Abstract

This project presents Safespace, a web-based healthcare application that provides personalized drug recommendations and enhances user safety. The system's core functionality is built around a drug recommendation engine that uses a Random Forest Classifier trained on a comprehensive drug dataset. This engine processes user-input descriptions, symptoms or conditions to suggest appropriate medications. The application integrates this machine learning model with a secure user authentication system, a Streamlit-based user interface, and an emergency alert feature. Safespace demonstrates the effective use of machine learning techniques in healthcare, combining drug recommendation capabilities with safety features to create a multifaceted tool for personal health management.

Project Documentation

Medical Drug Recommender

Overview

This project is a web application built using Streamlit, designed to provide personalized drug recommendations based on user input. The application uses machine learning techniques for drug recommendations and ensuring complete safety through communication to the user.

Introduction

Introduction: In today's fast-paced world, access to quick and reliable healthcare information is crucial. Many individuals face challenges in finding appropriate medications for their symptoms or conditions, often relying on potentially unreliable internet searches or waiting for doctor appointments. Additionally, in emergency situations, the ability to quickly alert loved ones and provide location information can be life-saving. There is a growing need for a comprehensive solution that addresses both everyday healthcare needs and emergency scenarios in a user-friendly, accessible manner.

Problem Statement:

The current healthcare landscape lacks an integrated platform that combines personalized drug recommendations with emergency alert capabilities. Many individuals struggle to find accurate information about medications suitable for their specific symptoms, leading to potential misinformation or delayed treatment. Furthermore, existing emergency alert systems often fail to incorporate real-time location data, crucial for swift assistance in critical situations. There is a clear need for a solution that leverages machine learning technology to provide accurate drug recommendations based on user-described symptoms, while also offering a reliable emergency alert system. This solution should be easily accessible, user-friendly, and capable of bridging the gap between everyday health management and emergency response. The challenge lies in creating a secure, efficient system that can process user health data, generate reliable drug recommendations, and seamlessly integrate with emergency contact protocols, all while maintaining user privacy and data security.

Table of Contents

User_page.py

- Purpose: Creates a user interface for displaying user details, fetching location, sending emergency contact messages, and recommending drugs.
- Key Functions:
  - create_connection(): Establishes a connection to the MySQL database.
  - get_user_details(user_id): Retrieves user details from the database.
  - fetch_location(user_id): Fetches the user's location.
  - send_emergency_message(contact, message): Sends an emergency message via WhatsApp using Twilio.
  - recommend_drugs(description): Recommends drugs based on the user's input description.

Register_page.py

- Purpose: Creates a user registration page and stores user information in the MySQL database.
- Key Functions:
  - create_connection(): Establishes a connection to the MySQL database.
  - register_user(user_details): Registers a new user by inserting their details into the database.

Login_page.py

- Purpose: Creates a login page and validates user credentials against the MySQL database.
- Key Functions:
  - create_connection(): Establishes a connection to the MySQL database.
  - validate_user(username, password): Validates the user's credentials.

app.py

- Purpose: Integrates the login, registration, and user functionalities into a multi-page web application.
- Key Functions:
  - main(): Main function to run the Streamlit application, handling navigation between login, registration, and user pages.

drug_recommender.py

- Purpose: Recommends drugs based on user input descriptions using machine learning techniques.

- Key Functions:
    - recommend_drugs(description, top_n=3): Recommends top n drugs based on the input description.
- Key Components:
    - **Data Preprocessing**: Reads and preprocesses data from output1.csv.
    - **Model Training**: Trains a RandomForestClassifier on the preprocessed data.
        - **Model Saving**: Saves the trained model, TF-IDF vectorizer, and label encoder using pickle.

Methodology and Processes:

Development approach used was Agile Development. The project followed an iterative development approach, focusing on one page at a time.

Project Development.

Planning and Development

The product was started off by creating the user page. The approach concentrated on the User Interface to ensure it is simple, easy and uncluttered.

The product login page interface came to focus and for simplicity, the email address and password were the best data to be captured and used for safe login.

The Register page interface needed new data to be used as the response scheme that was to be implemented using the picked API; “Twilio API”.

Adding the Geolocation Api and coding the ensemble to run the Random Forest Tree classifier algorithm. The data collected was given into the model and the process took two days on the Streamlit Server,

User Authentication System:

- Implemented a login and registration system using **Streamlit** for the frontend and **MySQL** for the backend.
- Created separate pages for login (Login_page.py) and registration (Register_page.py).
- Used MySQL Connector to establish database connections and perform user authentication.

Drug Recommendation System:

- Developed a machine learning model using **RandomForestClassifier** for drug recommendations (drug_recommender.py).
- Used **TF-IDF vectorization** for text feature extraction.
- Trained the model on a dataset containing **drug names, reasons, and descriptions.**

- Implemented a function to recommend drugs based on user input descriptions.

User Interface:

- Created a main application (app.py) that manages the overall flow and navigation between pages.
- Implemented a user dashboard (User_page.py) with features; **drug recommendations and emergency alerts.**

Emergency Alert System:

- Integrated Twilio API to send emergency WhatsApp messages.
- Implemented **geolocation** functionality to include user's location in emergency messages.

Testing

Unit Testing: I tested the individual components for functionality

Integration Testing: I tested all the component using (**app**.**py**), the control flow. Worked smoothly.

User Testing: I gave a few colleagues the link to use the system and they reported the model gave accurate response and a few errors because of general use drugs.

How to Run the Project

Setup:

- Ensure you have Python installed.
- Install the required libraries using: *pip*
- pip install streamlit, mysql-connector-python, requests, twilio, scikit-learn, pandas

Database Setup:

- Set up a MySQL database and create the necessary tables for storing user information.

Running the Application:

- Run the Streamlit application using: *streamlit run (app_name).py*
- streamlit run app.py

Additional Notes

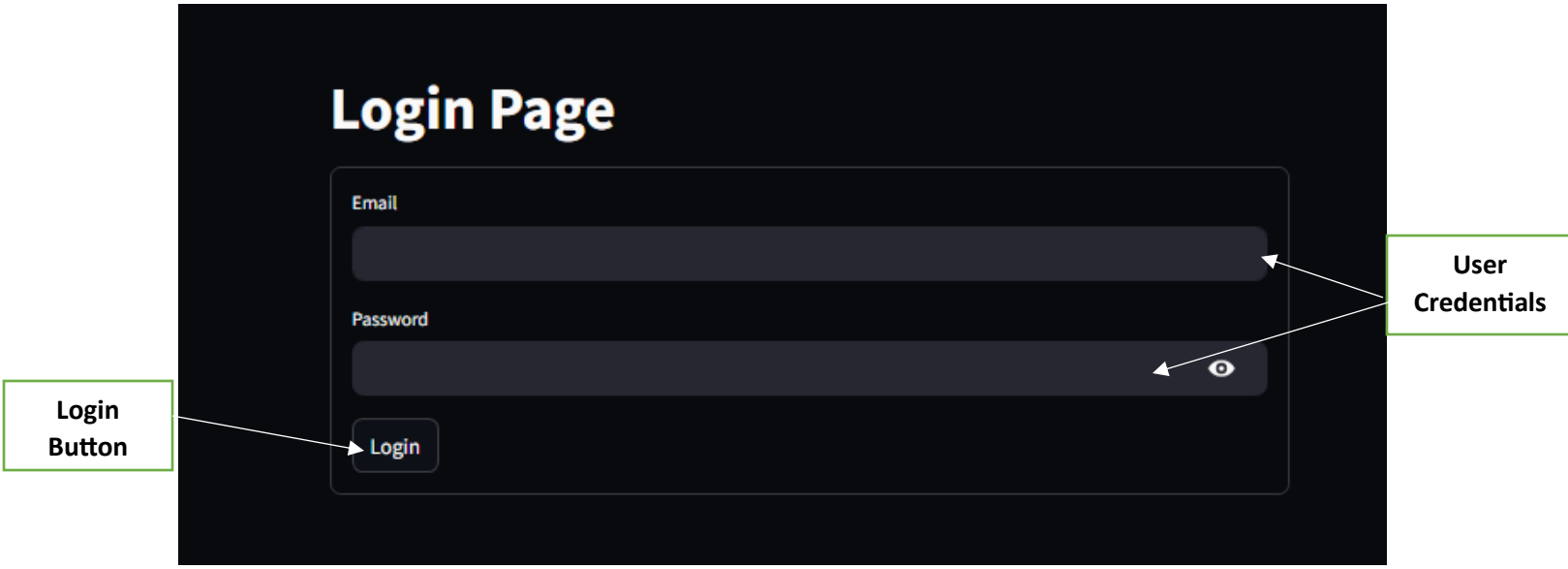
- **Dependencies**: Ensure all required libraries are installed.
- **Configuration**: Update the database connection details in each script as needed.
- **Security**: Ensure sensitive information like database credentials and Twilio API keys are securely managed. Streamlit hosting uses secrets to enforce encrypted and secure credentials of the developer.

User Manual.

Tailored experience

Login.

Welcome to the register page. First Name, Last Name, Phone Number, Email Address, Password, Emergency Contact Information. A user types their credentials into the respective fields and types the emergency contact information which will be logged into the user database. When all required spaces are filled, you will be able to register and be directed to the login page.



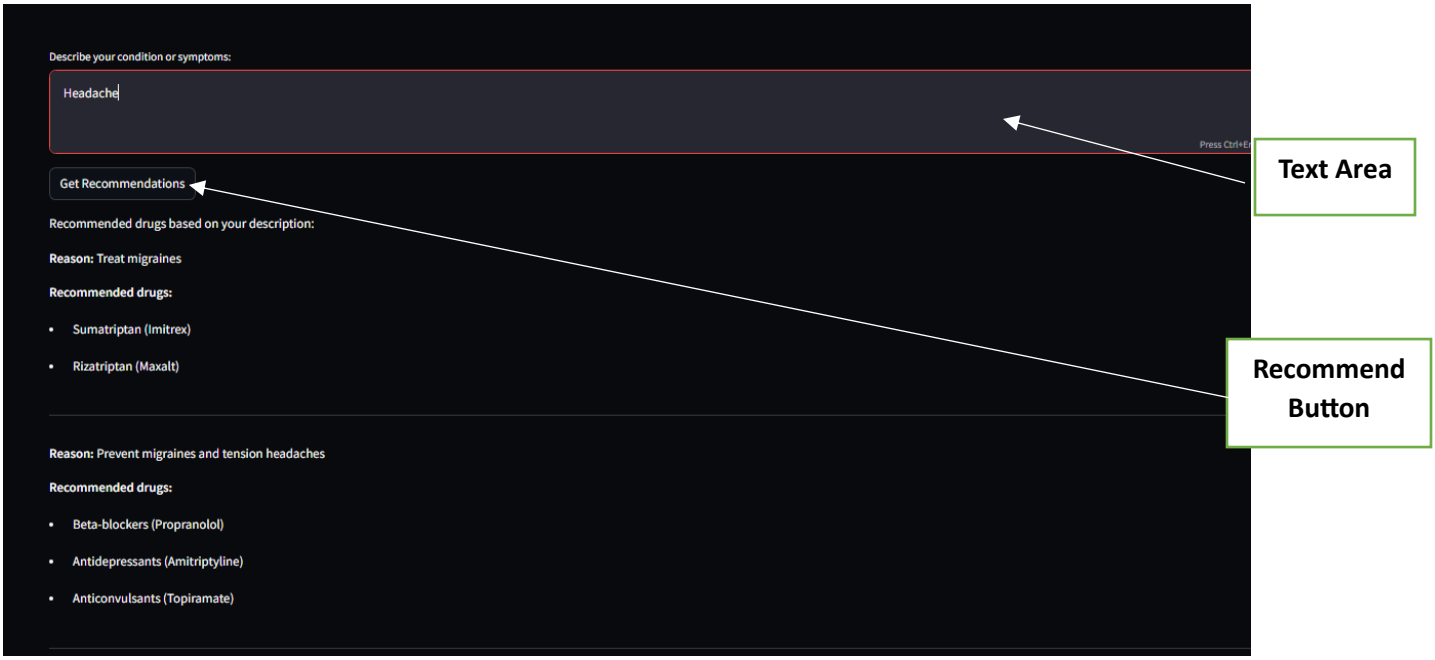
Welcome to the Login page. Email, Password, Login Button. With the email provided at the time of registration, fill the cards your email address and your password and click the login button to be directed to the user page.

User page.

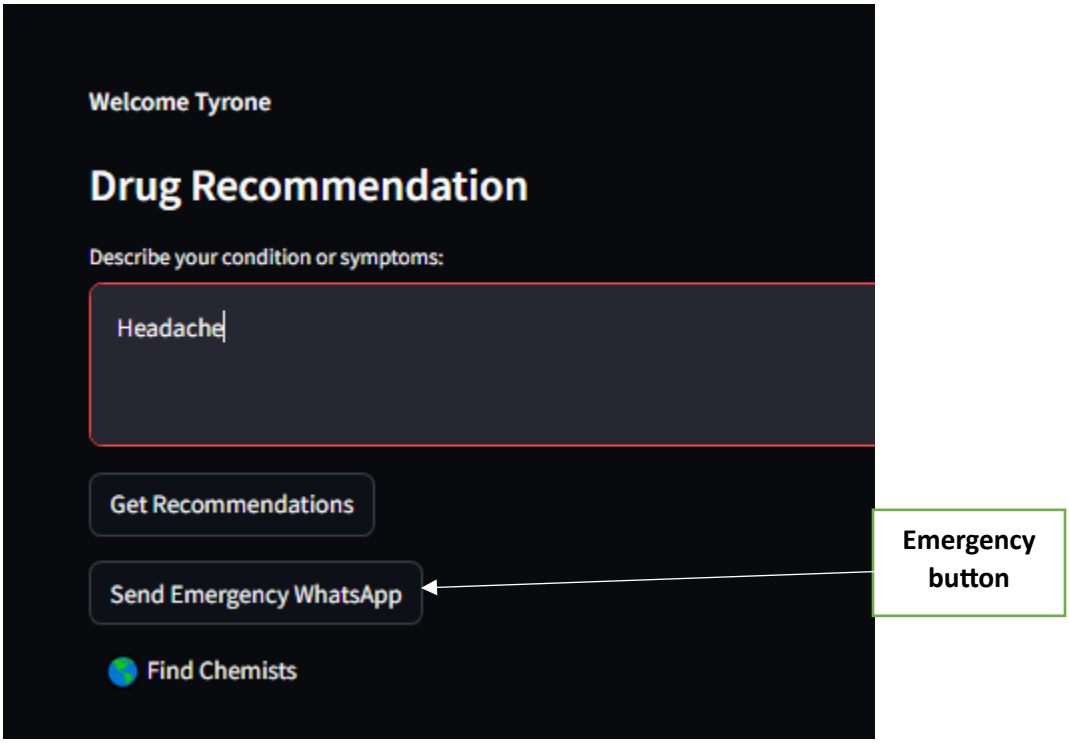
Successful entry directs the user to a page capturing their mail and user name registered.

The page is simple with clear interactive cards for use. Type the symptom(s) you are certain

of and press the *"Get Recommendation"* button. The application will run your input and give the best possible medicine to use.



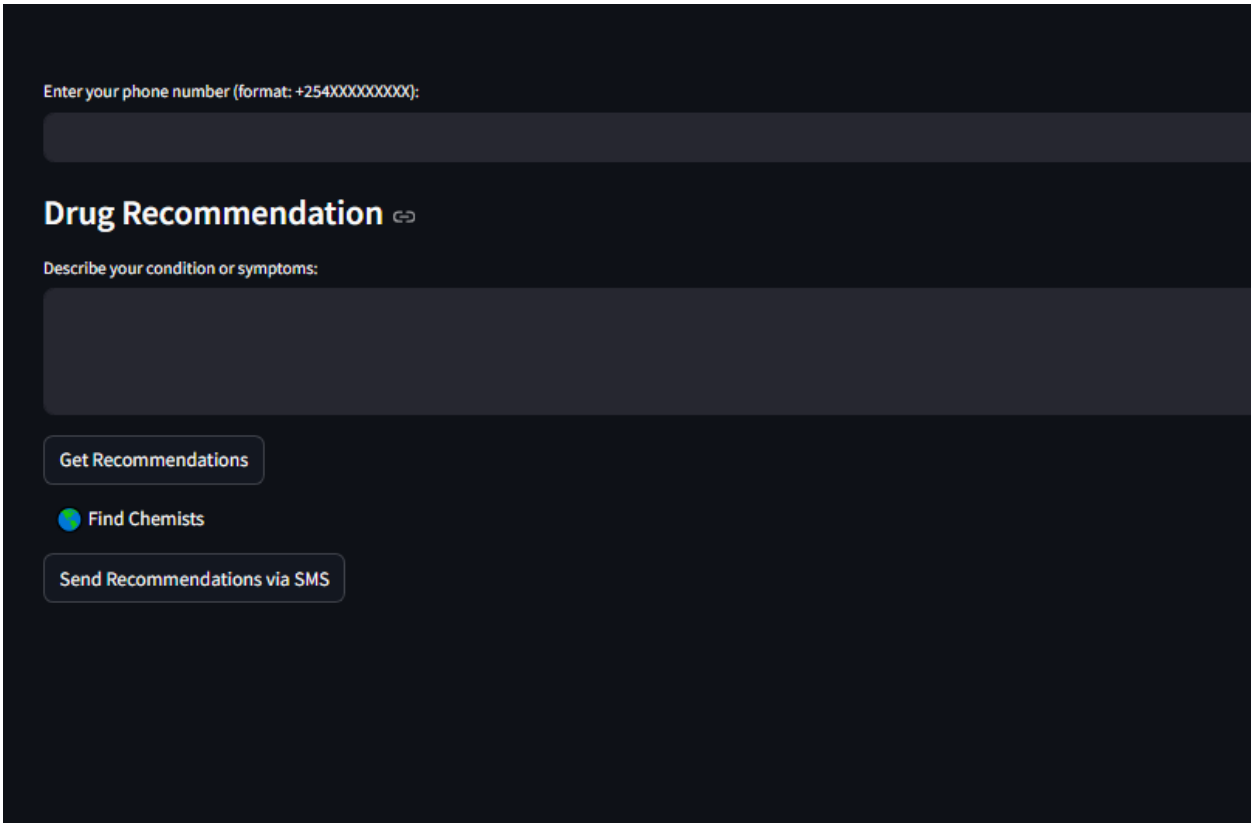
The application will load the medical recommendation and return a link to **Google Maps** which automatically queries the nearest chemist to your location.



Beta experience

User Page

User interacts with the application by typing their symptoms and having feed their approved phone number by twilio. Follow the link and register your phone number: twilio.com/en-us. The recommendations will then be sent to your phone number.



Findings and Results

Authentication System:

- Successfully implemented user registration and login functionality.
- Stored user information securely in a MySQL database.

Drug Recommendation System:

- The system can process user-input descriptions of conditions or symptoms.
- It provides recommendations for drugs based on the input, including the reason and top 5 recommended drugs for each category.

User Dashboard:

- Created a personalized dashboard displaying user information.
- Integrated the drug recommendation feature into the user interface.
- Implemented an emergency alert button that sends WhatsApp messages with user location.

Geolocation:

- Successfully integrated IP-based geolocation to determine user's approximate location for emergency alerts.

Integration:

- Successfully combined various components (authentication, recommendation system, emergency alerts) into a cohesive application.

Machine Learning Model for Drug Recommendations

Overview

The drug recommendation system utilizes a machine learning model to suggest appropriate medications based on user-described symptoms/conditions. The model is implemented in the `drug_recommender.py` file and uses a combination of natural language processing (NLP) and classification techniques.

Model Architecture

The recommendation system employs a Random Forest Classifier coupled with TF-IDF (Term Frequency-Inverse Document Frequency) vectorization for text processing.

TF-IDF Vectorization

- Used to convert text descriptions into numerical features
- Implemented using `TfidfVectorizer` from scikit-learn
- Parameters:
- `max_features=1000`: Limits the vocabulary to the top 1000 terms
- `stop_words='english'`: Removes common English stop words

Random Forest Classifier

- Ensemble learning method for classification
- Implemented using `RandomForestClassifier` from scikit-learn
- Parameters:
- `n_estimators=100`: Uses 100 trees in the forest
- `random_state=42`: Ensures reproducibility of results

Data Preprocessing

- The model uses a CSV file (`output1.csv`) containing drug information
- Text data is combined from 'Drug_Name', 'Reason', and 'Description' columns
- The 'Reason' column is encoded using `LabelEncoder` for classification

Training Process

- The combined text data is vectorized using TF-IDF
- The encoded 'Reason' serves as the target variable
- The Random Forest model is trained on the TF-IDF vectors and encoded reasons

Recommendation Process

- 1. User input is transformed using the same TF-IDF vectorizer
- 2. The model predicts probabilities for each possible reason
- 3. Top N reasons are selected based on these probabilities
- 4. Drugs associated with these reasons are retrieved from the dataset

Model Persistence

The trained model and associated transformers are saved to disk using pickle:

- - `rf_classifier.pkl`: Trained Random Forest model

- - `tfidf_vectorizer.pkl`: Fitted TF-IDF vectorizer
- - `label_encoder.pkl`: Fitted Label Encoder

Usage in Application

The `recommend_drugs` function in `drug_recommender.py`:

- Loads the saved model and transformers
- Processes user input
- Makes predictions
- Returns top N recommendations with associated drugs

Performance and Limitations

- The model's performance depends on the quality and quantity of the training data
- It may not capture complex medical relationships or contraindications
- Recommendations should be treated as suggestions and not as professional medical advice

Challenges

- Litle data to train the model. The anticipated method of data collection through APIs was impossible following much of the data management organisations limiting the use to Enterprises.
- Microsoft Sentinel Technology subscription was accepted and unfortunately rejected following unregistered data source.
- The Machine Learning Model was too big to increase the decision trees for the online servers and likewise the local machine. The project meets all objectives and recommends drugs to its best level.
- Government and regional Data Organisations are treating the requests as malicious intent, even though submitting proof of project.
- Twilio does not support sending information of any sort to unregistered users. It is advised to register your phone number on Twilio website for verification. Upon verification the alerts will be sent to you number.

Conclusion

The application demonstrates the effective use of machine learning for personalized healthcare recommendations and integrates modern web technologies for a user-friendly experience. The emergency alert feature adds a crucial safety element, potentially aiding users in critical situations.

Future Improvements

- Incorporate more advanced NLP techniques like word embeddings or BERT
- Implement a more sophisticated ranking system for drug recommendations
- Include a feedback loop to continuously improve recommendations based on user feedback
- Integrate with a regularly updated medical database for more accurate and current recommendations
- Include Government and Health organization APIs that will give a massive amount of data for training.
- Include Geocoding for location information.

https://safespace.streamlit.app/

References

- Miller, C. (2023, October 12). Flesh-eating fungus on the rise in American West, linked to climate change. *Los Angeles Times*. Retrieved from https://news.google.com/
- American Academy of Allergy, Asthma & Immunology. (n.d.). Drug allergies. https://www.aaaai.org/
- Centers for Disease Control and Prevention. (2023, March 21). Fungal diseases. https://www.cdc.gov/fungal/index.html