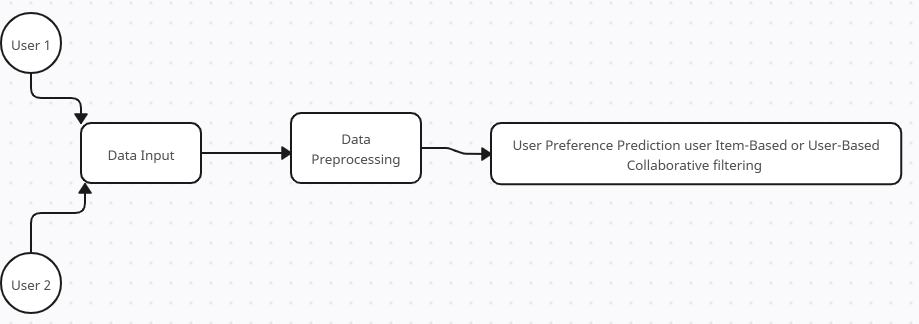
**CHAPTER THREE**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 Analysis of Existing System**

Existing users’ preference prediction models use the traditional collaborative filtering model. This model predicts user preferences by leveraging similarities between users or items, and collects data from user behavior and items interacted with, through clicks, views, likes, and comments. It relies on user-item interaction matrices and data sparsity reduces the model’s predictive accuracy, as it struggles to find enough common interactions to make reliable predictions (Su 2009). This model relies on explicit feedback like ratings to infer user preferences, it assumes static user preferences and does not account for context or changes in user interests over time.

**3.1.1 Architecture of the Existing System**

****

**Fig 3.1 Architecture of the existing system (Drawn using Creatly)**

**Components of the Architecture**

1. User: these people interact with the system and create traffic.
2. Data Input: When users interact with the system, they create data in the system.
3. Data Processing: Data created are collected and processed in the system to train the model.
4. User Preference Prediction (Item-based or User-Based collaborative filtering): After the data has been processed, it is used to train a model using the Collaborative filtering Machine Learning technique. This technique uses either user-based collaborative filtering or item-based collaborative filtering.

**3.1.2 Constraints of the Existing System**

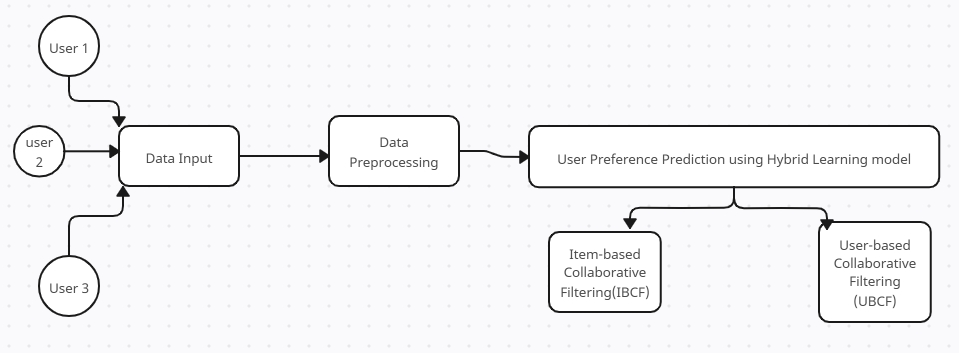
1. Cold Start Problem: they struggle with new users or items, as this model requires historical data to make recommendations.
2. Data Sparsity: users interact with only a small fraction of available items, making the datasets sparse.
3. Scalability issues: as numbers of users and items grow the model face scalability issues, reducing the system’s responsiveness in real-time applications.

**3.6 Analysis of the Proposed System**

The proposed system incorporates Hybrid models that combines collaborative filtering with content-based filtering, this helps overcome data sparsity and cold-start issues by integrating item attributes and contextual data with collaborative signals. The new system tailors the platform to meet individual user needs, creating a more enjoyable and seamless experience. In environments such as e-commerce, streaming services, or social media, personalized recommendations are key to engaging users. The system’s machine learning algorithms can analyze user behavior, preferences, and interaction history to recommend relevant products, services, or content.

Enhanced user preference prediction leads to higher levels of engagement and long-term user retention. The system becomes more attuned to user preferences, fostering a sense of connection and value. Personalized recommendations encourage users to explore more content or products, extending their time spent interacting with the system, the system can suggest new features or sections (e.g., specific categories, forums, or services) that align with the user’s preferences, increasing engagement across the platform. Personalization fosters loyalty, as users feel that the system understands their preferences and anticipates their needs.

**3.2.1 Architecture of the Proposed System**

****

**Fig 3.2 Architecture of the Proposed System (Drawn in Creatly)**

**Components of the architecture**

1. User: Individuals who interact with the system, through clicks, views, etc and generate traffic.
2. Data Input and Collection: User actions create data in the system, which is collected and stored in the form of cookies.
3. Data Preprocessing: Data collected are processed and used to train the model.
4. User Preference Prediction: The processed data is used to train the model, using Hybrid Learning Machine Learning Technique, which incoporate both Item-based and User-based Collaborative Filtering.

**3.2.2 Justification of the Proposed System**

1. Resolves Cold start issues
2. Handles data effectively, taking care of data sparsity.
3. Improved scalability.
4. Its user preferences are dynamic, accounting for changes in user interest over time.

**3.3 Method Adopted in the Proposed System**

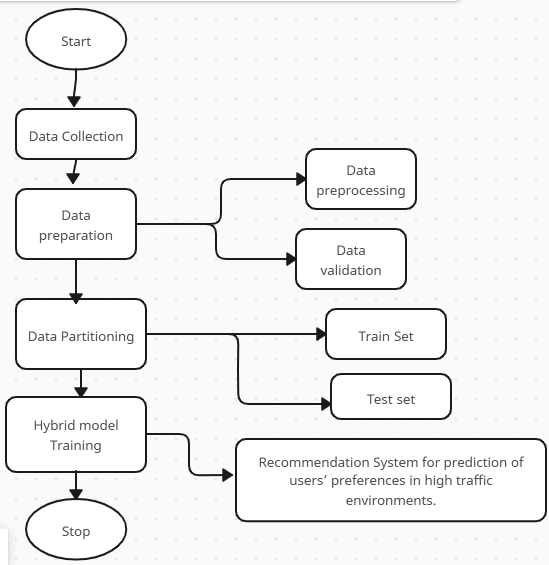
Prototype methodology was employed to develop the model. This approach facilitates the creation of a prototype that encompasses all necessary functionalities, allowing for further developments, relevant changes, and modifications.

**3.4 System Design**

Building an enhanced user-preference prediction system in high high-traffic environment involves building a recommendation system. Building a recommendation system involves a multi-step process that is both intricate and fascinating. Initially, one must understand the business objectives and define the goals of the recommendation system. Following this, data collection is paramount, where vast amounts of user interaction data, purchase histories, and item attributes are gathered. This data must then be meticulously cleaned and transformed, ensuring it is of high quality and in a format suitable for model training. The recommendation system will be built using a hybrid model, infusing both user-based and item-based filtering.

Model training and evaluation are the next steps, where machine learning models are trained using the prepared data, and their performance is evaluated using appropriate metrics. Once a model is chosen and trained, it must be fine-tuned to improve its accuracy and ensure it aligns with the business goals. The final step is deployment, where the model is integrated into the business ecosystem to start providing personalized recommendations to users.

**3.4.1 Process Design of the Proposed System**

.

**Fig 3.3 Process Design of the Proposed System (Drawn in Creatly)**

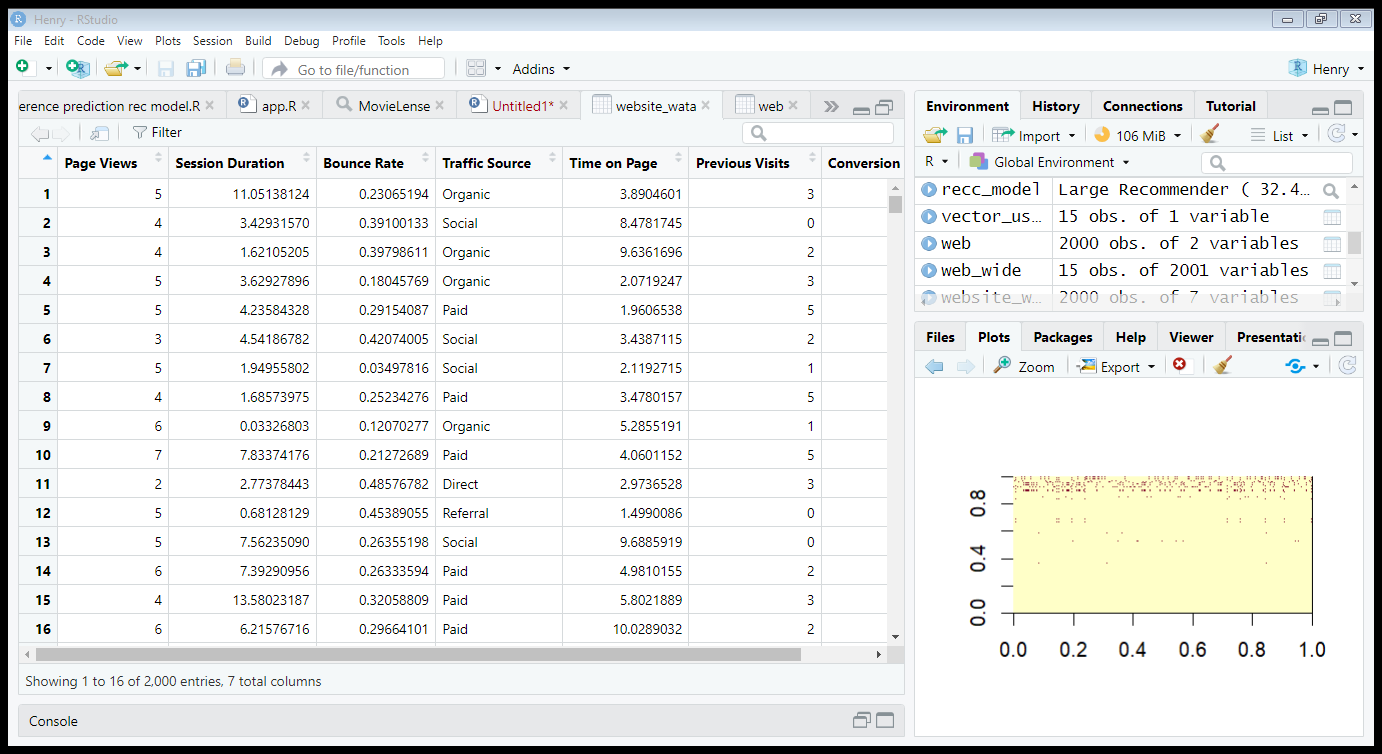
**3.4.2 Data Collection**

A website traffic dataset was obtained from Kaggle.com, an open-source data science website where datasets can be obtained without copyright issues. This dataset provides detailed information on website traffic, including page views, session duration, bounce rate, traffic source, time spent on the page, previous visits, and conversion rate. Variables like ‘spent on the page, page views, and traffic source’ are very important to our study as this will help us predict users’ preferences.

* + 1. **Dataset Description**

The dataset contains seven (7) variables from 2000 observations.

1. Page Views: The number of pages viewed during a session, feature indicates the engagement level of the visitors by showing how many pages they visit during their session.
2. Session Duration: The total duration of the session in minutes.
3. Bounce Rate: The percentage of visitors who navigate away from the site after viewing only one page. A critical metric for understanding user behavior. A high bounce rate may indicate that visitors are not finding what they are looking for.
4. Traffic Source: The origin of the traffic (e.g., Organic, Social, Paid).
5. Time on Page: The amount of time spent on the specific page.
6. Previous Visits: The number of previous visits by the same visitor.
7. Conversion Rate: The percentage of visitors who completed a desired action (e.g., making a purchase).



**Fig 3.4 Screenshot of the Dataset in R studio (First 16 observations).**

**3.4.4 Data Preparation**

Once collected, the data must be cleaned and validated to ensure consistency, accuracy, and relevance. Detect and remove noise, inconsistencies, and duplicates from raw data. For example, remove invalid entries, correct data formats, and handle missing values.

**3.4.5 Data Validation**

Check for format correctness (e.g., correct time format), value ranges (e.g., page views > 0), and completeness. Invalid data is either corrected or discarded. Considering data variables.

**3.4.6 Defining a Rating Matrix**

The target is to define a table having a row for each item and columns for each views.

You can use the following steps to define a rating matrix:

* 1. Label the cases.
  2. Define a table in the long format.
  3. Define a table in the wide format.
  4. Define the rating matrix.

**3.4.7 Computing Similarity Matrix**

Collaborative filtering algorithms are based on measuring user similarity or item similarity. For this purpose, recommenderlab contains the similarity function. The supported methods to compute similarities are Cosine, Pearson, and Jaccard. For instance, we might want to determine how similar the first five users are to each other.

**3.4.8 Data Partitioning**

In order to train our model, and get the best possible result, the data will be divided into two, a training and test set.

1. Training set: used to train the model and
2. Test set: used to test and evaluate the model.

This project uses a 70:30 approach in partitioning the dataset for training and testing the model. The larger portion of the data (70%) will be used to train the model while the smaller (30%) will be used to test the model.

**3.4.9 Hybrid Model Training and Prediction Process**

At the core of user preference prediction is the machine learning model, which uses the extracted features to make predictions about user preferences. The model is trained using unsupervised machine-learning techniques. Cross-validation is performed to ensure the model generalizes well on unseen data. In a real-time setting, the system uses the trained model to predict user preferences as the user interacts with the platform. Based on these predictions, the system generates personalized recommendations or content in real-time.

The recommendation system uses the machine learning model outputs to suggest relevant products or content to users. The Hybrid recommendation system combines both collaborative filtering and content filtering to create a more stable prediction model that ranks users’ preferences based on the prediction scores from the machine learning model. It then applies business rules (e.g., likely views, priority items) to finalize the recommendation list. This list is sent back to the front end, where users see personalized suggestions. Periodically, the system retrains the machine learning models using fresh data to improve accuracy and adapt to changing user behavior patterns.