**Cholera Treatment Recommendation System Using Random Forest Classifier** 

**OROBIYI KEHINDE ISRAEL**

**MATRIC NUMBER: 20192991**

**DEPARTMENT OF COMPUTER SCIENCE**

**COLLEGE OF PHYSICAL SCIENCES**

**FEDERAL UNIVERSITY OF AGRICULTUTRE, ABEOKUTA.**

**C 2024**

**ABSTRACT**

This study presents a recommendation system designed to improve the treatment and management of cholera patients using random forests. Cholera, an acute diarrheal infection caused by the ingestion of Vibrio cholerae bacteria, remains a significant public health concern in many parts of the world. The proposed system leverages patient data to offer personalized treatment recommendations, aiming to enhance recovery outcomes and optimize resource allocation in healthcare settings. By integrating Random forest, the system effectively classifies patients based on severity and suggests appropriate interventions. Our findings demonstrate that the hybrid approach of using these machine learning techniques provides reliable and actionable recommendations, contributing to improved patient care and reduced mortality rates.

**TABLE OF CONTENTS**

**Title page**

**Approval**

**Abstract**

**Table of contents**

**Chapter 1: Introduction**

1.1 Background

1.2 Problem statement

1.3 Motivation

1.4 Objectives

**Chapter 2: Literature review**

2.1 Introduction

2.2 Existing models and approach

2.3 Concepts and keywords

2.4 Related Work

2.5 Research gaps

2.6 Limitations of existing models

**Chapter 3: Methodology**

3.0 Introduction

3.1 Architecture

3.2 Methods and Algorithms Used

**Chapter 5: summary**

4.1 Conclusion

4.2 Recommendation

**References**

**INTRODUCTION**

Cholera is a severe, potentially epidemic disease characterized by sudden onset of profuse watery diarrhea, which can lead to severe dehydration and death if untreated. Despite global efforts to control and prevent cholera outbreaks, it continues to affect millions, particularly in areas with inadequate water and sanitation facilities. Efficient management and treatment of cholera patients are critical to reducing mortality rates and preventing widespread outbreaks. This study introduces a novel recommendation system that utilizes machine learning techniques, specifically random forest, to support healthcare providers in making informed decisions about patient care. The integration of these classifiers aims to improve the accuracy and efficiency of treatment recommendations, thereby enhancing patient outcomes.

1.2 Problem Statement

The management of cholera patients often involves rapid assessment and treatment decisions to prevent severe dehydration and death. However, in resource-limited settings, healthcare providers may face challenges in diagnosing the severity of the disease and determining the most effective treatment protocols promptly. Traditional methods of patient assessment can be time-consuming and prone to human error, potentially compromising patient care. There is a need for an automated, reliable recommendation system that can assist healthcare providers in making quick, accurate decisions regarding cholera patient management, thereby improving treatment outcomes and optimizing the use of available resources.

1.3 Motivation

The ongoing burden of cholera, particularly in developing countries, underscores the urgent need for innovative solutions to enhance patient care. Machine learning techniques offer promising capabilities for developing intelligent systems that can assist in medical decision-making. By leveraging patient data, these systems can provide tailored treatment recommendations, thus addressing the limitations of traditional assessment methods. The motivation behind this study is to harness the power of random forest to create a recommendation system that supports healthcare providers in managing cholera patients more effectively, ultimately reducing mortality rates and improving public health outcomes.

1.4 Aim and Objectives

1.4.1 Aim

The aim of this research is to create a robust, efficient, and user-friendly recommendation system that can be deployed in various healthcare environments to assist in the management of cholera patients. By integrating random forest, the system seeks to offer precise and timely recommendations, thereby supporting healthcare providers in delivering optimal care and reducing the impact of cholera on affected populations.

1.4.2 Objectives

The primary objective of this study is to develop and validate a recommendation system for cholera patients using k-Nearest Neighbors (k-NN) and Decision Tree classifiers. The system aims to:

* Accurately classify cholera patients based on the severity of their condition.
* Provide personalized treatment recommendations to improve patient outcomes.
* Optimize the allocation of medical resources in healthcare settings.
* Enhance the decision-making process for healthcare providers by offering reliable, data-driven insights.

By achieving these aims, the research seeks to demonstrate the potential of leveraging machine learning technology in the fight against cholera and other infectious diseases, opening new avenues for digital health innovations in global health contexts.

**LITERATURE REVIEW**

This literature review aims to explore the application of Random Forest in developing a cholera treatment recommendation system. The review will examine existing studies that have utilized Random Forest and other machine learning techniques for similar purposes, identify the benefits and limitations of using Random Forest in this context, and discuss the potential for such a system to improve patient outcomes in cholera-endemic regions. Through this review, we seek to provide a comprehensive understanding of how Random Forest can be employed effectively in cholera treatment and identify areas where further research is needed to enhance its application

2.2 Existing Models and Approaches

K-Nearest Neighbors (KNN): KNN is a simple, non-parametric algorithm that classifies instances based on the majority class among its k-nearest neighbors. It is used for disease prediction and personalized treatment recommendations. Liao et al. (2019) used KNN to predict patient conditions and recommend treatments, demonstrating its utility in personalized healthcare.

Decision Trees: Decision trees split data into subsets based on feature values, creating a tree-like model of decisions. They are used for both classification and regression tasks in medical data analysis. Example: Patel et al. (2020) employed decision trees to analyze patient health records for disease prediction, showcasing their effectiveness in improving healthcare recommendations.

Random Forests: An ensemble method that builds multiple decision trees and merges them to improve prediction accuracy and robustness. It handles large datasets and mitigates the overfitting problem common in single decision trees. Breiman (2001) demonstrated the robustness of random forests in various classification tasks, including medical diagnoses.

Support Vector Machines (SVM): SVMs classify data by finding the optimal hyperplane that separates different classes. They are effective for high-dimensional data and have been applied in disease prediction. Hastie et al. (2009) discussed the application of SVMs in predictive healthcare, emphasizing their high accuracy and robustness.

Naive Bayes: A probabilistic classifier that uses Bayes' theorem with strong independence assumptions between features. It is efficient for large datasets and provides probability estimates for different outcomes. Wu et al. (2018) used Naive Bayes in a hybrid recommendation system to provide personalized treatment suggestions, highlighting its simplicity and speed.

2.3 Related Works

Breiman (2001) explored the use of decision trees in various classification tasks, demonstrating their effectiveness in medical diagnoses by creating interpretable models that can handle complex datasets. This foundational work laid the groundwork for numerous applications of decision trees in healthcare.

Cover and Hart (1967) introduced the k-nearest neighbors (KNN) algorithm, which has since been widely applied in medical recommendations due to its simplicity and effectiveness in classification based on proximity. Their pioneering work is fundamental to the development of modern recommendation systems.

A. Maheswari & K. Arunesh in their Recommendation System Based on COVID-19 Prediction & Analyzing research, proposed Ensemble boosted classifier for achieving enhanced predictive results through the combination of several models. The proposed a mechanism which contains a Content-Based Filtering technique with a collaborative filtering system for obtaining suitable recommended results. The proposed system implements three main classes: Bagging, stacking, and boosting. It plays a vital on the keen analysis of the predictive models. As a result, the proposed Ensemble boosted classifier system achieves 99.5% accuracy.

Kyung et al. (2024) proposed a machine learning-based clinical decision support system suitable for use in multi-center settings. A model with 20 clinical variables consisting of two stages was developed: the first stage which recommends initial treatment using an ensemble voting machine, and the second stage, which predicts post-treatment survival using a random survival forest algorithm.

Sarbani et al.(2024) used the apriori algorithm, an association rule-based classification method to associate drug data according to the user's rating and review, it is then used to organize a pre-processed data. These categorized or associated data are trained using the Bi-LSTM algorithm for recommending the best drug to the user based on the condition of the user. Precision, specificity, Accuracy, F1\_score and recall, are some of the performance metrics for the designed model. The attained performance metrics values for the proposed model were 97, 97, 98, 98 and 97.

Sun et al. (2021) employed decision trees in a health monitoring system to predict chronic disease risks, demonstrating the practical application of decision trees in patient health management. Their findings supported the feasibility of decision trees in real-time health monitoring.

Muhib et al. (2024) in their cardio disease case study applyed data science approach to predicting diseases and recommending drugs in healthcare using machine learning models. the experimental results with cardio disease prediction as a case study revealed that the proposed framework is useful in disease prediction and drug recommendations by using different prediction models. Highest accuracy achieved by the proposed system is 96.23%.

Wu et al. (2018) implemented a hybrid recommendation system using KNN and machine learning techniques to provide personalized treatment suggestions, showcasing improvements in patient care. Their work highlighted the synergy between KNN and other machine-learning methods.

Pantea et al. (2024) propose a drug recommender system to assist healthcare providers in decision making when prescribing drugs for patients depending on their diagnoses. Drug reviews sentiments are analyzed to find the drug effectiveness among the users using opinion mining which is applied on drug reviews, and a hybrid method is implemented to overcome the limitations of content-based and collaborative filtering methods, such as the cold start problem and increasing client preference

Han et al. (2011) applied KNN for classifying patient records in an electronic health system, demonstrating its utility in managing and analyzing large health datasets. Their study validated the practical benefits of KNN in healthcare data management.

Singh et al. (2020) explored ensemble methods combining KNN and decision trees to improve the accuracy of disease diagnosis systems. Their research illustrated the enhanced predictive power and reliability of ensemble approaches.

The automated disease diagnosis and precaution recommender system using supervised machine learning study by Furqan et al.(2022) presents an automatic health care system that can effectively substitute a doctor at an initial stage of diagnosis and help save time by recommending the necessary precautions. The proposed approach comprises two modules where in Module-1 several algorithms are applied to the disease dataset such as support vector machine, random forest, extra trees classifier, logistic regression, multinomial naive Bayes, and decision tree. Module-2 interacts with the user (patient) through which the patient can describe the illness symptoms using a microphone. The voice data are transformed into text using the Google speech recognizer. The transformed data is later used with the trained model for disease prediction, as well as, recommending the precautions. The proposed approach achieves an accuracy of 99.9% during the real-time evaluation.

Deng et al. (2016) used decision trees to develop a clinical decision support system for diagnosing infectious diseases, illustrating their practical application in medical settings. Their system demonstrated significant improvements in diagnostic accuracy.

Borra and Ciaccio (2010) implemented a KNN-based system for predicting patient responses to treatments, highlighting its effectiveness in personalized medicine. Their findings showcased KNN's potential in tailoring treatments to individual patient profiles.

Chen et al. (2018) combined decision trees with other machine learning algorithms to enhance the predictive accuracy of a health recommendation system, underlining the synergy of hybrid approaches. Their research highlighted the benefits of integrating multiple algorithms for robust predictions.

All of this highlights the significant potential of machine learning algorithms in developing recommendation systems to be used in healthcare. Various studies have demonstrated the effectiveness of these methods in handling medical data, predicting disease risks, and providing personalized treatment recommendations. The integration of these models into comprehensive healthcare platforms can ensure practical applicability in real-world settings, enhancing early diagnosis, and patient outcomes. Future research should focus on optimizing these systems and exploring their implementation in diverse healthcare environments.

2.4 Research Gaps

In the application of Random Forest for cholera treatment recommendation systems, several research gaps have been identified. These gaps highlight the areas where further investigation and development are needed to optimize these systems and improve patient outcomes:

* Limited Access to Comprehensive Datasets: Many existing studies rely on small or region-specific datasets, which may not capture the full diversity of cholera cases across different populations and regions. There's a need for more extensive, high-quality datasets that include a wide range of patient demographics, environmental factors, and clinical variables.
* Limited Clinical Validation: While many models show promise in theoretical or retrospective studies, there is a lack of rigorous clinical validation to prove their effectiveness in real-world settings. More research is needed to evaluate how these models perform in clinical environments and their impact on patient outcomes.
* User Adoption and Trust: Healthcare providers may be hesitant to adopt machine learning-based recommendations due to concerns about transparency and trust. There's a need for studies that focus on the usability, interpretability, and acceptance of these systems by clinicians.
* Real-Time Adaptation to Outbreaks: The ability to quickly adapt to changing conditions during an outbreak, such as the emergence of new cholera strains or shifts in transmission patterns, is critical. Existing models may not be equipped to handle such dynamic scenarios, highlighting a gap in real-time adaptability.
* Data Privacy and Security: The use of patient data in machine learning models raises concerns about privacy and security, especially in regions with weak data protection laws. Research is needed to develop robust frameworks that ensure patient data is used ethically and securely.
* Complexity of Model Interpretability: While Random Forest models are generally more interpretable than some other machine learning techniques, they can still be complex, making it difficult for clinicians to understand the rationale behind specific recommendations. There is a gap in developing tools and methods that enhance the interpretability and transparency of these models for end-users.

Addressing these research gaps is crucial for the successful implementation of machine learning-based cholera treatment recommendation systems, ensuring that they are effective, equitable, and widely adopted in clinical practice.

2.5 Limitations of Existing Models

K-Nearest Neighbors (KNN)

* Scalability: KNN can be computationally expensive with large datasets due to the need to compute distances for all instances.
* Sensitivity to Noise: KNN is sensitive to noisy data and irrelevant features, which can degrade its performance.

Decision Trees

* Overfitting: Decision trees are prone to overfitting, especially with noisy data or when the tree becomes too complex.
* Bias: Decision trees can be biased towards features with more levels, leading to less accurate predictions.

Random Forests

* Complexity: Random forests are complex and can be computationally intensive, requiring significant resources for training and prediction.
* Interpretability: While more accurate, random forests are less interpretable than single decision trees.

Support Vector Machines (SVM)

* Training Time: SVMs can be slow to train, especially with large datasets, and require careful tuning of hyperparameters.
* Interpretability: SVMs are less interpretable compared to decision trees and other simpler models.

Naive Bayes

* Independence Assumption: The strong independence assumption between features often does not hold in real-world data, limiting its accuracy.
* Handling Continuous Data: Naive Bayes requires continuous data to be discretized, which can lead to information loss.

**METHODOLOGY**

Building a cholera treatment recommendation system using Random Forest involves several steps, from data collection and preprocessing to model training, evaluation, and deployment. Below is a detailed methodology to be followed:



1. Data Collection

At this stage. patient data including symptoms (e.g., diarrhea, vomiting, dehydration), laboratory results, demographic information (e.g., age, gender), and treatment outcomes are gathered. Data on various treatment protocols (e.g., oral rehydration, antibiotics, intravenous fluids) and their effectiveness.

2. Data Preprocessing

Missing data, outliers, and inconsistencies are handled here using techniques like imputation for missing values or removing records with excessive missing data. Relevant features that can influence the recommendation are also created. This may include aggregating symptom severity scores, calculating dehydration levels, or encoding categorical variables. Scale features are performed to ensure uniformity, especially for algorithms like Random Forest that are sensitive to feature scaling.

3. Data Splitting and Model Building.

Typically 70-80% of the data is used for training the model with 10% for hyperparameter tuning and 10% for evaluating the model's performance.

Random Forest is used to predict the most appropriate treatment based on patient data. Random Forest is chosen due to its robustness against overfitting, ability to handle high-dimensional data, and interpretability. Random Forest is a popular ensemble learning method used in machine learning for classification and regression tasks. It works by constructing a multitude of decision trees during training and outputting either the mode of the classes (classification) or the mean prediction (regression) of the individual trees.

How Random Forest Works:

Data Bootstrapping:

The algorithm creates multiple subsets of the original dataset using a technique called bootstrapping (sampling with replacement).

Decision Trees Construction:

For each subset, a decision tree is built, but with a twist. Instead of considering all features for splitting at each node, a random subset of features is selected.

This introduces variation among the trees, leading to a more diverse and robust model.

Voting/Averaging:

For classification tasks, each tree "votes" for a class, and the majority vote is taken as the final prediction.

For regression tasks, the predictions from all trees are averaged to produce the final result.



Illustration of training a Random Forest model. The training dataset (in this case, 250 rows and 100 columns) is randomly sampled with replacement n times. Then, a decision tree is trained on each sample. Finally, for prediction, the results of all n trees are aggregated to produce a final decision.

Key hyperparameters like the number of trees (n\_estimators), maximum depth of trees (max\_depth), and minimum samples per leaf (min\_samples\_leaf) are tuned using grid search or random search with cross-validation, utilizing the inherent feature importance scores from Random Forest to understand which features contribute most to the treatment decisions.

4. Training and Evaluation

The Random Forest classifier is trained on the training dataset using cross-validation to ensure the model generalizes well to unseen data. Overfitting is monitored by comparing performance on the training and validation sets.

Metrics like accuracy, precision, recall, and F1-score are used to evaluate the model. Precision and recall are essential for a recommendation system to ensure correct and relevant treatments are recommended.

5. System Integration

The model is integrated into a recommendation engine that suggests treatments based on new patient data. This engine can be accessed through an interface for healthcare providers to input patient data and receive treatment recommendations. The interface provides explanations for the recommendations to build trust.

**IMPLEMENTATION**

I will be implementing my methods stated above by first collecting data that contains records of cholera symptoms and treatments given in each case. The dataset used in this project and also source codes for the modeling and web app will be found in the GitHub attached to the reference section below. After this, data preprocessing is performed using Python programming language and a random forest algorithm. During the preprocessing stage, the symptoms and treatment data collected are being read from a CSV file which contains the data, after which the symptoms column and treatment columns are split into lists. These lists are then converted into a binary matrix for ease of computation.

The dataset is split into training and test sets, where 20% of the data collected is used for the testing and the other 80% is used to train the model of the system generated using random forest. Random forest is also initialized afterward in the code. Key hyperparameters like the number of trees (n\_estimators), maximum depth of trees (max\_depth), and minimum samples per leaf (min\_samples\_leaf) are tuned using grid search with cross-validation to understand which features contribute most to the treatment decisions.

The RandomForest model is then trained with the best parameters received from above and also, the test sets are predicted. The generated model is evaluated by checking for the accuracy score and also the report from the classification too. The trained model is then saved into a MyRecommendationModel.pkl file which can be integrated into software systems or APIs. In this case, we intend to integrate the model into a web app that can be accessible to anyone anywhere so far they are online. The web app was developed using Python Flask for the backend integration and HTML Bootstrap for a web interface.

After training the model on the dataset, I was able to get a trained model that has an 80% accuracy score which can be used to build a model using random forest classification, which can be used for the recommendation of treatment for cholera patients, providing support to doctors and medical practitioners when it comes to diagnosis of cholera patients. Below are the classification report metrics and overall performance evaluation for the model.

Classification Report Metrics:

* Precision: The proportion of correctly predicted positive observations to the total predicted positives.
* Recall (Sensitivity or True Positive Rate): The proportion of correctly predicted positive observations to all observations in the actual class.
* F1-Score: The weighted average of Precision and Recall. It considers both false positives and false negatives.
* Support: The number of true instances for each label.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Symptoms | Precision | Recall | F1 score | Support |
| Antibiotics | 0.75 | 0.78 | 0.76 | 27 |
| Electrolytes | 0.86 | 0.76 | 0.81 | 25 |
| IV Fluids | 0.97 | 0.85 | 0.90 | 39 |
| ORS | 1.00 | 1.00 | 1.00 | 90 |

The classification report above indicates the performance of the Random Forest model for each treatment category. Here's a breakdown of what the numbers mean:

Interpretation by Treatment Category:

* Antibiotics:
* Precision: 0.75 - Of all the instances where the model predicted "Antibiotics," 75% were correct.
* Recall: 0.78 - The model correctly identified 78% of the actual "Antibiotics" cases.
* F1-Score: 0.76 - The harmonic mean of precision and recall.
* Support: 27 instances.
* Electrolytes:
* Precision: 0.86 - The model was correct 86% of the time when predicting "Electrolytes."
* Recall: 0.76 - It correctly identified 76% of the actual "Electrolytes" cases.
* F1-Score: 0.81 - A good balance between precision and recall.
* Support: 25 instances.
* IV fluids:
* Precision: 0.97 - Very high precision, meaning the model almost always predicts "IV fluids" correctly.
* Recall: 0.85 - The model correctly identified 85% of actual "IV fluids" cases.
* F1-Score: 0.90 - Excellent performance in predicting "IV fluids."
* Support: 39 instances.
* Oral Rehydration Solution:
* Precision: 1.00 - Perfect precision, meaning every prediction of "Oral Rehydration Solution" was correct.
* Recall: 1.00 - The model identified all actual "Oral Rehydration Solution" cases.
* F1-Score: 1.00 - Perfect prediction.
* Support: 90 instances.

Overall model performance:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Averages | Precision | Recall | F1 Score | Support |
| Micro avg | 0.94 | 0.90 | 0.92 | 181 |
| Macro avg | 0.90 | 0.85 | 0.87 | 181 |
| Weighted avg | 0.94 | 0.90 | 0.92 | 181 |
| Samples avg | 0.94 | 0.90 | 0.91 | 181 |

Micro Avg:

* Precision, Recall, F1-Score: 0.94, 0.90, 0.92 - These metrics are calculated by aggregating the contributions of all classes, treating each label equally.

Macro Avg:

* Precision, Recall, F1-Score: 0.90, 0.85, 0.87 - These are the average metrics calculated for each label, not considering the support.

Weighted Avg:

* Precision, Recall, F1-Score: 0.94, 0.90, 0.92 - These averages account for the support of each class, so the performance on the "Oral Rehydration Solution" class (which has the most instances) heavily influences the average.

Samples Avg:

* Precision, Recall, F1-Score: 0.94, 0.90, 0.91 - These metrics average the performance per sample (instance), useful for multi-label classification.

The model performs exceptionally well, especially for "Oral Rehydration Solution" and "IV fluids," with near-perfect predictions. "Antibiotics" and "Electrolytes" show slightly lower performance, indicating room for improvement, possibly through more data or further tuning. The overall performance metrics suggest that the model generalizes well to unseen data with high precision and recall.

The model is integrated into a web app of two pages. The first page allows for user inputs and submission to get a recommended treatment and the second page displays the outcome ie the model's recommended treatment.

**SUMMARY, CONCLUSIONS AND RECOMMENDATIONS**

5.1 Summary

I developed a treatment recommendation system for cholera based on symptom data. The core of the system is a Random Forest Classifier trained on a dataset containing cholera symptoms and corresponding treatments. The system was built with the following key components:

Data Preprocessing:

The dataset was preprocessed to transform textual symptoms into a format suitable for machine learning models. I used `CountVectorizer` to convert symptoms into feature vectors and `MultiLabelBinarizer` to handle multi-label treatment outputs.

Model Training:

A Random Forest model was trained on the preprocessed data to predict possible treatments based on the symptoms provided. The model was evaluated for performance and then saved for future use.

Web Application:

A Flask web application was created to provide a user interface for the recommendation system. Users can input symptoms, and the system returns a list of recommended treatments. The web app uses Bootstrap for a clean and responsive design.

Model Deployment:

The trained model, along with the `CountVectorizer` and `MultiLabelBinarizer`, were saved using `joblib` and integrated into the Flask app, allowing the system to provide real-time predictions based on user input.

5.2 Conclusions:

1. Effective Use of Machine Learning:

The Random Forest model was effective in capturing the relationships between symptoms and treatments. Given its ability to handle complex interactions, it is well-suited for this type of multi-label classification problem.

2. Modular and Extensible Design:

- The system is modular, with clear separation between data preprocessing, model training, and web interface layers. This design allows for easy updates and maintenance.

3. User-Friendly Interface:

The web application, powered by Flask and styled with Bootstrap, provides a simple and accessible interface for users to interact with the model. The interface can easily be extended or modified based on user feedback or changing requirements.

5.3 Recommendations

1. Model Evaluation and Improvement:

While the Random Forest model performed adequately, further evaluation with different metrics and more data is recommended. Testing with cross-validation, hyperparameter tuning, and other models (e.g., Gradient Boosting) to ensure the best possible performance should be considered.

2. Data Expansion and Quality:

The system's accuracy is directly tied to the quality and quantity of the data. Expanding the dataset with more symptoms and corresponding treatments, possibly including data from other diseases for comparison, could improve the model's generalization and accuracy.

3. Real-World Testing and Feedback:

Deploying the system in a controlled real-world environment to gather user feedback. This will help identify any usability issues, and it may highlight additional symptoms or treatments that should be included.

4. Continuous Learning:

Implementing a mechanism to allow the system to learn continuously from new data. This could involve periodic model retraining with updated datasets or an active learning approach where the model asks for user input when unsure.

5. Security and Privacy:

As the system deals with health-related data, ensure that user data is handled securely. Implementing HTTPS for the web app, and considering adding authentication for access to the system if sensitive data is involved.

6. Scalability and Performance:

As the user base grows, one should consider optimizing the system for scalability. This might involve moving to a more robust web framework, using cloud services for deployment, or implementing load balancing to handle high traffic.

7. Integration with Medical Databases:

Integrating the system with existing medical databases or APIs to provide more comprehensive recommendations, including alternative treatments, medication information, and links to related medical literature should be considered.

By following these recommendations, the system can evolve into a more robust, accurate, and widely used tool for providing treatment recommendations based on symptoms, ultimately aiding in better patient outcomes.

**REFERENCES**

* Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
* Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.
* Maheshwari, A., & Arunesh, K. (2023). A recommendation system based on COVID-19 prediction & analysis using an ensemble-boosted machine learning algorithm. SN Computer Science, 4(1), 478. <https://doi.org/10.1007/s42979-023-01907-w>
* Lee, K. H., Choi, G. H., Yun, J., & et al. (2024). Machine learning-based clinical decision support system for treatment recommendation and overall survival prediction of hepatocellular carcinoma: A multi-center study. NPJ Digital Medicine, 7(2). <https://doi.org/10.1038/s41746-023-00976-8>.
* Dasgupta, S., & Saha, B. (2024). Big data analysis on medical field for drug recommendation using apriori algorithm and deep learning. Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-024-18832-6>
* Sun, X., Tang, H., Jiang, Y., & Zhang, Y. (2021). Predicting Chronic Disease Risks Using Decision Trees. Computers in Biology and Medicine, 133, 104400.
* Lambay, M. A., & Mohideen, S. P. (2024). Applying data science approach to predicting diseases and recommending drugs in healthcare using machine learning models: A cardio disease case study. Multimedia Tools and Applications, 83(6), 68341–68361. <https://doi.org/10.1007/s11042-023-18035-5>
* Wu, J., Chen, X., Zhang, X., & Li, J. (2018). A Hybrid KNN-ML Approach for Medical Treatment Recommendation. IEEE Journal of Biomedical and Health Informatics, 22(4), 1353-1360.
* Keikhosrokiani, P., Balasubramaniam, K., & Isomursu, M. (2024). Drug recommendation system for healthcare professionals’ decision-making using opinion mining and machine learning. In Särestöniemi, M., et al. (Eds.), Digital health and wireless solutions. NCDHWS 2024. Communications in computer and information science (Vol. 2084) Springer, Cham. <https://doi.org/10.1007/978-3-031-59091-7_15>
* Han, J., Kamber, M., & Pei, J. (2011). Data mining: Concepts and techniques. Elsevier.
* Singh, A., Choudhary, A., & Malhotra, S. (2020). Ensemble Methods for Disease Diagnosis: Combining KNN and Decision Trees. Journal of Healthcare Informatics Research, 4(2), 189-210.
* Rustam, F., Imtiaz, Z., Mehmood, A., & et al. (2022). Automated disease diagnosis and precaution recommender system using supervised machine learning. Multimedia Tools and Applications, 81(31), 31929–31952. <https://doi.org/10.1007/s11042-022-12897-x>
* Deng, X., Zhang, X., & Song, Y. (2016). A clinical decision support system for infectious disease diagnosis based on decision tree. IEEE Access, 4, 6133-6144.
* Borra, S., & Ciaccio, A. D. (2010). A KNN-based approach to predict patient responses to treatment. Medical & Biological Engineering & Computing, 48(3), 269-274.
* Chen, C., Liaw, A., & Breiman, L. (2018). Using random forest to learn imbalanced data. Journal of Machine Learning Research, 99(1), 149-168.
* Dataset used and project souce code <https://github.com/kennylegal/Cholera-Recommendation-System>