TRAVELOGGERS:

Machine Learning based web application

**1.ABSTRACT**

Blogging is one of the most widely used ways to showcase your content. However, travel  bloggers do not have a collective platform where one can connect and engage with a larger audience.  With the purpose of connecting bloggers to a larger audience and overcoming drawbacks studied  across various social platforms Traveloggers was created. Traveloggers is a Web Application built  with Python Flask using Machine Learning features like recommendation system,  chatbot, image processing, etc. Our main  goal in this website is to provide reliable content to our readers and make sure that no copied content  or no foul content or images are uploaded on the website by verifying all sorts of content using  different algorithms for preventing any copyrights. Along with this, the users will get recommendations of different places they’ll like visiting on the basis of user’s location and previous searches.

**Keywords:** web application, machine learning, Python flask,collaborative filtering, recommendation  system, chatbot

Prof. Mrs.Poonam Rakibe Patil

Sapana Bhate, Hiral Gujrathi, Adesh Watade, Sejal Yeola

*Department of Information Technology, PES MCOE Pune*

**2.INTRODUCTION**

Interacting with a broad audience has become very possible in this new technological age. Why not  create a website specifically for travel bloggers and create a web application for them to display their  work and interact digitally using machine learning? It will support the expansion of the blogging  community. We saw the necessity for this website since it will just focus on blogging, a feature that  is not already offered anywhere, and because it will include several features that use ML algorithms  like Shingle Algorithm and Rapid Automatic Keyword Extraction for  intelligent and efficient execution of the features.

We will be using Python, Flask, MongoDB, React for developing the website.The features in the  website such as recommendation system, weekend planner using different ML algorithms.

The following are the system's development goals:

To provide a system to recommend blogs and weekend plans for users using machine learning. To deliver trustworthy information to readers using algorithms such as Reverse Image Search, Shingle to verify content originality. To provide blog analytics to the creators to evaluate their progress and reach. To Integrating GPS for location based services.

The domain of this project is Machine Learning and Web Development. With web development  frameworks like Python-Flask and Machine Learning Algorithms like Shingle Algorithm and Rapid Automatic Keyword Extraction.

**3.BACKGROUND AND LITRERATURE SURVEY**

With the constant increase of the bloggers in the community, there is a need for a centralized  platform for the community of frequent bloggers and readers. Currently, we have separate websites  for individual bloggers. However, there are a few drawbacks. For example, for uploading blogs one  has to create a personal website which is mostly paid and would have to use other social media  platforms for promotion. On the other hand, from a reader’s perspective, one has to go through  various individual websites in order to explore. Therefore, a centralized platform where one can post  and read for free is vital. Bloggers would require comparatively less efforts for content creation and  promotion with the help of algorithms like Collaborative Filtering and readers would get to explore  top and relevant blogs based on their search queries with the help of Rapid Automatic Keyword  Extraction algorithm. Their user experience on the application will be increased by shingle  algorithm and image authentication.

**4. SYSTEM DESIGN**

**4.1 System Architecture**

Fig 4.1.1 System architecture



 Fig.4.1.2 Data Flow Diagram

**4.2 Modules:**

**1) Login Options:**The user will be given two login choices in this module:

1. Login as a creator.

2. Sign in as a standard user.

You can switch to creator mode at any moment.

**2) Analytics Dashboard:**For measuring their views, likes, and understanding what their followers are favouring more,  creators will have a dedicated dashboard.

**3) Database:**Since MongoDB allows for quicker data retrieval, we will use it to store our data.In this situation, it is easier to find data by segregating information based on the user types  that we have provided.

**4) Chatbot:**The chatbot will help the users if they face any difficulties in handling the website. This  chatbot will help to solve the maximum number of problems faced by the user.

 **5) Google Maps Integration :**To make it simple for users to find the place listed in the blog description without having to  invest additional effort to search it on another app, we will integrate Maps.

**6) Nearby places recommendation :**Every weekend, we'll recommend places to go in a certain region depending on the user's  location and tastes to make our application more engaging. We won't be informing the users frequently because doing so can irritate certain users. To  construct this kind of function, we'll need the Google Maps API and the GooglePlaces Python  package.

**5.METHODOLOGY**

Following are the various algorithms we’ll be using for implementation :

**5.1.Shingle Algorithm:** The shingling algorithm is a technique commonly used in text analysis and similarity detection. It involves representing a document as a set of overlapping subsequences, or "shingles," of length k. We are using this algorithm for plagiarism checking of the blogs created by the users. This will result in maintaining the originality and genuineness of the content uploaded on our website. Here's a basic overview of how the shingling algorithm works:

1. Tokenization: The document is first tokenized into individual words or characters, depending on the level of granularity desired.

2. Shingle Generation: Sliding windows of length k are applied to the tokenized document, generating a set of overlapping shingles. For example, if k is 3 and the document is "Hello, how are you?", the shingles would be {"Hel", "ell", "llo", "lo,", "o, ", ", h", " ho", "how", "ow ", "w a", " ar", "are", "re ", "e y", " yo", "you", "ou?"}.

3. Shingle Representation: Each shingle is hashed or assigned a unique identifier to represent it.

4. Shingle Set: The shingles are collected into a set, removing duplicates. This set represents the document.

The resulting shingle set can be used for various purposes, such as detecting document similarity by comparing shingle sets of different documents. Similarity can be measured by calculating the Jaccard similarity coefficient, which is the size of the intersection of two shingle sets divided by the size of their union.The shingling algorithm is often used as a preliminary step in tasks like plagiarism detection, document clustering, and information retrieval. By representing documents as sets of shingles, it enables efficient comparison and analysis of large text datasets.



Fig 5.1 Plagiarism Detected

**5.2 Reverse Image Search:** A reverse image search is a technique that allows you to find similar or related images based on a given image. Instead of using keywords or text-based queries, you use an image as the input to retrieve relevant images from various sources on the internet. This can be helpful in identifying the source of an image, finding higher-resolution versions, or discovering visually similar images. In our website we will be using this technique to authenticate the images posted by the blogger. The image posted should be original and not copied from any other website.



Fig. 5.2 Copied Image found

**5.3 Rapid Automatic Keyword Extraction (RAKE):** This algorithm is a keyword extraction technique used in natural language processing (NLP). It is designed to identify important keywords or key phrases from a given text document.

Here's a high-level overview of how the RAKE algorithm works:

1. Preprocessing: The text document undergoes various preprocessing steps, including tokenization, removing stop words (common words like "and," "the," "is"), and punctuation.

2. Phrase Extraction: The text is then split into candidate phrases using phrase boundary detection. Phrases can be formed by identifying sequences of words that are connected by certain types of punctuation, such as hyphens or colons.

3. Keyword Scoring: Each candidate phrase is scored based on two criteria: frequency and degree of word co-occurrence. The frequency score measures how often the phrase appears in the document. Phrases with high scores are considered potential keywords.

4. Keyword Extraction: The candidate phrases are ranked based on their scores, and a set of keywords or key phrases is extracted from the top-scoring candidates. These keywords can be used to represent the main topics or themes present in the text.

The RAKE algorithm is designed to handle domain-specific terminology and extract meaningful keywords even in the absence of predefined dictionaries or training data. It is a simple and efficient method for keyword extraction, although it may not capture contextual or semantic relationships between words.



Fig. 5.3 Keyword extracted using RAKE

**5.4 Collaborative Filtering**: Collaborative filtering is a technique used in recommender systems to provide personalized recommendations to users based on their past behaviors or preferences, as well as the behaviors and preferences of similar users. It is a widely used approach for generating recommendations in various domains, such as e-commerce, movie streaming platforms, and music services.

We will be using this algorithm for recommending different places to users based on their previous searches and the user’s location. For example: If a user reads blogs related to trekking, hiking or mountains, he/she will be recommended places related to this category

Collaborative filtering typically involves two main types: user-based collaborative filtering and item-based collaborative filtering.

* User-Based Collaborative Filtering: This approach focuses on finding users who have similar preferences to the target user
* Item-Based Collaborative Filtering: In this approach, the focus is on finding similar items based on user ratings or interactions.

For this application we’ll be using Item-based collaborative filtering. The steps involved in item-based collaborative filtering are:

a. Item Similarity Calculation: Similarity measures, such as cosine similarity or Jaccard similarity, are used to calculate the similarity between items based on the ratings or interactions of users.

b. Neighbor Selection: The most similar items to the target item are selected as neighbors.

c. Recommendation Generation: Items that are similar to the ones the target user has already rated or interacted with are recommended to the target user.



Fig. 5.4 .Blog Recommendation

**6.SPECIFICATIONS**

**6.1 Project Scope:**Travel bloggers can share their traveling experience using this platform. Those who like writing can  also be benefited by this platform. Planning your weekends is a tiresome task for some, Traveloggers  makes this task much easier by recommending different weekend plans which are practically  possible based on the user’s interests and location.

**6.2 Assumptions And Dependencies:**

Assuming that our platform has been set up on Heroku and is operational:  Any browser can be used by users to access it, regardless of their location. Although they  cannot use mobile devices to browse this website.

**6.3 Functional Requirements:**

**User-Interface:** User will have a landing page as the first interface from there he/she can navigate to other pages such  as blog page, profile page, authors profile page.  Creators will be having some additional interfaces such as analytics page, create blog page,etc.

**External Interface Requirement :**

**User Interfaces :**

The system shall use standard user interface controls such as buttons, text boxes, radio  buttons, check boxes, labels, list boxes, spin boxes, combo boxes, sliders, scroll bars, tabs,  tool tips, progress bars, and file selection dialogues.

The user interfaces shall be presented as web pages and shall be displayed by web browsers.

**6.4 Non-functional Requirements**

**6.4.1 Performance Requirements**: RAM should be empty enough to run the program smoothly. Minimum 4GB RAM.

**6.4.2 Safety Requirements**

1.The data safety must be ensured by arranging for a secure and reliable transmission media.

2.The source and destination information must be entered correctly to avoid any misuse or  malfunctioning.

3.Safety requirements against natural disasters and accidents.

Failures due to technical issues.

**6.4.3 Security:**The access to the system is given only to valid users. We need a specific ID and password to  get access to the system.

1. Communication needs to be restricted when the application is validating the user.

2. By incorporating a robust and proven MongoDB into the system, reliable performance and  integrity of data is ensured. There must be a power backup for the server system.

**7. CONCLUSION & FUTURE WORK**

In this project, we created a social media web application using Python Flask with additional features of machine learning like recommendation system using Collaborative Filtering, Plagiarism checker using Python Libraries, Weekend Plan Recommendation using the outputs of recommended blogs to the user, RAKE algorithms will be used for searching and Google Maps Integration. Currently, the project will be implemented as a Web Application but will further be  deployed as a Mobile Application using Flutter.We use the Reverse Image Search method to assess the authenticity of images, which  is 70-80% accurate. In the future, we intend to improve the algorithm's accuracy. Establishing a rewards-based loyalty programme for Users for their loyalty towards  the website.

**8.REFERENCES**

**[1]**Schafer, Ben & Frankowski, Dan & Dan, & Herlocker, & Jon, & Shilad, & Sen, Shilad. (2007). Collaborative Filtering Recommender Systems.

**[2]**Rose, Stuart & Engel, Dave & Cramer, Nick & Cowley, Wendy. (2010). Automatic

Keyword Extraction from Individual Documents. 10.1002/9780470689646.ch1.

**[3]**An intelligent testing system development based on the shingle algorithm for assessing humanities students academic achievements © The Author(s), under exclusive licence  to Springer Science+Business Media, LLC, part of Springer Nature 2022

**[4]**Baker JR, Moore SM. Blogging as a social tool: a psychosocial examination of the  effects of blogging. Cyberpsychol Behav.2008Dec;11(6):747-9. doi: 10.1089/cpb.2008.0053.PMID: 19072151.

**[5]**Research on Collaborative Filtering Recommendation Algorithm Based on Mahout and  User Model © The Author(s),Bo Song and Yue Gao and Xiao-Mei Li,  To cite this article: Bo Song et al 2020 J. Phys.: Conf. Ser. 1437 012095