics:**

AI-driven robots perform tasks like cleaning (smart vacuum cleaners), social interaction (humanoid robots), and autonomous driving (self-driving cars).

6. **Financial Services:**

AI automates credit scoring, fraud detection, and investment strategies.

Robo-advisors provide personalized financial advice.

7. **Education:**

AI assists in personalized learning, adaptive assessments, and intelligent tutoring systems.

8. **Social Media:**

AI analyzes user behavior, recommends content, and detects fake news.

It powers features like facial recognition and personalized filters in photo apps.

9. **Entertainment:**

AI generates music, art, and movie scripts.

It enhances gaming experiences through procedural content generation and adaptive gameplay.

10. **Autonomous Vehicles:**

AI enables self-driving cars (e.g., Waymo) and improves safety on the roads

CHAPTER 5

SYSTEM DESIGN

5.1 System Architecture

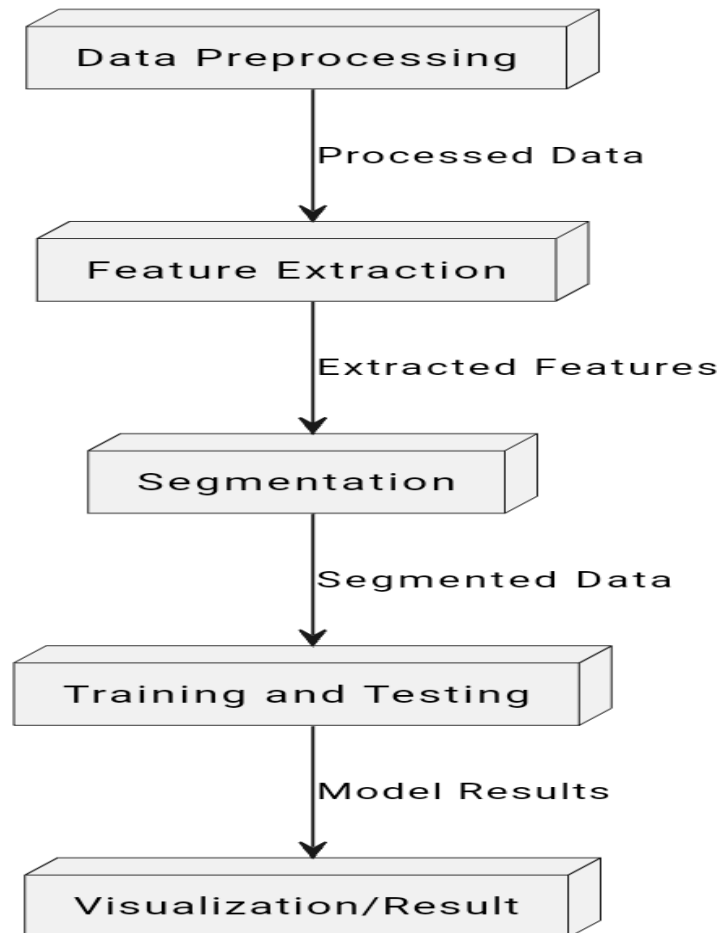


Fig 5.1: Architecture

Data Preprocessing

Data preprocessing ensures the clickstream data is clean and structured for analysis. It is used to handle missing values, eliminating duplicates, converting timestamps, and encoding categorical features like device type and platform. Numerical values such as session duration and total clicks are normalized and scaled to ensure consistency for clustering algorithms.

Feature Extraction

Key user behavior metrics are extracted, including session duration, total clicks, unique pages visited, bounce rate, and conversion actions. Other important features include entry/exit pages,

navigation patterns, and action sequences, which help in identifying user behavior trends. These structured features make the data ready for segmentation.

Segmentation

Users are segmented using K-Means and DBSCAN clustering. K-Means groups users into clusters based on similar browsing patterns, such as frequent buyers, casual visitors, and cart abandoners. DBSCAN identifies dense clusters and detects outliers like bots or fraudulent activities, improving the accuracy of segmentation.

Training and Testing

The K-Means model is trained on extracted features, determining the optimal number of clusters using the Silhouette Score. DBSCAN is tested with different distance parameters to detect meaningful clusters. Once trained, the models are applied to new data to predict user segments and assess clustering performance.

Visualization & Result

The segmented results are visualized using interactive charts and graphs, including bar charts for click sequences, funnel charts for conversion analysis, and scatter plots for cluster representation. Businesses gain insights into user behavior, drop-off points, and high-engagement users, enabling them to optimize marketing strategies and improve retention of customer.

5.2 Module Description

1. Data Upload & Validation Module

This module handles the process of uploading and validating clickstream data before analysis begins. It ensures that the data is structured correctly, free from errors, and contains all necessary information such as user interactions, session details, timestamps, and actions performed. Any inconsistencies, such as missing values or incorrect formats, are detected at this stage, preventing errors in later analysis. The primary purpose of this module is to provide a clean and reliable dataset, ensuring that further processing and insights are accurate and meaningful. If the uploaded file does not meet the expected structure, the system alerts the user to correct the data before proceeding.

2. Clickstream Analysis Module

This module is responsible for processing user interactions and extracting insights from

browsing behavior. It identifies key behavioral patterns such as the average number of pages visited per session, total session duration, and common navigation paths. By analyzing how users move through a website, businesses can determine which areas generate the most engagement and which sections may require improvement. This module also detects frequently clicked elements and actions performed by users, helping in identifying key points of interest on the website. By understanding these navigation patterns, businesses can optimize their websites to enhance user experience and reduce drop-off rates.

3. Conversion Funnel Analysis Module

This module focuses on analyzing the user journey through the conversion funnel, from the initial visit to the final purchase or desired action. It tracks the number of users at each stage of the funnel, identifying how many visitors progress and where most drop-offs occur. The funnel analysis helps businesses pinpoint weak points in the customer journey, such as high abandonment rates at the checkout stage or users leaving after viewing a product page. By understanding where users disengage, businesses can optimize these steps by improving content, simplifying navigation, or providing incentives to encourage conversions.

4. Session Metrics Module

This module analyzes session-based interactions to measure user engagement levels and browsing patterns. It identifies key metrics such as the most common entry and exit pages, session depth, and time spent on different sections of the website. By examining where users enter and leave, businesses can make data-driven decisions to improve retention, reduce bounce rates, and enhance content relevance. Additionally, this module helps identify frequent customer actions, such as product views and add-to-cart events, which are essential for optimizing sales funnels and marketing strategies.

5. Data Visualization Module

To simplify the interpretation of user behavior, this module generates interactive visualizations that display key insights in a clear and engaging manner. It presents data through bar charts, pie charts, and funnel diagrams, making it easier to identify trends and patterns. By visualizing metrics such as conversion rates, session activity, and click patterns, businesses can quickly assess website performance and identify areas that need improvement. Well-structured visual representations help decision-makers understand complex data more effectively, allowing them to implement targeted strategies to enhance user experience and increase conversions.

5.3 UML Diagrams

5.3.1 Use Case Diagram:

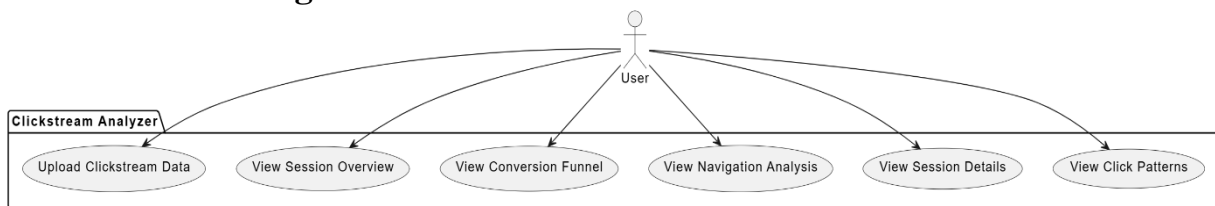


Fig 5.3.1: Use Case Diagram

As defined by and derived from a Use-case analysis, a use case diagram is a sort of behavioral diagram in the Unified Modeling Language (UML). Its goal is to provide a graphical overview of the functionality that a system offers by showing the actors, their objectives (expressed as use cases), and any interdependencies among those use cases. A use case diagram's primary objective is to illustrate which actors use the system and for what purposes. One can illustrate the roles that the system's actors play.

5.3.2 Class Diagram:

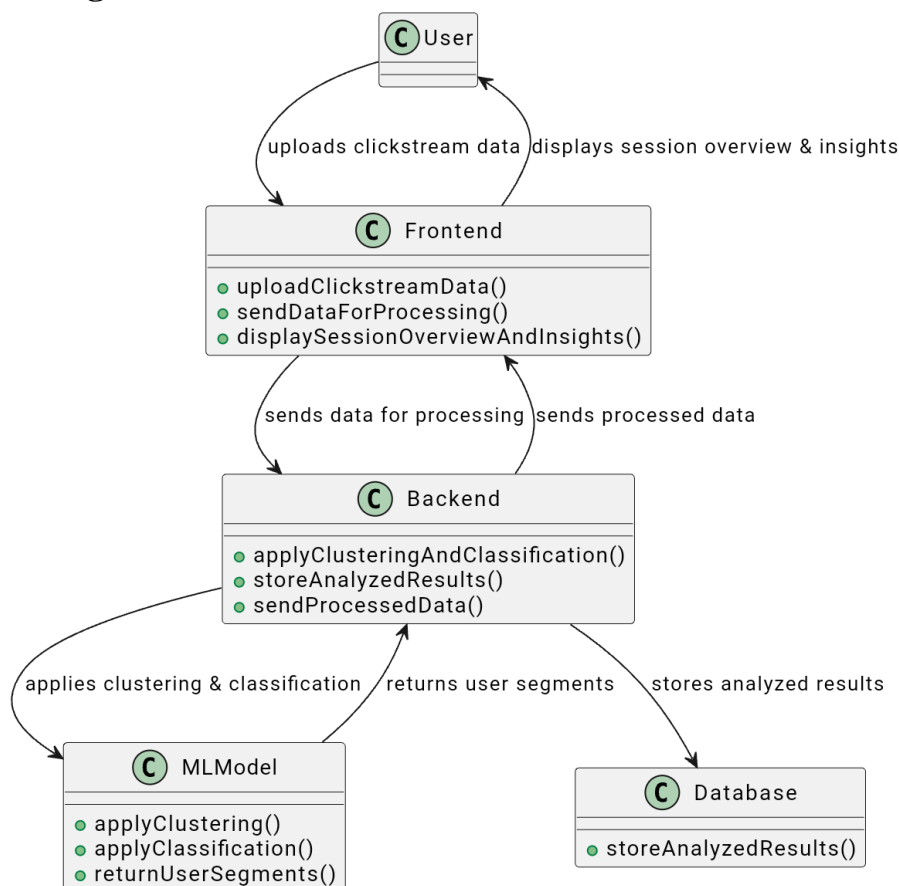


Fig 5.3.2: Class Diagram

A class diagram, as used in software engineering, is a sort of static structural diagram in the Unified Modeling Language (UML) that illustrates a system's classes, attributes, operations (or methods), and interactions between the classes. It indicates which class has the data.

5.3.3 Sequence Diagram:

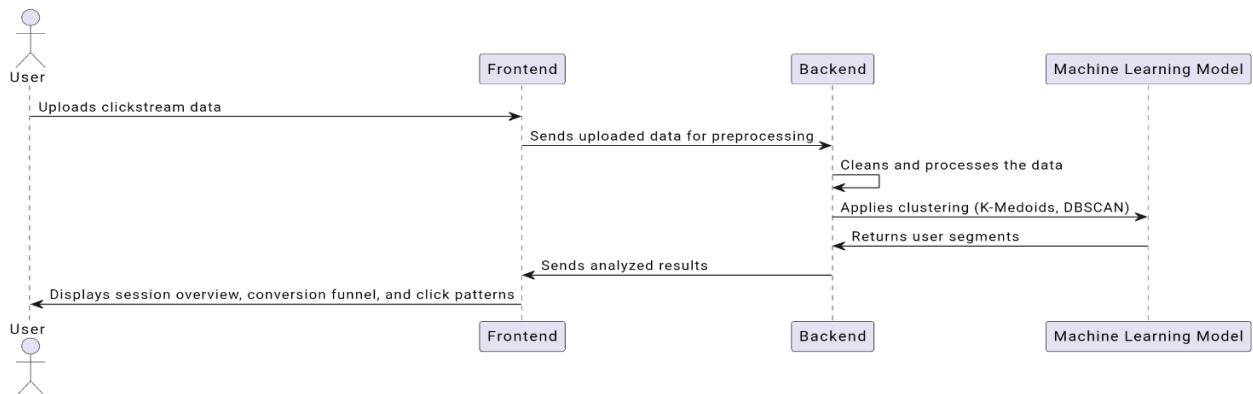


Fig 5.3.3: Sequence Diagram

In the Unified Modeling Language (UML), a sequence diagram is a type of interaction diagram that illustrates the relationships and sequence in which processes operate with one another. It is a Message Sequence Chart construct. Event diagrams, event situations, and timing diagrams are other names for sequence diagrams.

5.3.4 Collaboration Diagram:

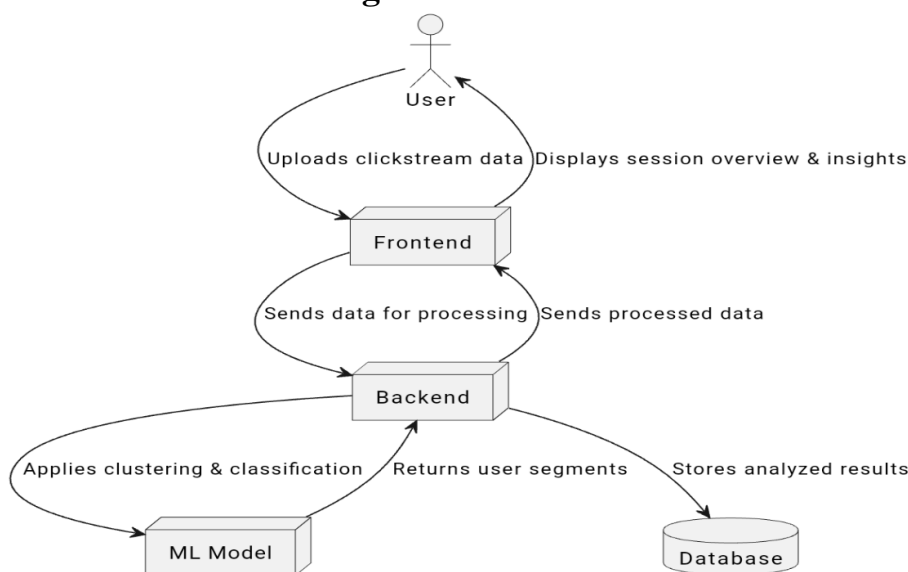


Fig 5.3.4: Collaboration Diagram

The method call sequence in a collaboration diagram is denoted by a numbering scheme, as demonstrated below. The sequence of the approaches is indicated by the number. The order management system that we are using to explain the collaboration diagram is the same one. The calls to the methods are akin to those in a sequence diagram. The cooperation diagram, on the other hand, depicts the object organization, but the sequence diagram does not explain it.

5.3.5 Deployment Diagram:



Fig 5.3.5: Deployment Diagram

The deployment view of a system is represented by a deployment diagram. It's connected to the schematic of components. due to the fact that deployment diagrams are used to deploy the components. Each node in a deployment diagram is an entity. Nodes are nothing more than the actual hardware that the program is deployed on.

5.3.6 Activity Diagram:

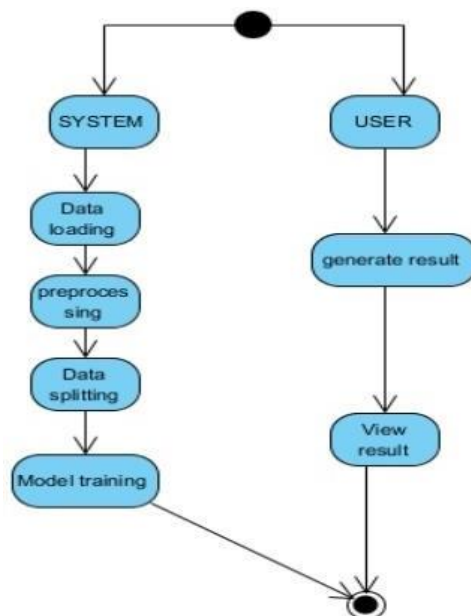


Fig 5.3.6: Activity Diagram

Activity diagrams are graphical depictions of workflows that allow for choice, iteration, and concurrency. They consist of sequential activities and actions. Activity diagrams in the Unified Modeling Language are a useful tool for describing the sequential business and operational workflows of system components. The whole flow of control is depicted in an activity diagram.

5.3.7 Component diagram:



Fig 5.3.7: Component Diagram

A component diagram, sometimes referred to as a UML component diagram, shows how the actual components of a system are wired and arranged. Component diagrams are frequently created to assist in modeling implementation specifics and ensure that all necessary functionalities of the system are addressed.

5.3.8 ER Diagram:

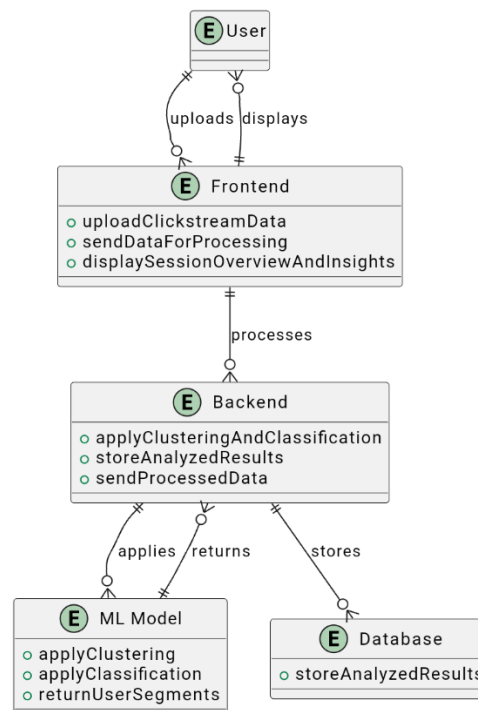


Fig 5.3.8: ER Diagram

An entity-relationship model, or ER model, uses an entity relationship diagram (ER Diagram)

to illustrate how a database is structured.

An entity-relationship diagram (ER diagram) illustrates this. A collection of related entities that may or may not contain attributes is called an entity set.

An ER diagram illustrates the entire logical structure of a database by illustrating the relationships between tables and their attributes. In terms of DBMS, an entity is a table or an attribute of a table in a database.

5.3.9 DFD Diagram:

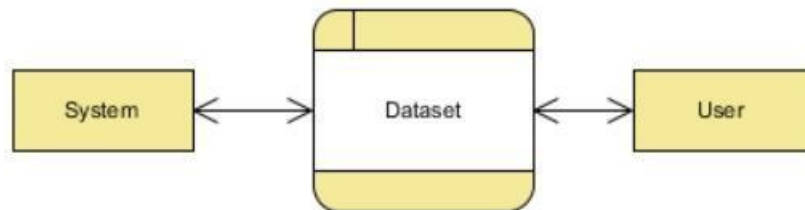


Fig 5.3.9: DFD Diagram

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

CHAPTER 6

PROPOSED METHODOLOGY

6.1 System

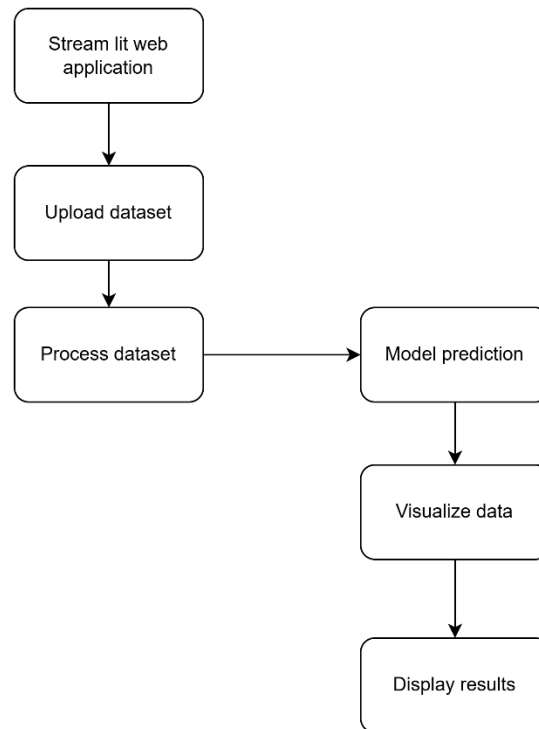


Fig 6.1: Project Flow

The above figure shows us the flow chart of the steps involved in the model. The proposed system for revealing consumer segments using clickstream data involves a multi-step process that leverages advanced data analytics and machine learning techniques to understand and categorize consumer behavior on e-commerce platforms. The system begins with the collection of clickstream data, which includes detailed logs of user interactions, such as page views, clicks, product searches, and purchase actions.

6.1.1 Data Base:

The dataset contains 100,000 rows of data, with each row representing a single transaction or event. The dataset has 11 columns, which are:

1. **User_ID:** A unique identifier for each user.
2. **Session_ID:** A unique identifier for each user session.

3. **Timestamp:** The timestamp of the transaction in the format "YYYY-MM-DD 6.HH:MM:SS".
4. **Page_Type:** The type of page the user visited, which can be "Product", "Category", "Home", or "Cart".
5. **Product_ID:** The ID of the product the user interacted with.
6. **Category:** The category of the product the user interacted with.
7. **Action:** The action the user took on the product, which can be "Add to Cart", "Purchase", or "View".
8. **Device_Type:** The type of device the user used to interact with the product, which can be "Desktop", "Mobile", or "Tablet".
9. **Location:** The location of the user, which can be a country or region.

6.1.2 Create Dataset:

In this module, the dataset containing all the details of the customers browsing the website. It is gathered by monitoring multiple customers all over the world visiting the website.

6.1.3 Pre-Processing:

Before the data can be used to train the model, they need to be pre-processed. This involves tasks such as re structuring and normalizing into a standardized format. Pre-processing ensures that the data are in a consistent format suitable for training the model effectively.

6.1.4 Training:

In the training module, the pre-processed training dataset is fed into the model. The model learns from these data to develop its predictive abilities. This is a critical step where the model learns to recognize patterns in the data.

6.1.5 Prediction:

Once the model is trained, it can be used to classify new, unseen data. In the classification module, the model takes an data as input and predicts the usage and category of the user by using clustering techniques.

CHAPTER 7

IMPLEMENTATION

7.1 Algorithm

Import Libraries:

Import necessary libraries: streamlit, pandas, plotly.express, plotly.graph_objects, and custom utility functions.

Set Up Streamlit Page:

Configure the Streamlit page with a title, icon, and layout.

Custom CSS Styling:

Define custom CSS styles for the application to enhance the visual appearance.

Display Title and Description:

Use Streamlit to display the title and a brief description of the application.

File Upload:

Create a file uploader widget to allow users to upload a CSV file containing clickstream data.

Data Loading and Validation:

If a file is uploaded:

Call the load_and_validate_data function to load the data into a DataFrame and validate its structure.

If there is an error during loading/validation, display an error message.

Data Analysis:

If the data is valid:

Call the analyze_clickstream function to analyze user behavior and extract key metrics.

Call the analyze_conversion_funnel function to analyze the conversion funnel and get funnel stages and rates.

Call the analyze_session_metrics function to analyze session metrics.

Display Key Metrics:

Create three columns to display key metrics:

Average Session Duration

Average Page Views

Conversion Rate

Visualize Conversion Funnel:

Create a funnel chart using Plotly to visualize the user journey through different stages of the conversion funnel.

Navigation Analysis:

Create two columns to display:

A bar chart of the most common navigation paths.

A bar chart of common action sequences taken by users.

Session Metrics Visualization:

Create two columns to display:

A pie chart of the top entry pages.

A pie chart of the top exit pages.

Display a bar chart showing the session depth distribution.

Click Patterns Visualization:

Create a DataFrame to analyze click patterns by category and page type.

Create a bar chart to visualize click patterns, distinguishing between electronics and other categories.

Handle No File Uploaded:

If no file is uploaded, display an informational message prompting the user to upload a CSV file.

Provide details about the expected data format.

7.2 Implementation of the Model

Data Collection & Preprocessing – The model starts by cleaning clickstream data, handling missing values, removing duplicates, and converting timestamps into a structured format. Categorical values like device type and platform are encoded, while numerical features such as session duration and clicks are normalized for consistency.

Feature Extraction – Key behavioral attributes are extracted, including session duration, total clicks, unique pages visited, bounce rate, and conversion actions. These features provide meaningful insights into user engagement and navigation patterns.

User Segmentation using Clustering – The model applies K-Means clustering to group users into distinct segments, such as frequent buyers, casual visitors, and cart abandoners. Simultaneously, DBSCAN is used to detect anomalies, such as bots or unusual browsing behaviors, by identifying outliers in the dataset.

Model Training & Evaluation – The model is trained on extracted features to identify optimal clusters, using techniques like the Elbow Method and Silhouette Score to validate clustering quality. The trained model is then tested on new data to assign user segments accurately.

Visualization & Results– The final clusters are visualized through interactive charts and graphs, including bar charts for click sequences, funnel charts for conversion analysis, and scatter plots for cluster representation. These insights help businesses optimize marketing strategies, personalize user experiences, and improve conversion rates.

Advantages of the Model:

1.Automated User Segmentation

The model efficiently groups users based on their browsing behavior, eliminating the need for manual analysis. This helps businesses understand different customer types, such as frequent buyers, casual visitors, and cart abandoners, improving decision-making.

2.Improved Personalization

By identifying unique user segments, businesses can deliver personalized recommendations, targeted marketing campaigns, and customized website experiences, leading to higher engagement and conversion rates.

3.Anomaly Detection

Unlike traditional clustering methods, DBSCAN detects outliers, such as bots, fraudulent users, or unusual browsing behaviors, ensuring data integrity and preventing misleading insights.

4.Scalability and Efficiency

The model can process large-scale clickstream data, making it suitable for businesses with high web traffic. Optimized preprocessing and clustering techniques enable efficient handling of millions of user interactions.

5.Actionable Business Insights

By analyzing session durations, click paths, and conversion funnels, the model provides insights into drop-off points, high-engagement areas, and underperforming pages, helping businesses refine their strategies.

6.Flexibility with Clustering Approaches

The combination of K-Means for structured segmentation and DBSCAN for density-based clustering makes the model more adaptable to varied user behaviors, capturing both common trends and anomalies.

7.Continuous Learning & Adaptability

The model can be retrained on new data, allowing businesses to keep up with changing user behavior, seasonal trends, and market shifts, ensuring long-term relevance.

Challenges:

1. High Dimensionality of Clickstream Data

Clickstream data contains multiple attributes, including user interactions, timestamps, device types, and navigation paths. Handling such high-dimensional data and selecting the most relevant features for clustering can be challenging, requiring effective feature selection and dimensionality reduction techniques like PCA.

2. Determining the Optimal Number of Clusters

K-Means requires specifying the number of clusters in advance, which may not always be clear. Using methods like the Elbow Method or Silhouette Score helps, but there is still a risk of over- or under-segmentation, affecting the quality of insights.

3. Handling Noisy and Incomplete Data

Clickstream data often contains missing values, duplicate entries, and inconsistent timestamps, which can impact clustering accuracy. Proper data cleaning and preprocessing are necessary to ensure high-quality input for the model.

4. Scalability Issues

Processing large-scale clickstream data for real-time segmentation can be computationally expensive. Efficient data handling techniques, such as batch processing and distributed computing, may be required to scale the model effectively.

5. Interpreting Clusters for Business Use

While clustering identifies user segments, understanding and labeling these clusters in a meaningful way for businesses can be difficult. Additional analysis is needed to correlate user segments with actual business outcomes, such as conversions or engagement levels.

6. Detecting Outliers and Anomalies

DBSCAN is effective for anomaly detection, but setting the right parameters (eps and min_samples) can be challenging. Too strict parameters might classify valid users as outliers, while loose parameters may fail to detect fraudulent activity.

7. Dynamic User Behavior

User behavior changes over time due to seasonal trends, promotions, or new website features. A static model may become outdated quickly, requiring continuous retraining and adaptation.

CHAPTER 8

SOFTWARE LIBRARIES

8.1 Python

What is Python :

Python is a programming language that is interpreted, object-oriented, and considered to be high-level too. Python is one of the easiest yet most useful programming languages which is widely used in the software industry. People use Python for Competitive Programming, Web Development, and creating software. Due to its easiest syntax, it is recommended for beginners who are new to the software engineering field. Its demand is growing at a very rapid pace due to its vast use cases in Modern Technological fields like Data Science, Machine learning, and Automation Tasks. For many years now, it has been ranked among the top Programming languages.

Python is a set of instructions that we give in the form of a Programme to our computer to perform any specific task. It is a Programming language having properties like it is interpreted, object-oriented and it is high-level too. Due to its beginner-friendly syntax, it became a clear choice for beginners to start their programming journey. The major focus behind creating it is making it easier for developers to read and understand, also reducing the lines of code.

today's time, python is the most popular language among development professionals which was developed by Guido Van Rossum in 1991. Python is an open-source and high-level programming language that supports much functionality that makes it so popular among developers. Programming in Python is much easier in comparison to other programming languages like C++ and Java due to its easy syntax and flow. Python also has its self memory management system which makes this language stand out. Artificial Intelligence and Machine Learning are some trending technologies in which we can use the Python language

Advantages of Python :

1. **Presence of third-party modules:** Python has a rich ecosystem of third-party modules and libraries that extend its functionality for various tasks.

2. **Extensive support libraries:** Python boasts extensive support libraries like NumPy for numerical calculations and Pandas for data analytics, making it suitable for scientific and data-related applications.
3. **Open source and large active community base:** Python is open source, and it has a large and active community that contributes to its development and provides support.
4. **Versatile, easy to read, learn, and write:** Python is known for its simplicity and readability, making it an excellent choice for both beginners and experienced programmers.
5. **User-friendly data structures:** Python offers intuitive and easy-to-use data structures, simplifying data manipulation and management.
6. **High-level language:** Python is a high-level language that abstracts low-level details, making it more user-friendly.
7. **Dynamically typed language:** Python is dynamically typed, meaning you don't need to declare data types explicitly, making it flexible but still reliable.
8. **Object-Oriented and Procedural programming language:** Python supports both object-oriented and procedural programming, providing versatility in coding styles.
9. **Portable and interactive:** Python is portable across operating systems and interactive, allowing real-time code execution and testing.
10. **Ideal for prototypes:** Python's concise syntax allows developers to prototype applications quickly with less code.

11. **Highly efficient:** Python's clean design provides enhanced process control, and it has excellent text processing capabilities, making it efficient for various applications.
12. **Internet of Things (IoT) opportunities:** Python is used in IoT applications due to its simplicity and versatility.
13. **Interpreted language:** Python is interpreted, which allows for easier debugging and code development.

Disadvantages of Python :

1. **Performance:** Python is an interpreted language, which means that it can be slower than compiled languages like C or Java. This can be an issue for performance-intensive tasks.
2. **Global Interpreter Lock:** The Global Interpreter Lock (GIL) is a mechanism in Python that prevents multiple threads from executing Python code at once. This can limit the parallelism and concurrency of some application.
3. **Memory consumption:** Python can consume a lot of memory, especially when working with large datasets or running complex algorithms.
4. **Dynamically typed:** Python is a dynamically typed language, which means that the types of variables can change at runtime. This can make it more difficult to catch errors and can lead to bugs.
5. **Packaging and versioning:** Python has a large number of packages and libraries, which can sometimes lead to versioning issues and package conflict.

6. **Lack of strictness:** Python's flexibility can sometimes be a double-edged sword. While it can be great for rapid development and prototyping, it can also lead to code that is difficult to read and maintain.
7. **Steep learning curve:** While Python is generally considered to be a relatively easy language to learn, it can still have a steep learning curve for beginners, especially if they have no prior experience with programming.

History of Python :

Python is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application-based work in December of 1989 at Centrum Wiskunde & Informatica (CWI) which is situated in the Netherlands. It was started as a hobby project because he was looking for an interesting project to keep him occupied during Christmas. The programming language in which Python is said to have succeeded is ABC Programming Language, which had interfacing with the Amoeba Operating System and had the feature of exception handling. He had already helped create ABC earlier in his career and had seen some issues with ABC but liked most of the features. After that what he did was very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and created a good scripting language that had removed all the flaws. The inspiration for the name came from the BBC's TV Show – 'Monty Python's Flying Circus', as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he named it Python! He was the "Benevolent dictator for life" (BDFL) until he stepped down from the position as the leader on 12th July 2018. For quite some time he used to work for Google, but currently, he is working at Dropbox. The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy

was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released, it had more than enough capability to provide classes with inheritance, several core data types of exception handling and functions.

8.2 Operating System

Operating System lies in the category of system software. It basically manages all the resources of the computer. An operating system acts as an interface between the software and different parts of the computer or the computer hardware. The operating system is designed in such a way that it can manage the overall resources and operations of the computer. Operating System is a fully integrated set of specialized programs that handle all the operations of the computer. It controls and monitors the execution of all other programs that reside in the computer, which also includes application programs and other system software of the computer. Examples of Operating Systems are Windows, Linux, Mac OS, etc. An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. The operating system is the most important type of system software in a computer system.

The operating system helps in improving the computer software as well as hardware. Without OS, it became very difficult for any application to be user-friendly. The Operating System provides a user with an interface that makes any application attractive and user-friendly. The operating System comes with a large number of device drivers that make OS services reachable to the hardware environment. Each and every application present in the system requires the Operating System. The operating system works as a communication channel between system hardware and system software. The operating system helps an application with the hardware part without knowing about the actual hardware configuration. It is one of the most important parts of the system and hence it is present in every device, whether large or small device.

8.3 Pandas

Pandas is a powerful Python library for data manipulation and analysis. It provides data structures and functions to efficiently handle structured data, including tabular data such as spreadsheets and SQL tables.

Key Features:

- Data Structures: Pandas introduces two primary data structures:
- Series (1-dimensional labeled array): similar to a column in a spreadsheet.
- DataFrame (2-dimensional labeled data structure): similar to an Excel spreadsheet or a table in a relational database.
- Data Operations: Pandas provides various functions for filtering, sorting, grouping, merging, reshaping, and pivoting data.

8.4 Streamlit

Streamlit is an open-source Python library that allows you to create web applications for data science and machine learning. It enables you to turn your data scripts into interactive web apps with minimal code.

Key Features

- Easy to Use: Streamlit has a simple and intuitive API, making it easy to create web apps without extensive web development knowledge.
- Interactive Visualizations: Streamlit supports various visualization libraries, including Matplotlib, Plotly, and Altair, allowing you to create interactive visualizations.
- Real-time Updates: Streamlit enables real-time updates, making it ideal for applications that require live data updates.
- Customizable: Streamlit provides various customization options, including themes, layouts, and widgets.

8.5 Plotly

Plotly is a popular Python library for creating interactive, web-based visualizations. It supports a wide range of charts, including line plots, scatter plots, bar charts, histograms, and more.

Key Features:

- Interactive Visualizations: Plotly creates interactive visualizations that allow users to hover, zoom, and pan.
- Wide Range of Charts: Plotly supports over 40 different chart types, including 3D charts and maps.
- Customizable: Plotly provides various customization options, including colors, fonts, and layouts.

- Integration with Other Libraries: Plotly integrates well with other popular data science libraries in Python, including Pandas, NumPy, and Scikit-learn.

8.6 Scikit-learn (Sklearn)

Scikit-learn, also known as Sklearn, is a widely-used Python library for machine learning. It provides a range of algorithms for classification, regression, clustering, and other tasks, along with tools for model selection, data preprocessing, and feature selection.

Key Features:

- Algorithms: Sklearn provides a wide range of algorithms for various machine learning tasks, including:
 - Classification: logistic regression, decision trees, random forests, support vector machines (SVMs)
 - Regression: linear regression, ridge regression, Lasso, elastic net
 - Clustering: k-means, hierarchical clustering, DBSCAN
- Model Selection: Sklearn provides tools for model selection, including cross-validation, grid search, and random search.
- Data Preprocessing: Sklearn provides tools for data preprocessing, including data normalization, feature scaling, and encoding categorical variables.
- Feature Selection: Sklearn provides tools for feature selection, including recursive feature elimination and mutual information-based feature selection.

8.7 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc. Windows, Linux, and macOS distributions have Matplotlib and most of its dependencies as wheel packages. Run the following command to install the Matplotlib package. But before that make sure Python and PIP are already installed on a system.

CHAPTER 9

SYSTEM STUDY AND TESTING

9.1 Feasibility Study

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ Economical feasibility
- ◆ Technical feasibility
- ◆ Social feasibility

9.1.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

9.1.2 Technical Feasibility

This study is done to check the specialized practicality, or at least, the specialized prerequisites of the framework. Any framework created should not have a popularity on the accessible specialized assets. This will prompt high requests on the accessible specialized assets. This will prompt high requests being put on the client. The created framework should have a humble prerequisite, as just insignificant or invalid changes are expected for executing this framework.

9.1.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

9.2 System Testing

The reason for testing is to find mistakes. Testing is the most common way of attempting to find each possible shortcoming or shortcoming in a work item. It gives a method for really taking a look at the usefulness of parts, sub-congregations, gatherings as well as a completed item. It is the most common way of practicing programming with the expectation of guaranteeing that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

9.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

9.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components

is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.2.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.2.4 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for

testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

9.2.5 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

9.2.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed

Test Cases:

Input	Output	Result
Input image	Output will be the segmentation	Success

Table 9.1: Test cases

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Read the dataset.	Dataset path.	Dataset need to read successfully.	Dataset fetched successfully.	P
2	Performing pre-processing on the dataset	Pre-processing part takes place	Pre-processing should be performed on dataset	Pre-processing successfully completed.	P
3	Model Building	Model Building for the clean data	Need to create model using required algorithms	Model Created Successfully.	P
4	visualization	Input insights, patterns provided.	Output should be segmentation of users	Model visualized successfully	P

CHAPTER 10

SOURCE CODE

App.py

```
import streamlit as st
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from utils import (
    load_and_validate_data,
    analyze_clickstream,
    analyze_conversion_funnel,
    analyze_session_metrics
)

# Page configuration
st.set_page_config(
    page_title="Clickstream Analyzer",
    page_icon="🔍",
    layout="wide"
)

# Custom CSS
st.markdown("""
<style>
.main {
    padding: 1rem;
}
.stMetric {
    background-color: #f0f2f6;
    padding: 1rem;
    border-radius: 0.5rem;
}
.plot-container {
    background-color: white;
    border-radius: 0.5rem;
    padding: 1rem;
    margin: 1rem 0;
}
</style>
""", unsafe_allow_html=True)

# Title and description
st.title("🔍 Clickstream Analyzer")
st.markdown("""
```

```
Analyze user behavior patterns from clickstream data.  
Upload your data to get insights about user navigation, conversion funnels, and session metrics.  
""")
```

```
# File upload
```

```
uploaded_file = st.file_uploader("Upload Clickstream Data (CSV)", type=['csv'])
```

```
if uploaded_file is not None:
```

```
    # Load and validate data
```

```
    df, error = load_and_validate_data(uploaded_file)
```

```
    if error:
```

```
        st.error(error)
```

```
    else:
```

```
        # Analyze clickstream
```

```
        clickstream_analysis = analyze_clickstream(df)
```

```
        funnel_stages, funnel_rates = analyze_conversion_funnel(df)
```

```
        session_metrics = analyze_session_metrics(df)
```

```
    # Display key metrics
```

```
    st.header("Session Overview")
```

```
    col1, col2, col3 = st.columns(3)
```

```
    with col1:
```

```
        st.metric("Avg Session Duration", f"{clickstream_analysis['avg_session_duration']:.1f}s")
```

```
    with col2:
```

```
        st.metric("Avg Page Views", f"{clickstream_analysis['avg_page_views']:.1f}")
```

```
    with col3:
```

```
        st.metric("Conversion Rate", f"{funnel_rates['Purchase']:.1f}%")
```

```
    # Conversion Funnel
```

```
    st.header("Conversion Funnel")
```

```
    fig_funnel = go.Figure(go.Funnel(
```

```
        y=list(funnel_stages.keys()),
```

```
        x=list(funnel_stages.values()),
```

```
        textposition="inside",
```

```
        textinfo="value+percent initial"
```

```
    ))
```

```
    fig_funnel.update_layout(title_text="User Journey Funnel")
```

```
    st.plotly_chart(fig_funnel, use_container_width=True)
```

```
    # Common Navigation Paths
```

```
    st.header("Navigation Analysis")
```

```
    col1, col2 = st.columns(2)
```

```
    with col1:
```

```
        fig_paths = px.bar(
```

```
            x=list(clickstream_analysis['common_paths'].keys()),
```

```
            y=list(clickstream_analysis['common_paths'].values()),
```

```

        title='Most Common Navigation Paths',
        labels={'x': 'Path', 'y': 'Frequency'}
    )
    st.plotly_chart(fig_paths, use_container_width=True)

with col2:
    fig_actions = px.bar(
        x=list(clickstream_analysis['action_sequences'].keys()),
        y=list(clickstream_analysis['action_sequences'].values()),
        title='Common Action Sequences',
        labels={'x': 'Action Sequence', 'y': 'Frequency'}
    )
    st.plotly_chart(fig_actions, use_container_width=True)

# Session Metrics
st.header("Session Details")

# Entry and Exit Pages
col1, col2 = st.columns(2)

with col1:
    fig_entry = px.pie(
        values=list(session_metrics['top_entry_pages'].values()),
        names=list(session_metrics['top_entry_pages'].keys()),
        title='Top Entry Pages'
    )
    st.plotly_chart(fig_entry, use_container_width=True)

with col2:
    fig_exit = px.pie(
        values=list(session_metrics['top_exit_pages'].values()),
        names=list(session_metrics['top_exit_pages'].keys()),
        title='Top Exit Pages'
    )
    st.plotly_chart(fig_exit, use_container_width=True)

# Session Depth Distribution
fig_depth = px.bar(
    x=list(session_metrics['depth_distribution'].keys()),
    y=list(session_metrics['depth_distribution'].values()),
    title='Session Depth Distribution',
    labels={'x': 'Number of Pages', 'y': 'Number of Sessions'}
)
st.plotly_chart(fig_depth, use_container_width=True)

# Click Patterns
st.header("Click Patterns")

# Create color map for categories

```



```

click_pattern_data = pd.DataFrame({
    'pattern': list(clickstream_analysis['click_patterns'].keys()),
    'clicks': list(clickstream_analysis['click_patterns'].values())
})
click_pattern_data['is_electronics'] = click_pattern_data['pattern'].str.startswith('Electronics')

fig_clicks = px.bar(
    click_pattern_data,
    x='pattern',
    y='clicks',
    color='is_electronics',
    title='Click Patterns by Category and Page Type',
    labels={
        'pattern': 'Category - Page Type',
        'clicks': 'Number of Clicks',
        'is_electronics': 'Electronics Category'
    },
    color_discrete_map={True: '#ff7f0e', False: '#1f77b4'}
)

fig_clicks.update_layout(
    showlegend=False,
    xaxis_tickangle=-45,
    height=500
)

st.plotly_chart(fig_clicks, use_container_width=True)

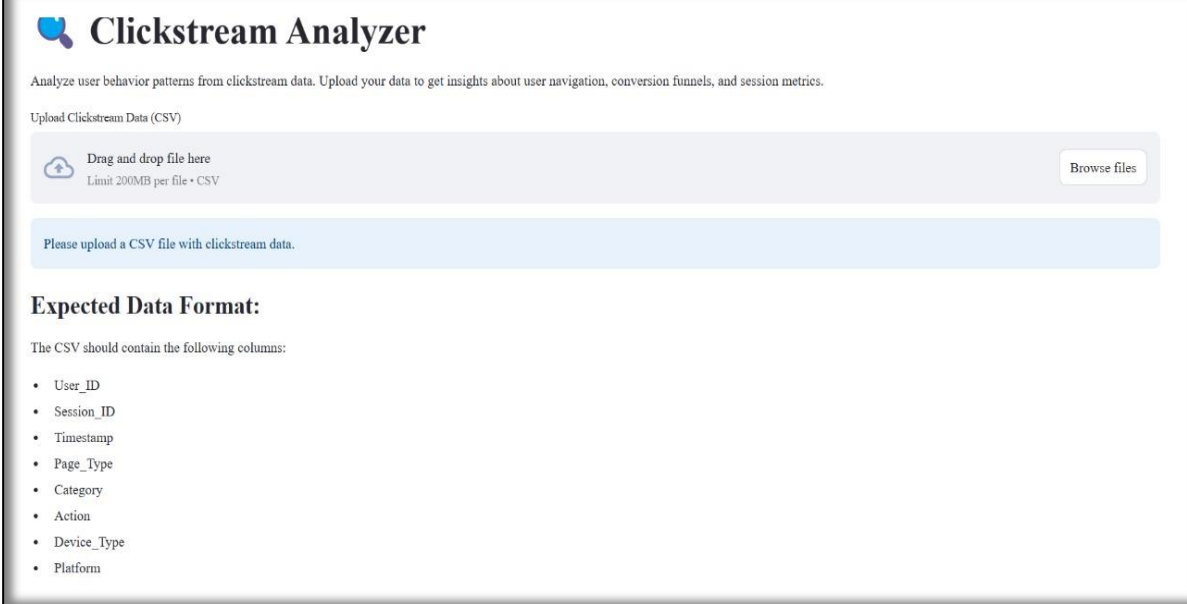
else:
    st.info("Please upload a CSV file with clickstream data.")
    st.markdown("""
    ### Expected Data Format:
    The CSV should contain the following columns:
    - User_ID
    - Session_ID
    - Timestamp
    - Page_Type
    - Category
    - Action
    - Device_Type
    - Platform
    """)

```

CHAPTER 11

OUTPUT SCREENSHOTS

Upload Page:



The screenshot shows the 'Clickstream Analyzer' interface. At the top, there's a logo and the title 'Clickstream Analyzer'. Below it, a subtitle reads: 'Analyze user behavior patterns from clickstream data. Upload your data to get insights about user navigation, conversion funnels, and session metrics.' The main section is titled 'Upload Clickstream Data (CSV)' and features a large light blue box with the text 'Drag and drop file here' and 'Limit 200MB per file • CSV'. To the right of this box is a 'Browse files' button. Below the upload area, a light blue bar contains the text 'Please upload a CSV file with clickstream data.' Underneath this, the section 'Expected Data Format:' is followed by the text 'The CSV should contain the following columns:'. A bulleted list follows, specifying the required columns: User_ID, Session_ID, Timestamp, Page_Type, Category, Action, Device_Type, and Platform.

Fig 11.1 Clickstream Analyzer

This is the input page which takes the data set as an input. The data which is to be processed has to be in the designated format otherwise errors are shown.

Output page:



Fig 11.2 Output page

The output page shows the data after it is processed and the average usage time of each customer and the time they spent on a particular object. Then it also shows the plot of the users along with their buying capacity and usage time.

The output page is user friendly and the data visualization can be visually attractive and informative to the users.

The output page consists of the following visualization steps in which each step visualizes different segments and details of the visualized data and the following steps are present in the output page of the website:

1) Conversion Funnel

2) Navigation Analysis

3) Session Details

4) Session Depth Distribution

5) Click Patterns

To simplify the interpretation of user behavior, this module generates interactive visualizations that display key insights in a clear and engaging manner. It presents data through bar charts, pie charts, and funnel diagrams, making it easier to identify trends and patterns. By visualizing metrics such as conversion rates, session activity, and click patterns, businesses can quickly assess website performance and identify areas that need improvement. Well-structured visual representations help decision-makers understand complex data more effectively, allowing them to implement targeted strategies to enhance user experience and increase conversions.

The segmented results are visualized using interactive charts and graphs, including bar charts for click sequences, funnel charts for conversion analysis, and scatter plots for cluster representation. Businesses gain insights into user behavior, drop-off points, and high-engagement users, enabling them to optimize marketing strategies and improve retention of customer.

CHAPTER 12

RESULTS AND DISCUSSION

12.1 RESULT & DISCUSSION

1. Conversion Funnel

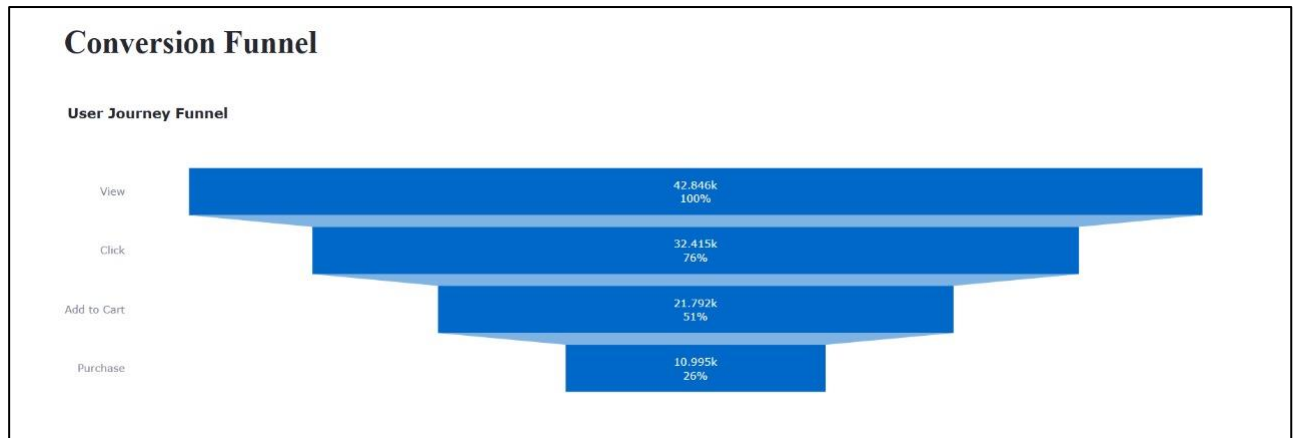


Fig 12.1.1 Conversion Funnel

This funnel visualizes the drop-off rates at different stages of the user journey:

View (100%): 42.846k users viewed a page.

Click (76%): 32.415k users interacted by clicking.

Add to Cart (51%): 21.792k users added items to their cart.

Purchase (26%): 10.995k users completed the transaction.

The significant drop-off from “Add to Cart” to “Purchase” indicates potential friction in the checkout process.

2. Navigation Analysis

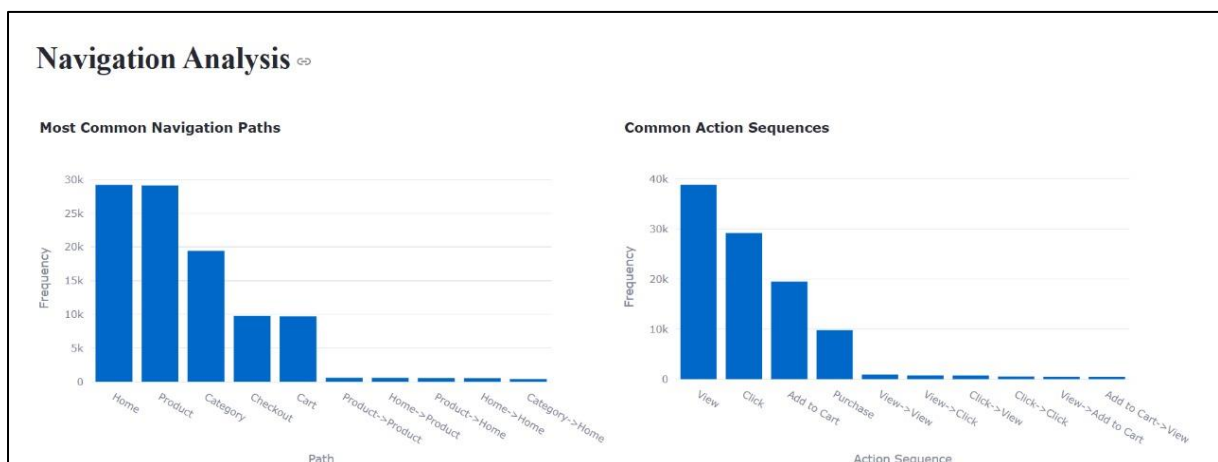


Fig 12.1.2: Navigation Analysis

This section highlights how users navigate the site.

Most Common Navigation Paths:

Home and Product pages are the most visited. Category and Checkout pages see fewer visits.

Less common paths include transitions between specific pages.

Common Action Sequences:

Users typically start with viewing a page, then clicking, followed by adding items to the cart, and finally purchasing.

Some users revisit pages before taking further actions.

3. Session Details

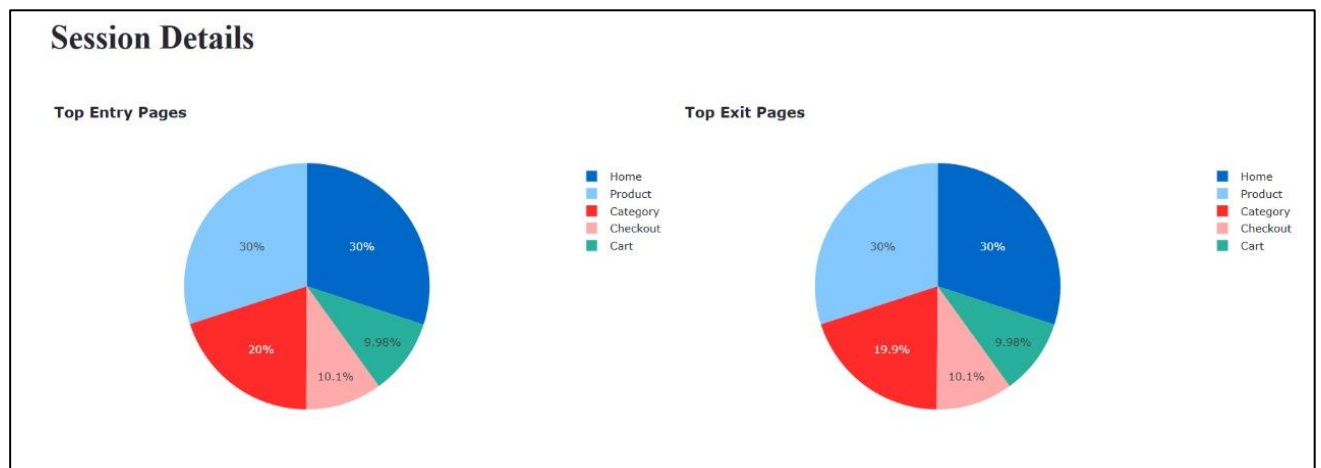


Fig 12.1.3: Session Details

This provides insights into where users enter and exit the website.

Top Entry Pages:

Home (30%) and Product (30%) pages are the most common entry points.

Category (20%), Checkout (10.1%), and Cart (9.98%) have lower entry rates.

Top Exit Pages:

Home and Product pages also have the highest exit rates.

Category and Checkout pages also see notable user exits, potentially indicating abandonment before completing a purchase.

4. Session Depth Distribution

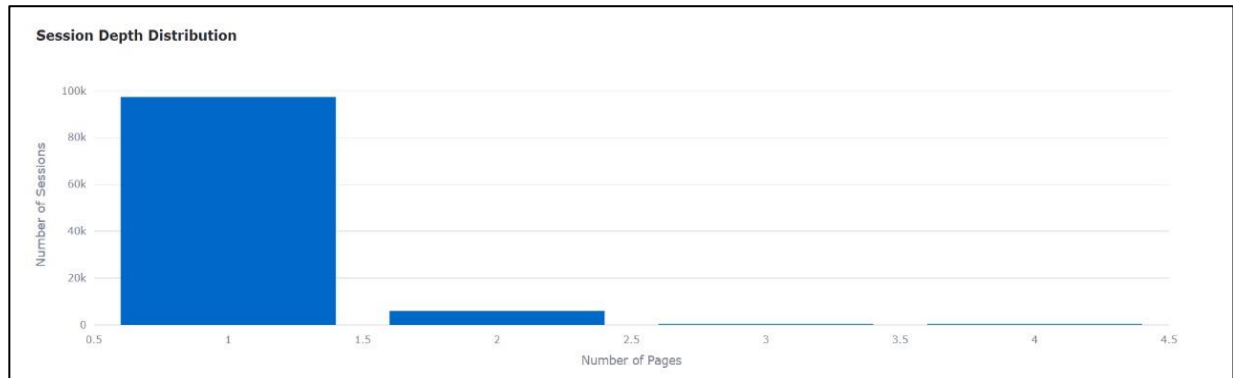


Fig 12.1.4 Session Depth Distribution

Number of Pages Visited Per Session:

The majority of sessions consist of just **one page view**, meaning many users leave without further interaction.

Only a small percentage of users visit multiple pages, suggesting engagement issues.

5. Click Patterns

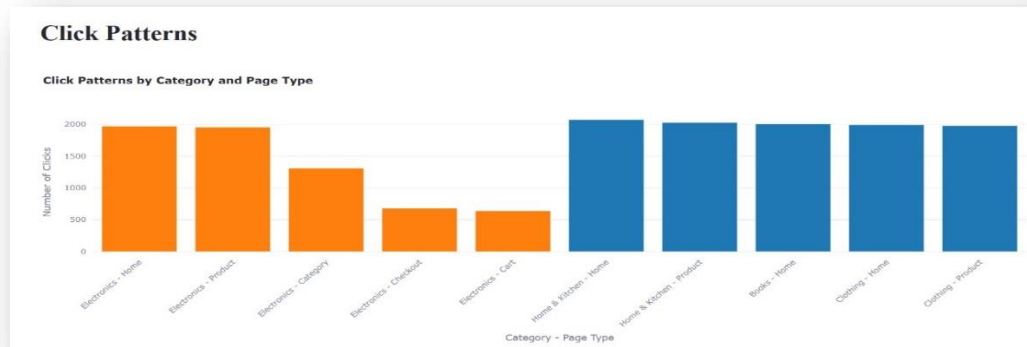


Fig 12.1.5: Click Patterns

This visualization shows where users click the most based on **category and page type**.

1. **Electronics (Home & Product)** have the highest number of clicks.

2. **Electronics (Category, Checkout, and Cart)** receive fewer clicks.

3. Other categories, such as **Home & Kitchen, Books, and Clothing**, also see significant interactions.

CHAPTER 13

CONCLUSION AND FUTURE WORK

13.1 Conclusion

This study developed an innovative system for identifying consumer segments using clickstream data and machine learning techniques. By applying K-Means and DBSCAN clustering, we effectively categorized user behaviors, uncovering valuable insights into shopping patterns. These findings enable businesses to personalize user experiences, optimize marketing strategies, and enhance customer engagement. The system's ability to adapt and learn from new data ensures its relevance in the evolving digital landscape, making it a powerful tool for e-commerce platforms seeking data-driven decision-making and long-term customer retention. The segmented results are visualized using interactive charts and graphs, including bar charts for click sequences, funnel charts for conversion analysis, and scatter plots for cluster representation. Businesses gain insights into user behavior, drop-off points, and high-engagement users, enabling them to optimize marketing strategies and improve retention of customer.

13.2 Future Enhancements

Future enhancements of this model focus on increasing efficiency, accuracy, and scalability to provide even more actionable insights. The integration of AI-driven recommendation systems will allow businesses to deliver highly personalized user experiences based on browsing behavior. Implementing real-time data processing will enable instant insights, allowing businesses to respond dynamically to user actions. Enhancing predictive analytics will help forecast user behavior trends, optimizing marketing strategies and reducing churn rates. Additionally, cross-platform tracking will ensure seamless user segmentation across web, mobile, and IoT environments. The development of advanced fraud detection mechanisms will further improve security by identifying suspicious activities and anomalies in real time. Cloud-based deployment and scalability solutions will enhance the model's ability to handle large datasets efficiently. These enhancements will ensure that the system remains adaptive, intelligent, and highly effective in supporting data-driven business decisions.

REFERENCES

1. Gumber, M., Jain, A., & Amutha, A. L. (2021, May). Predicting Customer Behavior by Analyzing Clickstream Data. In 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP) (pp. 1-6). IEEE.
2. Wen, Z., Lin, W., & Liu, H. (2023). Machine-learning-based approach for anonymous online customer purchase intentions using clickstream data. *Systems*, 11(5), 255. <https://doi.org/10.3390/systems11050255>
3. Zavali, M., Lacka, E., & De Smedt, J. (2021). Shopping hard or hardly shopping: Revealing consumer segments using clickstream data. *IEEE Transactions on Engineering Management*, 70(4), 1353-1364.
4. Anitha, M., Ramya, K. B., & Zeenath, MD. (2024). Predicting online shopping behavior through clickstream analysis. *SRK Institute of Technology*, 20.
5. Lee, H. (2024). Interest-Based E-Commerce and users' purchase intention on social network platforms. *IEEE Access*, 12, 87451–87466. <https://doi.org/10.1109/access.2024.3417440>
6. Cachero-Martínez, S., & Vázquez-Casielles, R. (2021). Building consumer loyalty through e-shopping experiences: The mediating role of emotions. *Journal of Retailing and Consumer Services*, 60, 102481. <https://doi.org/10.1016/j.jretconser.2021.102481>
7. Koehn, D., Lessmann, S., & Schaal, M. (2020). Predicting online shopping behavior from clickstream data using deep learning. *Expert Systems with Applications*, 150, 113342. <https://doi.org/10.1016/j.eswa.2020.113342>
8. Somya, R., Winarko, E., & Privanta, S. (2021, September). A novel approach to collect and analyze market customer behavior data on online shop. In 2021 2nd International Conference on Innovative and Creative Information Technology (ICITech) (pp. 151-156). IEEE.
9. Noviantoro, T., & Huang, J. P. (2021). Applying data mining techniques to investigate online shopper purchase intention based on clickstream data. *Rev Bus Account Financ*, 1(2), 130-159.
10. Santhosh Sv & Dr. Basava Rajappa B. (2022). A Study on Consumer Behaviour Towards Online Shopping. *International Journal of Research in Analytical Reviews (IJRAR)*, 9(1), February 2022.
11. Siddhant Sharma, Akhilesh A. Wao, "Customer Behaviour Analysis in E-Commerce using Machine Learning Approach: A Survey", *International Journal of Scientific Research in*

12. Mofokeng, T. E. (2021). The impact of online shopping attributes on customer satisfaction and loyalty: Moderating effects of e-commerce experience. *Cogent Business & Management*, 8(1), 1968206. <https://doi.org/10.1080/23311975.2021.1968206>
13. Chen, Y., & Yao, S. (2017). Sequential search with refinement: Model and application with click-stream data. *Management Science*, 63(12), 4345-4365.
14. Hyun, H., Thavisay, T., & Lee, S. H. (2021). Enhancing the role of flow experience in social media usage and its impact on shopping. *Journal of Retailing and Consumer Services*. <https://doi.org/10.1016/j.jretconser.2021.102492>
15. Chaffey, D., Ellis-Chadwick, F., Johnston, K., & Mayer, R. (2019). *Digital marketing: Strategy, implementation and practice*.
16. Pal, G., Atkinson, K., & Li, G. (2021). Real-time user clickstream behavior analysis based on apache storm streaming. *Electronic Commerce Research*, 23, 1829–1859. <https://doi.org/10.1007/s10660-021-09518-4>
17. Requena, B., Cassani, G., Tagliabue, J., Greco, C., & Lacasa, L. (2020). Shopper intent prediction from clickstream e-commerce data with minimal browsing information. *Scientific reports*, 10(1), 16983.
18. Kumar, A., Salo, J., & Li, H. (2019). Stages of User Engagement on Social Commerce Platforms: Analysis with the Navigational Clickstream Data. *International Journal of Electronic Commerce*, 23(2), 179–211. <https://doi.org/10.1080/10864415.2018.1564550>
19. C. Kaushal and H. Singh, “Comparative study of recent sequential pattern mining algorithms on web clickstream data,” in 2015 IEEE Power, Communication and Information Technology Conference (PCITC), Conference Proceedings, pp. 652–656C.
20. Dextras-Romagnino, K., & Munzner, T. (2019). Segmentifier: interactive refinement of clickstream data. *Computer Graphics Forum*, 38(3), 623–634. <https://doi.org/10.1111/cgf.13715>

