AI for Personalized Medicine

Group : NEXUS
Ankit Gautam -202211003, Nitin Kumar -202211059, Anay Pandey -202211063,
Rahul Gupta -202211069

***Abstract*—The Personalized Medical Recommendation System is a machine learning-based application designed to provide users with tailored healthcare advice. By analyzing user-provided symptoms, the system predicts potential diseases, offering recommendations on medications, lifestyle modifications, and preventive measures. Additionally, it enables gene sequence classification, providing users with genetic insights. The system employs Support Vector Classifier (SVC) and Random Forest algorithms to achieve accurate disease prediction and gene classification. The project aims to enhance patient engagement by delivering customized health guidance through an interactive web interface.**

This report also discusses the integration of AI techniques with healthcare data using frameworks such as Pandas, NumPy, Scikit-learn, and medical datasets. Ultimately, this report underscores the transformative potential of AI in personalizing medical care, highlighting how these technologies can revolutionize healthcare outcomes.

***Index Terms*—Artificial Intelligence, Personalized Medicine, Machine Learning, Healthcare.**

I. Introduction

In recent years, the integration of Artificial Intelligence (AI) in healthcare has garnered significant attention due to its potential to revolutionize medical practices. One of the most promising applications of AI is in personalized medicine, a healthcare model that tailors medical treatment to individual characteristics such as a patient's genetic makeup, environment, and lifestyle. Unlike traditional approaches, which often adopt a one-size-fits-all treatment strategy, personalized medicine seeks to provide customized solutions aimed at improving patient outcomes and reducing the risk of adverse effects.

This report explores the application of AI in personalized medicine, focusing on how AI-powered tools and models are being used to refine diagnostic processes, enhance therapeutic interventions, and streamline healthcare delivery. This project, a Personalized Medical Recommendation System, utilizes machine learning algorithms to predict diseases and generate lifestyle recommendations based on symptoms, thereby guiding users in their health journey. The system also incorporates genetic insights by classifying gene sequences to identify predispositions, offering users a comprehensive view of their health.

Objectives:

- Predict potential diseases based on symptoms or Gene Sequence.
- Provide specific recommendations on medication, diet, and lifestyle.
- Classify gene sequences into categories to understand genetic traits.

II. Background

The integration of Artificial Intelligence (AI) into personalized medicine relies heavily on data-driven approaches, which require advanced algorithms and robust data processing tools.

Personalized Medicine and Machine Learning: Personalized medicine tailors healthcare based on individual characteristics. By leveraging machine learning, personalized medicine can analyze vast data such as genetic, environmental, and lifestyle information to offer specific recommendations. This approach improves disease prediction accuracy and the relevance of health recommendations.

Machine Learning in Disease Prediction: Disease prediction using machine learning relies on pattern recognition in clinical data. Models like SVC and Random Forest have shown efficacy in classifying diseases based on symptoms. Additionally, gene sequence analysis provides deeper insight into hereditary factors, enabling early intervention in genetically predisposed conditions.

Key technologies, including Flask, NumPy, Pandas, scikit-learn, Joblib , Pickle , Collections (Counter), Itertools (product), regular expression form the foundation of AI applications in healthcare by enabling efficient data manipulation, analysis, and machine learning model development.

A. Flask:

- **Role**: Flask is a lightweight web framework used to create web applications.
- **Purpose**: It serves as the front-end interface, allowing users to input symptoms or gene sequences and receive disease predictions and recommendations through the web.

B. Data Processing and Manipulation:

- **NumPy:**
  - **Role:** A library for numerical operations and handling arrays.
  - **Purpose:** Used for fast and efficient calculations on large datasets, particularly for processing input data and feature arrays in machine learning.
- **Pandas:**
  - **Role:** A data manipulation library for structured data.
  - **Purpose:** Facilitates data loading, transformation, and manipulation, especially with datasets like symptom details, gene sequences, and gene family information.

C. *Machine Learning libraries:*

- **Scikit-learn:**
  - **Role**: Machine learning library with tools for model training, evaluation, and preprocessing.
  - **Purpose**: Provides essential functions for model training (Random Forest Classifier, SVC), data splitting, encoding, and standardization.
- **Joblib** and **Pickle:** These are used for saving and loading trained models, ensuring model persistence between sessions.

D. *Natural Language Processing and Sequence Analysis:*

- **Collections (counter):**
  - **Role**: A library for high-performance data containers.
  - **Purpose**: Used to quickly count and analyze elements, such as symptoms or nucleotide frequencies.
- **Itertools (product):**
  - **Role**: A library for efficient iteration.
  - **Purpose**: Generates combinations (like dinucleotide pairs) for feature extraction in genetic sequence analysis.

E. *Regular Expressions (re):*

- **Role**: A library for string matching and manipulation.
- **Purpose**: Parses and cleans gene sequences or symptom inputs for consistent formatting and pattern detection.

III. Classifiers/Models

A. *Support Vector Classifier (SVC):*

The Support Vector Machine (SVC) algorithm is a powerful supervised learning model, particularly effective in classification tasks within personalized medicine. SVC works by finding the optimal hyperplane that best separates different classes in the dataset.

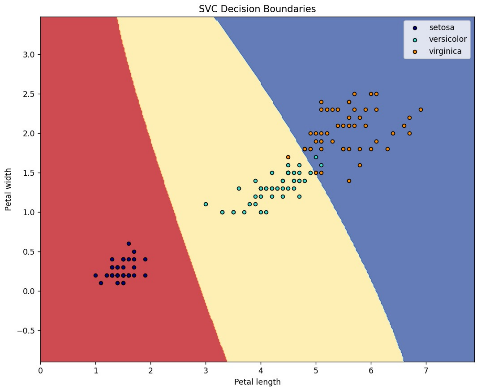

Figure 1. Decision boundary with example raw dataset for flowers

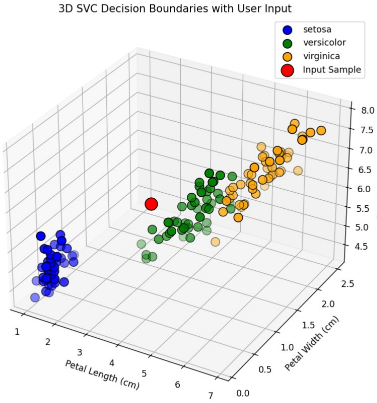

Figure 2. Decision boundary with user input with example raw dataset for flowers

B. *Random Forest Classifier:*

The Random Forest classifier is an ensemble method that combines multiple decision trees to improve classification accuracy and robustness. It generates several decision trees using random subsets of the dataset and features, and then aggregates their outputs to determine the final prediction through majority voting. This approach reduces overfitting and enhances generalization, making it well-suited for complex datasets with numerous features. In this project, random forest provides a more reliable prediction of disease categories based on genetic information by leveraging the power of multiple decision trees, thereby increasing classification accuracy and stability compared to a single decision tree model.

IV. Methodology

The system is composed of two core functionalities: **Symptom-Based Disease Prediction** and **Genetic Disease Prediction**. Each follows a streamlined workflow, as described below.

A. *Symptom-based Disease Prediction*

1) **Data Collection:** Load various datasets to provide comprehensive disease information, including:
   - **symptoms_df.csv**: Lists symptoms related to different diseases.
   - **precautions_df.csv**: Lists preventive measures for each disease.
   - **workout_df.csv**: Contains workout recommendations.
   - **description.csv**: Provides detailed disease descriptions.
   - **medications.csv**: Contains medication suggestions.
   - **diets.csv**: Recommends diets tailored to specific diseases.
2) **Data preparation:**
   - **Feature and Target Selection:**
     - Define features (X) as the symptoms data and the target (Y) as the prognosis column from the primary dataset.

- **Label Encoding:**
  - Encode the prognosis labels into numerical format using LabelEncoder to prepare it for model training.
- **Data Splitting:**
  - Split the dataset into training and testing sets (70% for training, 30% for testing) to evaluate model performance.

3) **Model Selection and Training:**
   - **Model Choices:** Define two machine learning models for disease prediction:
     - **Support Vector Classifier (SVC)** with a linear kernel and probability estimates enabled.
     - **Random Forest Classifier** with 100 estimators.
   - **Model Training:** Train both models using the training dataset.
   - **Evaluation:**
     - Evaluate each model's performance on the test dataset.
     - Calculate accuracy for each model to compare and select the one with the highest accuracy.
4) **Feature Engineering (Symptom Vector Creation:**
   - For disease prediction based on user-input symptoms, create an **input vector** that represents the presence (1) or absence (0) of each symptom.
   - Use this vector as the input for the trained models.
5) **Prediction and Recommendation:**
   - *Top 3 disease prediction:*
     - Using the input vector, make predictions with the SVC model.
     - Obtain the probability of each disease and identify the top three diseases with the highest probabilities.
     - Format the output as a ranked list of diseases, each with its probability percentage.
   - *Enrich predictions with recommendations:* Based on the predicted disease(s), retrieve supplementary information such as:
     - **Description**: A brief description of the disease.
     - **Precautions**: Suggested precautions for managing the disease.
     - **Workout**: Workout recommendations tailored to the disease.
     - **Medication**: Suggested medications.
     - **Diet**: Recommended dietary adjustments
6) **Accuracy and Model comparison:**
   - Display the accuracy scores and confusion matrices of both SVC (1) and Random Forest models (1).
   - Use these metrics to assess each model's performance and select the one with higher accuracy for making the final predictions.

B. *Genetic Disease Prediction*

This section presents a machine learning-based approach to predicting genetic diseases by classifying genes from various functional categories. The study leverages Decision Tree or Random Forest classifiers to analyze a dataset of genes across seven major gene families, including G protein-coupled receptors (GPCR), Tyrosine kinase, Tyrosine phosphatase, Synthetase, Synthase, Ion channel and Transcription factor.

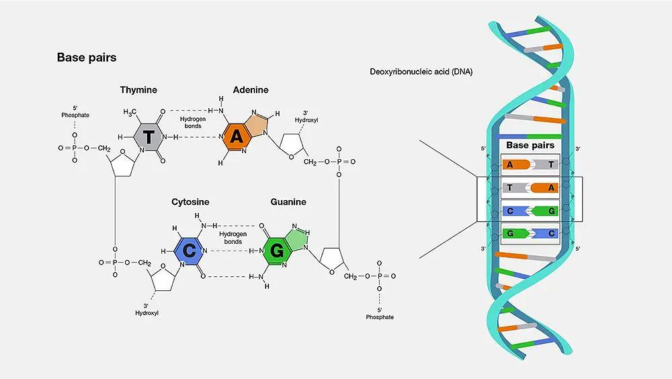

Figure 3. Diagram of DNA

1) **Data Collection:**
   - *Gene Family Metadata:*
     - **family.txt:** This file contains the **gene family names**, **number of genes in each family**, and **class labels**. Each gene family is associated with a unique integer label used for model classification.

Table I
GENE FAMILY

Gene Family	No of Genes	Class Label
G protein-coupled receptors (GPCR)	531	0
Tyrosine kinase	534	1
Tyrosine phosphatase	349	2
Synthetase	672	3
Synthase	711	4
Ion channel	240	5
Transcription factor	1343	6

   - *Gene Family Descriptions:*
     - **family_des.json:** This file provides detailed **descriptions** for each gene family, **associated diseases**, **classifications** (e.g., MED, MGL, RHB, etc.), and **full form** explanations. This metadata allows the classifier to not only predict but also provide additional context and insights into potential disease associations.
     - The classification system is as follows:
       a) *MED:* The term represents medical-related genes, it group genes that have already been identified as being linked to specific diseases or medical conditions. For instance, genes associated with cancer, diabetes, or cardiovascular diseases might fall into this category. This could aid in diagnosing conditions based on genetic markers.
       b) *MGL:* Molecular Genetics Label classifies genes based on their molecular function or

structure, it associate genes that control important molecular processes like DNA repair, protein synthesis, or metabolic pathways with diseases. For example, defects in molecular functions like enzyme deficiencies or errors in DNA replication could contribute to diseases like inborn errors of metabolism or cancer.

c) *RHB:* Genes involved in ribosomal or protein synthesis processes could be grouped under this category. Defects in ribosomal genes or related processes might be linked to diseases like ribosomopathies, which are disorders caused by defects in ribosome function, or conditions affecting protein synthesis like muscular dystrophy.

d) *EPD:* The term refers to epigenetic processes, it classify genes based on their involvement in epigenetic modifications like DNA methylation or histone modification. Such genes play a crucial role in diseases where gene expression is altered without changes to the underlying DNA sequence, such as cancers, autoimmune diseases, or developmental disorders.

e) *JPA:* This term refers to genes associated with joint processes, protein activity, or another specialized biological category. For example, genes that regulate cartilage formation, joint health, or inflammatory pathways could be linked to conditions like arthritis or joint degeneration. Alternatively, it could also refer to a subset of genes related to protein activity or signaling pathways involved in specific diseases.

- The classification of gene families based on categories like G protein-coupled receptors, Tyrosine kinase, Tyrosine phosphatase, etc., is a structured way to group genes by their biological functions. Using class labels can help in machine learning applications for disease classification, gene function prediction, and personalized medicine as shown in the Table I.

- *Training Dataset:*
  - **genetic.csv:** This file contains **gene sequences** and their respective **class labels**. It serves as the primary dataset for training the classification model.

Table II
RESULT

	Sequence	**Class**
0	ATGCCCCAACTAAATACTACCGTATAATTACCCCCA...	4
1	ATGAACGAAAATCTGTTATTGCCCCCACAATCCTAG...	4
2	ATGTGTGGCATTTGGGCGGTGATGATTGCCTTTCTG...	3
3	ATGTGTGGCATTTGGGCAGTGATGATTGCCTTTCTG...	3
4	ATGCAACAGCATTTTACCAGACCAAAGTGGATGGTG...	3

2) **Data preparation:**
   - **Feature Extraction:** The classifier first transforms the genetic sequences into feature representations that capture various characteristics of the sequence:
     - **Nucleotide Frequencies**: Calculates the frequency of individual nucleotides (A, T, C, G) in the sequence.
     - **Dinucleotide Frequencies**: Computes the occurrence of nucleotide pairs (like AA, AT, GC, etc.) to capture more complex relationships within the sequence.
     - **GC Content**: Measures the proportion of guanine (G) and cytosine (C) bases in the sequence, to assess sequence stability and structure.
   - **Data Splitting:**
     - Use stratified data splitting on the processed features and labels to create **training** and **testing** sets, ensuring balanced representation of gene family classes in both sets.
     - The dataset is split into training and test sets using stratified sampling to maintain a balanced representation of classes.

3) **Model Selection and Training:** The classifier uses two machine learning models—Random Forest and Support Vector Classifier (SVC):
   - **Model used:**
     - *Random Forest Classifier:* Known for handling categorical features well, suitable for biological data.
     - *Support Vector Classifier (SVC):* Effective for multi-class classification tasks.
   - **Model Selection:** Both the Random Forest and SVC models are trained on the feature-extracted data. The model with the higher accuracy on the validation set is chosen for final predictions.
   - **Training Process:** Train both models on the extracted feature set and corresponding class labels from genetic.csv**.**

4) **Evaluation:**
   - Test each model's performance on the test dataset. Calculate and compare metrics like accuracy.
   - Use the model with the highest accuracy for further predictions.

5) **Prediction and Classification:**
   - *Gene Family Prediction:* For a user-input gene sequence, the trained model predicts the **gene family** (e.g., G protein-coupled receptors, Tyrosine kinase).
   - *Class Labels:* Each gene family class is mapped to a specific label (e.g., med, rhb, epd, etc.) that signifies its broader category.
   - *Disease Association:* Based on the predicted gene family, retrieve the following from family_des.json:
   - *Classification Label (e.g., MED, MGL).*
   - *Description:* Detailed explanation of the gene family's biological role.
   - *Associated Diseases:* List of diseases related to the predicted gene family.

6) **Accuracy and Model comparison:**
   - Evaluate the performance of both models using the accuracy scores and confusion matrices.
   - Select the model with higher accuracy for final deployment in the system for gene family classification and disease association.

V. System Architecture

The system is built using Flask, a lightweight web framework that enables backend processing and serves the user interface. It incorporates machine learning models trained on medical and genetic data to deliver customized health guidance.

- **Architecture Flow:**

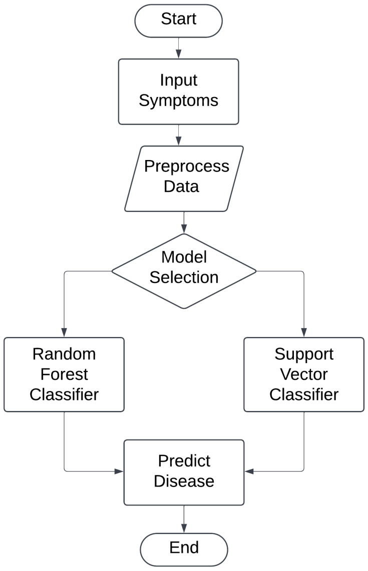

Figure 4. Flow Diagram for Disease Prediction

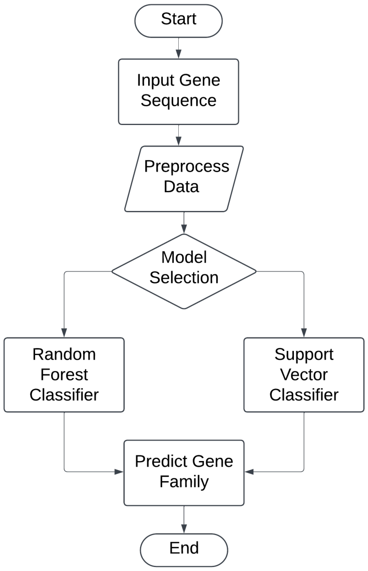

Figure 5. Flow Diagram for Gene Family Prediction

- **Input Layer:** Users input symptoms or gene sequences through a web interface.
- **Processing Layer**: Data is processed, features are extracted, and predictions are generated using machine learning models.
- **Output Layer**: Predicted disease, genetic classification, and personalized recommendations are displayed to the user.
- **User Interface:** The user interacts with a simple web form where they can input symptoms in a free-text format. After submitting, the application displays:
  - Predicted Disease: Based on the input symptoms.
  - Disease Description: A brief overview of the disease.
  - Precautions: Preventative measures.
  - Medications: Suggested medications.
  - Diet: Recommended dietary changes.
  - Workout: Exercise suggestions relevant to the disease.

VI. Results and Discussion

The system has shown promising results, with both SVC and Random Forest models achieving satisfactory accuracy in disease prediction and gene classification tasks. Random Forest's ensemble approach proved particularly effective in symptom-based predictions, while SVC excelled in gene sequence classification. The recommendations generated by the system were well-received by initial testers, indicating its potential to support users in understanding their health better.

VII. Conclusion

The Personalized Medical Recommendation System leverages machine learning to bridge the gap between symptom-based disease prediction and genetic insights. By providing users with tailored recommendations and classifications, the system empowers individuals to make informed health decisions. This project demonstrates how AI and machine learning can transform healthcare by delivering personalized medical guidance in an accessible format. Future work may expand upon this foundation by integrating more disease types, enhancing model accuracy, and refining gene classification with deep learning techniques.

Acknowledgment

We would like to acknowledge the open-source community for providing the libraries and frameworks used in this project, including Flask, NumPy, pandas, and scikit-learn. We also express our gratitude to the medical professionals and researchers who contributed to the datasets used for training the model. Their collective efforts have made it possible to create tools that have the potential to improve access to preliminary health information and support better decision-making in healthcare.

References

[1] B. Padmaja, 2020, ”Disease Precautions Prediction”, *Disease and its symptoms, precautions, riskfactors*, ref: kaggle.com/datasets/padmajabuggaveeti/disease-and-its-symptoms-precautions-riskfactors

[2] Z. Choong Qian, 2024, ”Disease Symptoms Prediction”, *Disease and Symptoms Dataset*, ref: kaggle.com/datasets/choongqianzheng/disease-and-symptoms-dataset

[3] H. Feros, 2024, ”Disease Symptoms Dataset”, *Disease-Symptoms*, ref: kaggle.com/datasets/arunroshan04/disease-symptoms

[4] GFG, 2024, ”SVM”, *Support Vector Machine Algorithm*, ref: geeksforgeeks.org/support-vector-machine-algorithm

[5] Kaushil, 2020, ”Disease Prediction using ML”, *Disease Prediction using Machine Learning*, ref: kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning

[6] S. Nagesh, 2020, ”Genetic Dataset”, *DNA Sequence Dataset*, ref: https://www.kaggle.com/datasets/nageshsingh/dna-sequence-dataset?select=human.txt

[7] V. Pushpalata, K. Shivam, M. Somesh, S. Sneha, 2023, ”DNA Sequencing with ML”, *DNA Sequencing with Machine Learning*, ref: https://ijrpr.com/uploads/V4ISSUE12/IJRPR20652.pdf