IS 733 DATA MINING

FINAL PROJECT REPORT

“ HEALTHMATE ”

[PERSONALIZED MEDICAL RECOMMENDATION SYSTEM USING MACHINE LEARNING]

Submitted By

Group 5:

PRATHYUSHA HARISH KUMAR

SRI KAVYA PENTA

ARYAN JAGANI

SAI TARUN DANTULURI

TABLE OF CONTENTS

ABSTRACT

The evolution of digital health technologies has opened new frontiers in personalized medicine, enabling proactive, data-driven approaches to healthcare delivery. In response to the growing prevalence of chronic illnesses, limited access to timely clinical care, and the increasing demand for intelligent healthcare systems, this project presents a **Machine Learning-Based Personalized Medical Recommendation system** that predicts potential diseases from user-reported symptoms and provides targeted health guidance.

The proposed system is built upon a supervised learning model trained on a comprehensive symptoms-disease dataset. The dataset includes a variety of symptom inputs mapped to known medical conditions, enabling the model to learn and generalize patterns that associate specific combinations of symptoms with potential diagnoses. The data underwent meticulous preprocessing, including normalization, encoding of categorical features, and handling of missing values, to enhance model quality and reduce noise. Feature selection techniques were employed to retain only the most relevant attributes, ensuring efficient and accurate model learning.

Multiple machine learning algorithms were evaluated for performance, including Decision Trees, Random Forests, and Support Vector Machines. Among them, Random Forest was selected as the optimal model due to its robustness, interpretability, and superior performance across key evaluation metrics such as accuracy, precision, recall, and F1-score.

Upon receiving a user's symptom input, the system predicts the most probable disease and subsequently generates a set of **Personalized Medical Recommendations**. These recommendations are curated based on the disease prediction and are categorized into three key domains: suggested medications (non-prescriptive, general drug names for awareness), prescription guidance (advice on when to seek medical attention), and lifestyle recommendations (including dietary, hydration, and physical activity tips tailored to the condition). The logic for recommendation generation draws upon publicly available health guidelines and research-informed best practices to ensure relevance and safety.

The objective of this system is not to replace medical professionals, but to serve as a preliminary health assistant that empowers individuals to take informed actions and seek timely professional care. The tool is especially useful in remote or underserved regions where healthcare access is limited, offering users immediate feedback and guidance based on data-driven analysis. Additionally, by promoting early detection and intervention, the system can contribute to better long-term health outcomes and reduced healthcare costs.

In conclusion, this project showcases the transformative potential of machine learning in healthcare by demonstrating how intelligent systems can be leveraged to deliver scalable, accessible, and personalized medical insights. It stands as a step toward democratizing healthcare through technology, putting critical health knowledge and guidance into the hands of individuals when they need it most.

1. INTRODUCTION

Healthcare is a significant aspect of human well-being, and early identification of illness is one of the most important elements of effective treatment. Access to immediate medical consultation, however, is typically limited by location, affordability, or availability of medical experts in far-flung areas. To address this issue, this project aims to develop a machine learning system of personalized medical recommendation from symptom input. The system utilizes sophisticated classification models such as Random Forest in predicting possible diseases from symptoms input by the user.

The **Personalized Medical Recommendation System** leverages advanced machine learning techniques to provide tailored healthcare guidance. By integrating user-provided symptoms and health data, this system predicts potential diseases, recommends personalized medications, and suggests suitable workout routines. This system improves user experience by utilizing natural language processing (NLP) mechanisms such as a synonym dictionary to map user-friendly symptom descriptions to medical terms and auto-spelling correction to minimize input errors. By optimizing model performance through feature selection, hyperparameter tuning, and data preprocessing, this method provides high accuracy and reliability in disease prediction and treatment recommendation.

Healthcare accessibility and accuracy in preliminary diagnoses remain significantly challenging. Many individuals lack immediate access to medical professionals, leading to delayed treatment or reliance on inaccurate self-diagnoses. The proposed system aims to bridge this gap by providing a user-friendly, intelligent medical assistant that offers preliminary health insights and personalized health recommendations.

2. BACKGROUND

Artificial Intelligence (AI) and Machine Learning (ML) has significantly transformed the landscape of modern healthcare, offering tools that go beyond traditional diagnostics and treatment planning. With the proliferation of health data from clinical records, wearable devices, and patient-reported symptoms, there is a growing need for intelligent systems capable of extracting actionable insights from these large and complex datasets. Among the most impactful AI applications are Personalized Medical Recommendation Systems (PMRS), which use predictive modeling and pattern recognition to offer individualized health advice based on a user’s symptoms and health profile.

Machine learning algorithms such as Decision Trees, Random Forests, Naïve Bayes, and Neural Networks are commonly used in clinical informatics to support predictive diagnosis and risk stratification. These models are particularly valuable in scenarios where rapid decision-making is critical and real-time access to professional medical consultation may be limited. Unlike traditional rule-based systems, ML-driven recommendations adapt and improve over time, learning from data patterns to offer more accurate and context-sensitive advice.

The integration of these systems into digital health ecosystems represents a paradigm shift toward proactive and personalized care. In particular, PMRS can bridge gaps in early detection by analyzing combinations of user symptoms to predict possible diseases and provide non-clinical recommendations that encourage timely medical intervention.

PROBLEM STATEMENT

Previous work in this domain has primarily focused on disease-specific applications such as diabetes prediction, heart disease monitoring, or cancer classification using clinical and imaging datasets. However, there remains a significant opportunity to develop generalized systems that cater to a wider array of conditions using simpler, more accessible inputs—such as symptoms self-reported by users. These models rely on structured datasets where symptoms are mapped to known diseases, allowing for the creation of scalable, data-driven solutions that can be deployed across web or mobile platforms.

This project aims to build such a system by leveraging supervised learning models trained on symptom-based datasets. It demonstrates how intelligent health assistants can offer meaningful guidance to users through the combination of accurate disease prediction and context-aware recommendations. By prioritizing interpretability, scalability, and user accessibility, this approach contributes to the broader mission of enhancing patient autonomy in health management.

3. METHODOLOGY

The main chosen methodology to develop this project is CRISP-DM [Cross-Industry Standard Process for Data Mining]. This is a methodology for the development of the deep learning model, as it provides a greater understanding on how to deliver this model and this comprises six major phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment.

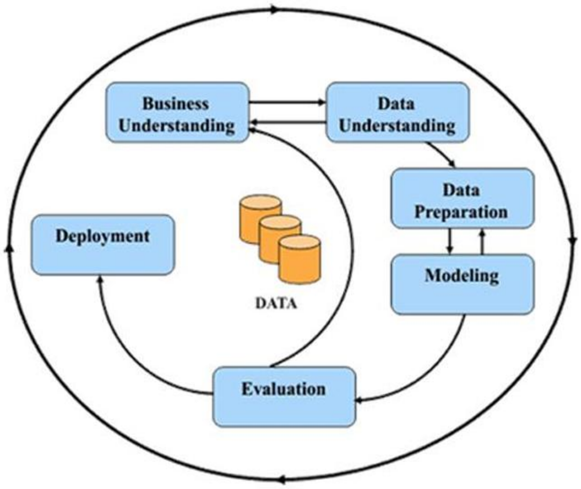


Fig 1: CRISP-DM Methodology

According to Shearer (2000), CRISP-DM is a non-proprietary, documented and freely available data mining model. It offers an industry tool and application neutral model that encourages best practices and offers organizations the structure needed to realize better and faster results from data mining. Figure 1 displays the 6 phases of CRISP-DM. According to Shapiro, CRISP-DM is the leading methodology for analytics, data mining or data science projects. The next sections will give a brief overview of each step and how it is being utilized in this project.

3.1 BUSINESS UNDERSTANDING

The primary objective of this project is to develop an intelligent system that can predict potential diseases based on user-input symptoms and offer personalized medical recommendations, including suggested medications, prescriptions, and lifestyle advice. The system is designed to enhance early detection, support preliminary health decisions, and empower users with accessible medical insights, particularly in environments with limited access to professional healthcare services.

KEY GOALS:

- Predict diseases based on symptoms using a trained machine learning model.
- Provide personalized recommendations tailored to the predicted condition.
- Ensure interpretability, accuracy, and user trust.

3.2 DATA UNDERSTANDING

Understanding the structure, composition, and characteristics of the dataset is a critical step in any machine learning project. In this personalized medical recommendation system, the goal is to predict potential diseases based on user-reported symptoms and offer appropriate recommendations. Achieving high accuracy in this task requires thorough familiarity with the dataset and an awareness of potential challenges, such as imbalance, redundancy, or noise.

The dataset used for this project is a curated collection of symptom-disease mappings from **CBC Healthcare System**, commonly used in health informatics research. It includes thousands of records, each representing an individual patient case with a set of reported symptoms and a corresponding diagnosed disease. The features consist of around **130 symptoms**, encoded as binary variables (1 for presence, 0 for absence). The **target variable** is the disease class, encompassing over **40 unique conditions**, including both common ailments like flu and cold as well as more complex diseases like hepatitis, tuberculosis, and diabetes.

The first step in understanding the dataset involved exploratory data analysis (EDA). This helped uncover the distribution of symptoms and disease labels, detect any data quality issues, and identify patterns that could influence model performance.

Feature validation checks confirmed that all symptom entries were within expected binary ranges (0 or 1), ensuring the data's structural integrity. Redundancy checks were also conducted to eliminate duplicate records, which could introduce bias during model evaluation. Additionally, co-occurrence matrices and heatmaps were used to visualize relationships between symptoms, allowing for the identification of symptom clusters commonly associated with certain disease classes.

This phase revealed key insights into the dataset's characteristics:

- The high dimensionality (130+ symptoms) necessitates robust algorithms capable of handling sparse binary vectors.
- Some features contribute more to prediction than others; thus, **feature importance** evaluation or dimensionality reduction can enhance performance.
- The dataset supports multi-class classification and requires handling of **overlapping symptom sets**, which is common in real clinical scenarios.

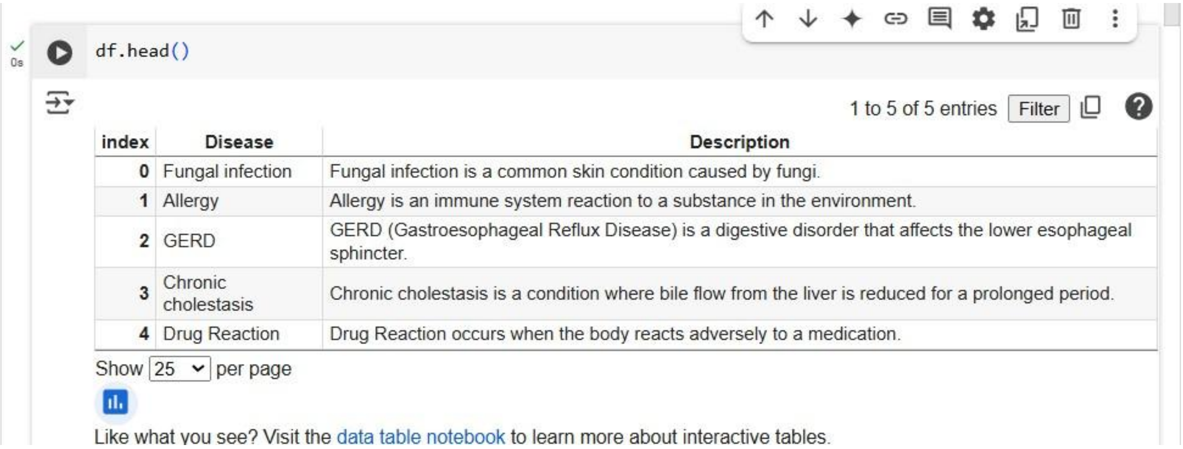
By thoroughly analyzing and understanding the dataset, the foundation was laid for building a machine learning model that is accurate, interpretable, and effective in a healthcare setting.

DATA ANALYSIS:

- **Total Data Points:** The Training.csv file contains 4920 data entries (rows), each representing a unique case or instance linking symptoms to a prognosis.
- **Attributes:** The training dataset consists of 133 columns (attributes).
    - 132 columns represent different symptoms. These features are primarily binary (0 or 1), indicating the absence or presence of a specific symptom for that data entry.
    - 1 column (prognosis) represents the target variable, containing the name of the diagnosed disease.
- **Number of Classes:** The target variable prognosis has 41 unique values, meaning the model is trained to predict one of 41 distinct diseases.

We have 6 different files of the data sets for our project.

1. Description Dataset



```
df.head()
```

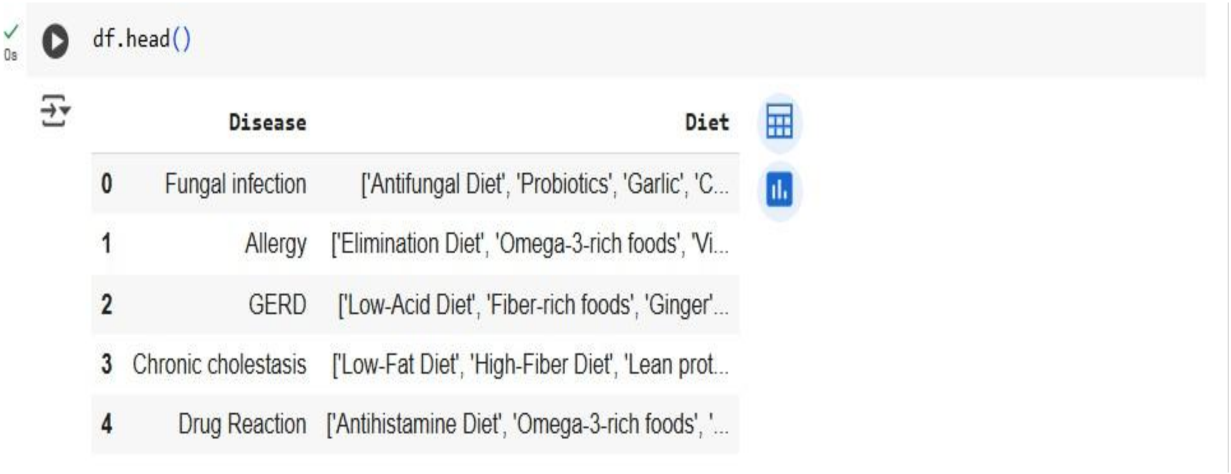
1 to 5 of 5 entries

index	Disease	Description
0	Fungal infection	Fungal infection is a common skin condition caused by fungi.
1	Allergy	Allergy is an immune system reaction to a substance in the environment.
2	GERD	GERD (Gastroesophageal Reflux Disease) is a digestive disorder that affects the lower esophageal sphincter.
3	Chronic cholestasis	Chronic cholestasis is a condition where bile flow from the liver is reduced for a prolonged period.
4	Drug Reaction	Drug Reaction occurs when the body reacts adversely to a medication.

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

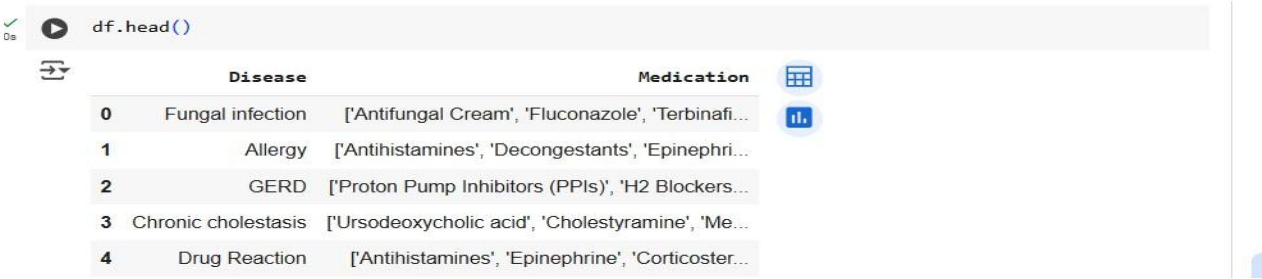
2. Diet Dataset



```
df.head()
```

	Disease	Diet
0	Fungal infection	['Antifungal Diet', 'Probiotics', 'Garlic', 'C...
1	Allergy	['Elimination Diet', 'Omega-3-rich foods', 'Vi...
2	GERD	['Low-Acid Diet', 'Fiber-rich foods', 'Ginger'...
3	Chronic cholestasis	['Low-Fat Diet', 'High-Fiber Diet', 'Lean prot...
4	Drug Reaction	['Antihistamine Diet', 'Omega-3-rich foods', '...

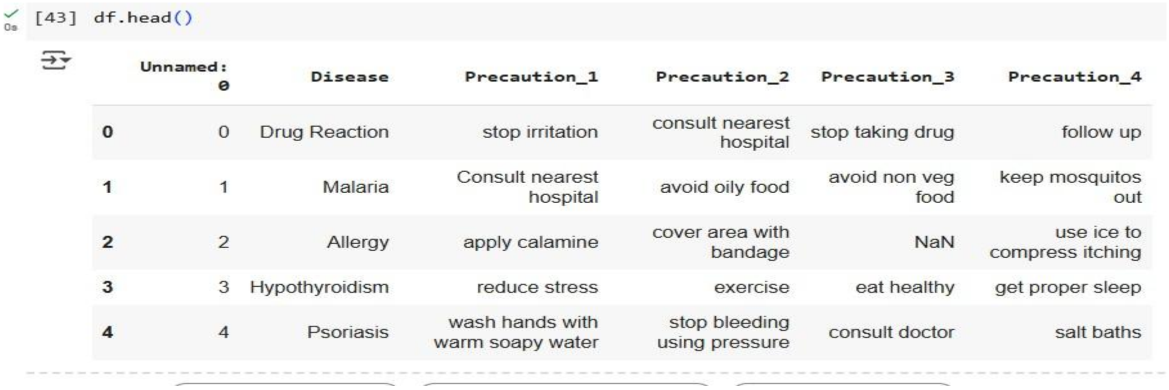
3. Medication Dataset



```
df.head()
```

	Disease	Medication
0	Fungal infection	['Antifungal Cream', 'Fluconazole', 'Terbinafi...
1	Allergy	['Antihistamines', 'Decongestants', 'Epinephri...
2	GERD	['Proton Pump Inhibitors (PPIs)', 'H2 Blockers...
3	Chronic cholestasis	['Ursodeoxycholic acid', 'Cholestyramine', 'Me...
4	Drug Reaction	['Antihistamines', 'Epinephrine', 'Corticoster...

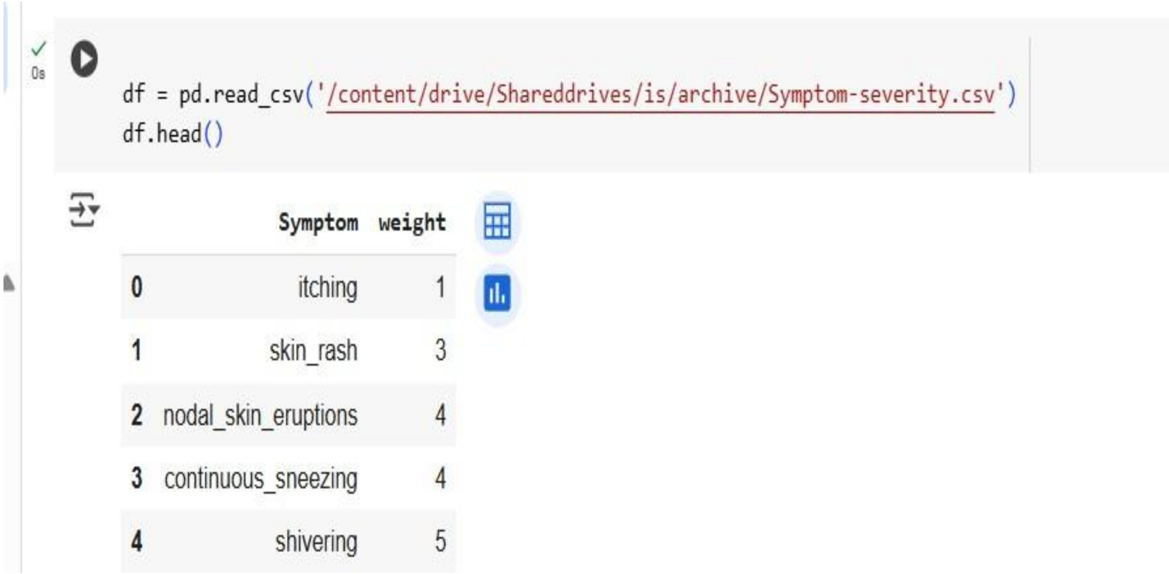
4. Precaution Dataset



```
[43] df.head()
```

	Unnamed: 0	Disease	Precaution_1	Precaution_2	Precaution_3	Precaution_4
0	0	Drug Reaction	stop irritation	consult nearest hospital	stop taking drug	follow up
1	1	Malaria	Consult nearest hospital	avoid oily food	avoid non veg food	keep mosquitos out
2	2	Allergy	apply calamine	cover area with bandage	NaN	use ice to compress itching
3	3	Hypothyroidism	reduce stress	exercise	eat healthy	get proper sleep
4	4	Psoriasis	wash hands with warm soapy water	stop bleeding using pressure	consult doctor	salt baths

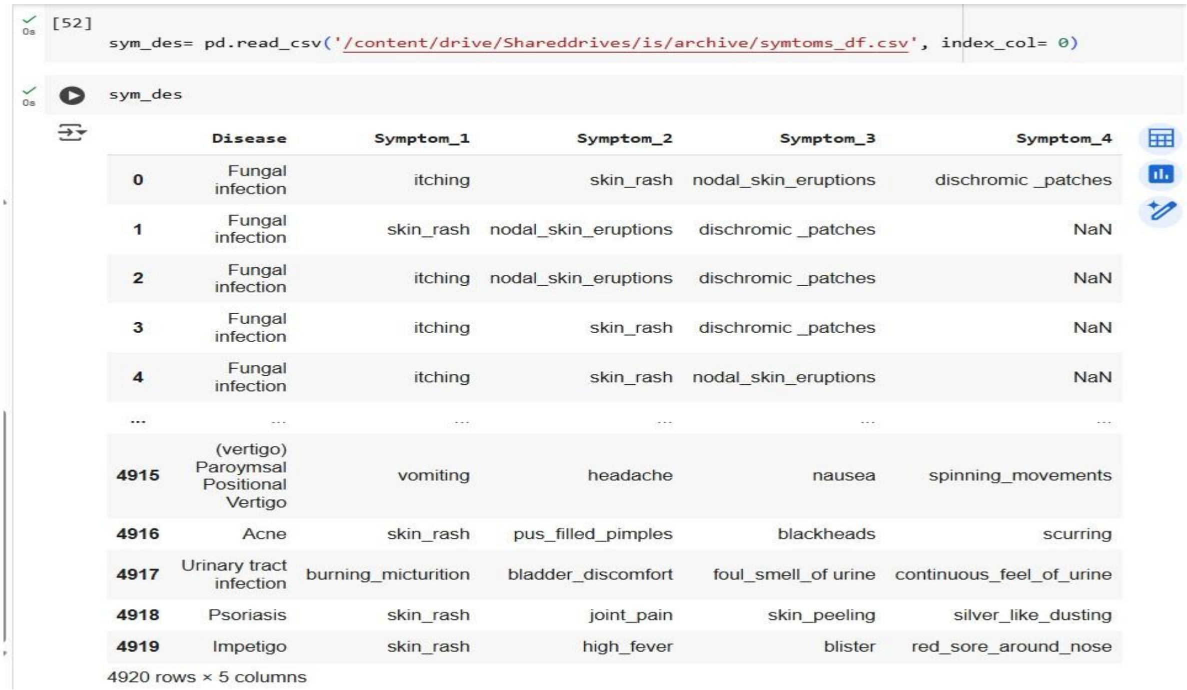
5. Symptom Severity Dataset



```
df = pd.read_csv('/content/drive/Shareddrives/is/archive/Symptom-severity.csv')
df.head()
```

	Symptom	weight
0	itching	1
1	skin_rash	3
2	nodal_skin_eruptions	4
3	continuous_sneezing	4
4	shivering	5

6. Symptom Dataset



```
[52]
sym_des= pd.read_csv('/content/drive/Shareddrives/is/archive/symtoms_df.csv', index_col= 0)
```

```
sym_des
```

	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	NaN
3	Fungal infection	itching	skin_rash	dischromic _patches	NaN
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	NaN
...	...	...	...	...	...
4915	(vertigo) Paroymsal Positional Vertigo	vomiting	headache	nausea	spinning_movements
4916	Acne	skin_rash	pus_filled_pimples	blackheads	scurring
4917	Urinary tract infection	burning_micturition	bladder_discomfort	foul_smell_of urine	continuous_feel_of_urine
4918	Psoriasis	skin_rash	joint_pain	skin_peeling	silver_like_dusting
4919	Impetigo	skin_rash	high_fever	blister	red_sore_around_nose

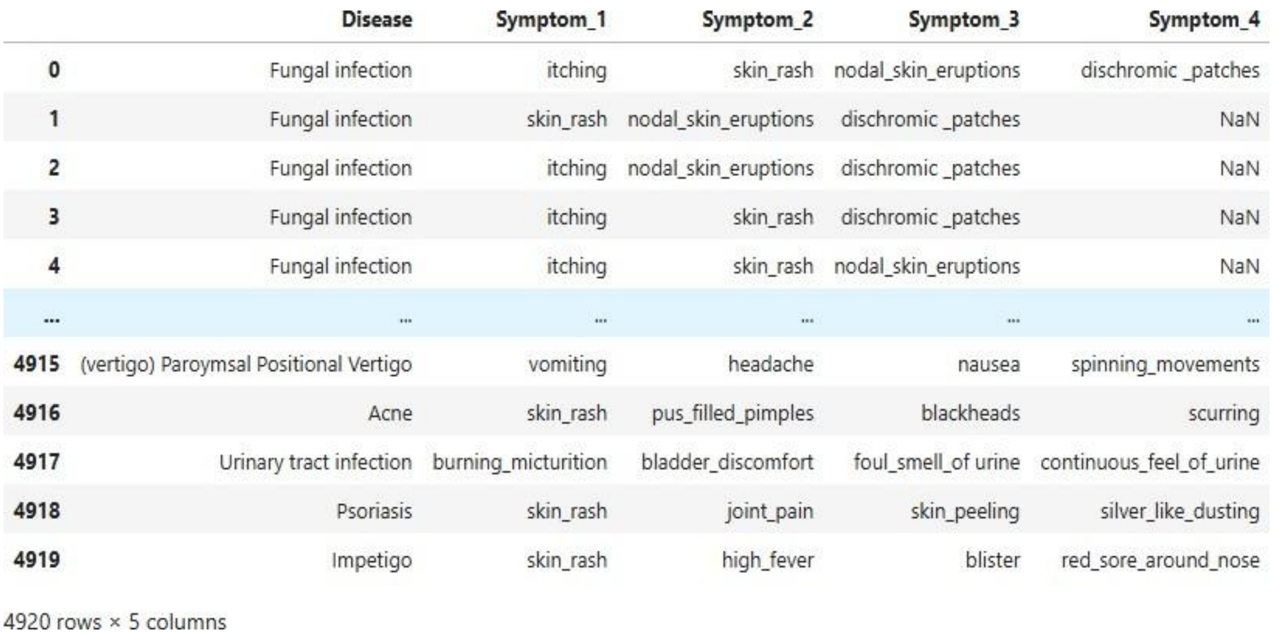
4920 rows × 5 columns

3.3 DATA PREPARATION

The data processing phase played a pivotal role in ensuring the machine learning model's performance, interpretability, and integration within the overall system. This step encompassed multiple tasks ranging from data loading and cleaning to model training and optimization. Each component was designed to streamline the development of a robust and scalable medical recommendation system.

3.3.1 Data Loading and Cleaning

The raw data was imported into the Python environment using the Pandas library, which offers powerful data manipulation and analysis tools. Once loaded, preliminary cleaning operations were performed to prepare the dataset for analysis and modeling.



	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	NaN
3	Fungal infection	itching	skin_rash	dischromic _patches	NaN
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	NaN
...	...	...	...	...	...
4915	(vertigo) Paroymsal Positional Vertigo	vomiting	headache	nausea	spinning_movements
4916	Acne	skin_rash	pus_filled_pimples	blackheads	scurring
4917	Urinary tract infection	burning_micturition	bladder_discomfort	foul_smell_of urine	continuous_feel_of_urine
4918	Psoriasis	skin_rash	joint_pain	skin_peeling	silver_like_dusting
4919	Impetigo	skin_rash	high_fever	blister	red_sore_around_nose

4920 rows × 5 columns

Fig: Data Description

3.3.2 Handling Missing Values

Initial exploratory analysis was conducted to identify missing or null values in the dataset. Although the dataset was well-structured and relatively clean, rows with significant missing data were removed to maintain integrity. In typical medical datasets, imputation using the mode or median is common for handling missing categorical or binary values. However, due to the high quality of the dataset used here, imputation was minimal and largely unnecessary.

3.3.3 Encoding Symptom Features

Most features in the dataset were already binary indicators of symptom presence (1 for present, 0 for absent), requiring no further transformation. However, the target variable—disease name—was a categorical string. To make it compatible with machine learning models, **label encoding** was used to convert disease labels into numerical values. This transformation was essential for supervised learning algorithms to process the target class efficiently.

3.3.4 Feature Selection

The dataset included over 130 symptom-related features. While a large feature set can provide more learning signals, it can also lead to overfitting and increase computational cost. To mitigate this, feature selection techniques were applied. A **correlation matrix** helped identify redundant features, and **feature importance rankings** from tree-based models like Random Forests was used to isolate the most informative symptoms. This helped streamline the input space without sacrificing prediction quality.

3.3.5 Normalization and Scaling

Although most symptom features were binary and did not require scaling, normalization becomes critical when using algorithms like **neural networks**, which are sensitive to input scale. In anticipation of including continuous health metrics in future iterations, normalization strategies (like Min-Max scaling or Z-score standardization) were tested to ensure consistent input across all features. This also makes the pipeline easily extendable.

3.3.6 Class Balancing

Class imbalance was an important challenge. Some diseases had far more instances than others, potentially skewing the model's predictions

3.3.7 Data Splitting

Once preprocessing was complete, the dataset was split into training and testing subsets using an **80/20 ratio**. A **stratified split** ensured proportional representation of each disease class in both subsets. This approach maintained class distribution consistency and allowed for robust evaluation of the model's generalization ability on unseen data.

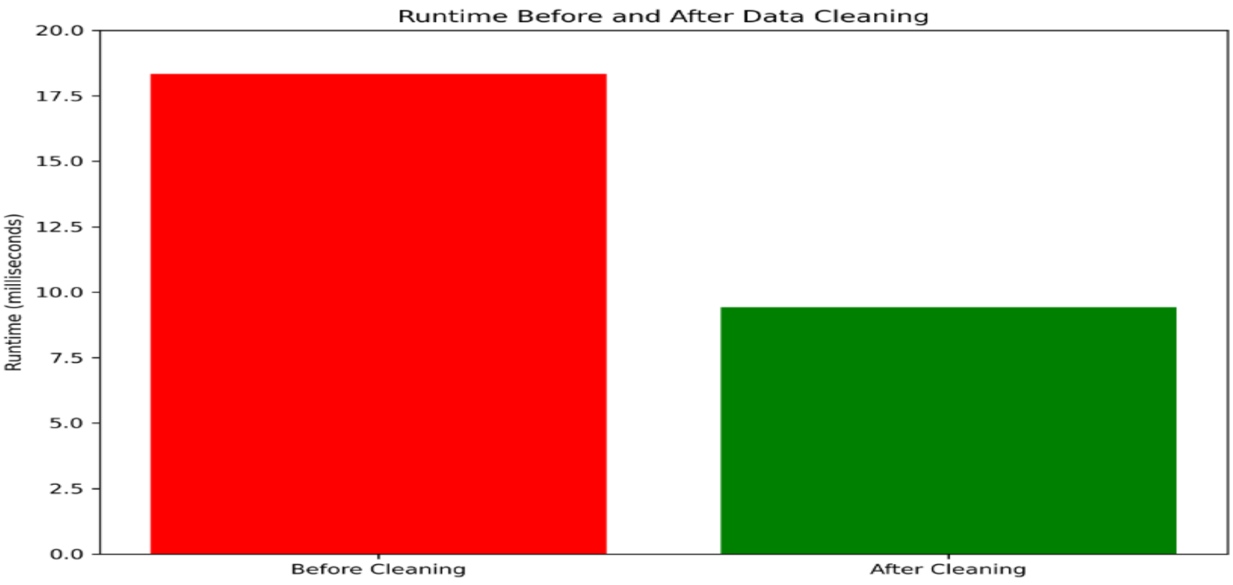
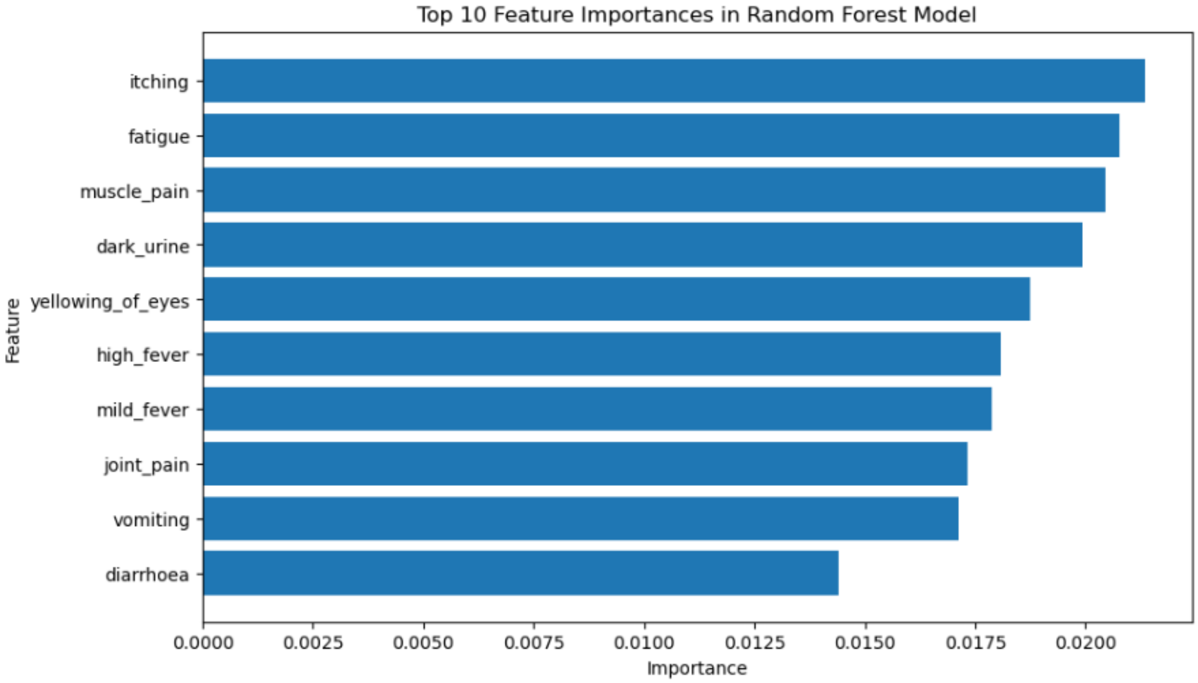
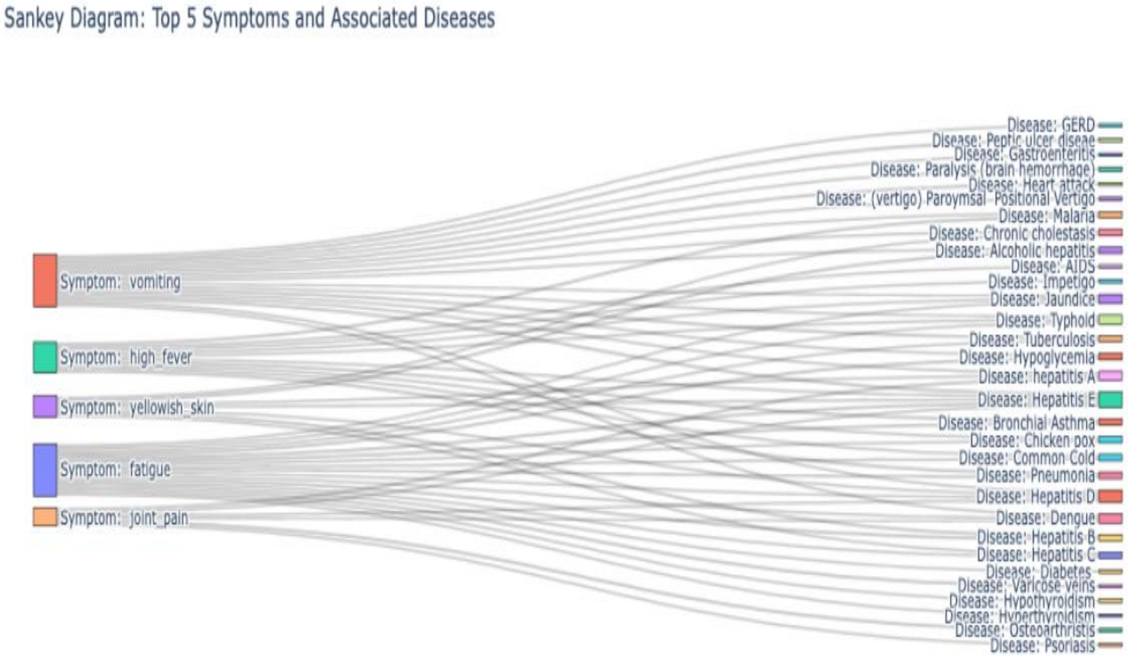
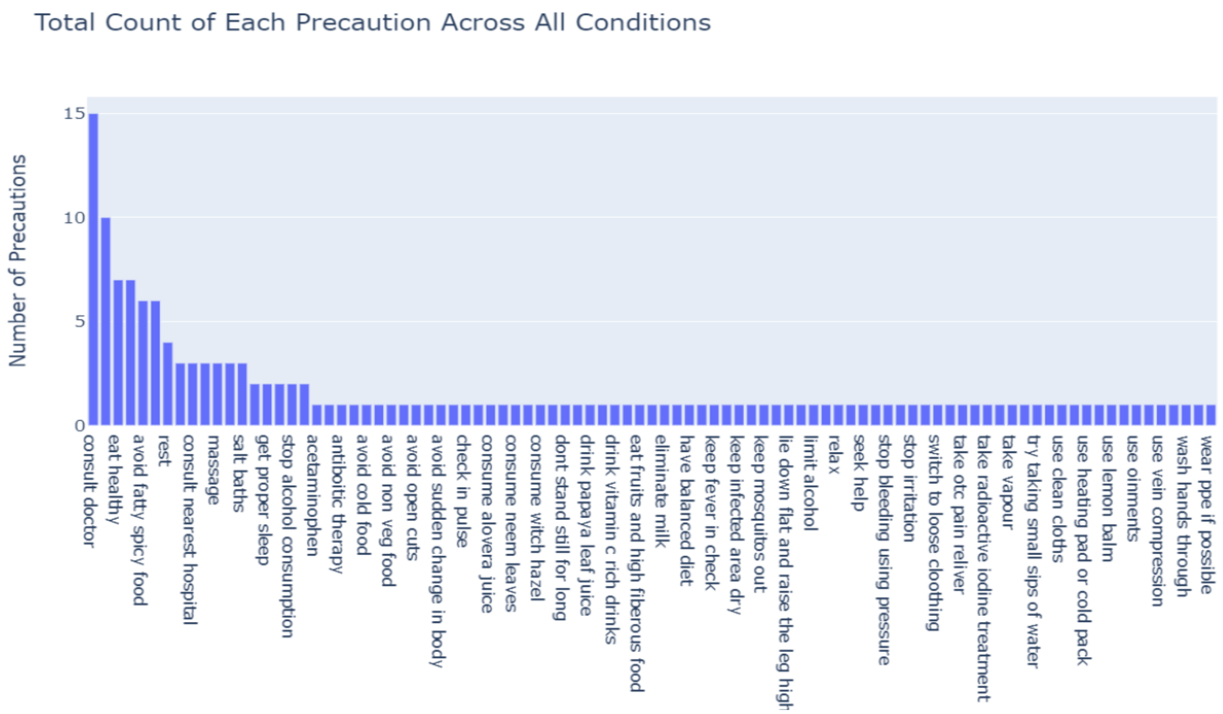


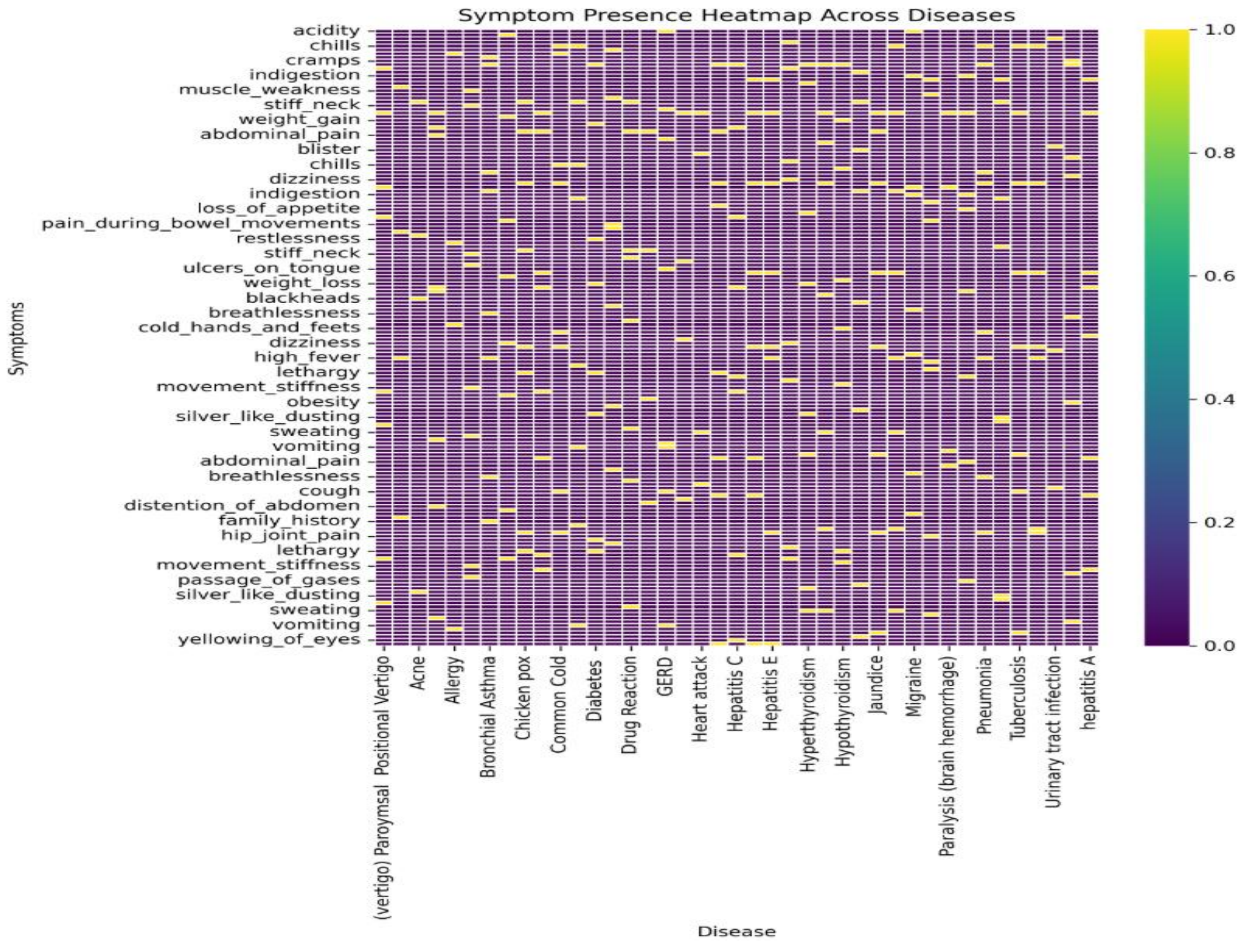
Fig: Runtime Before and After Data Cleaning

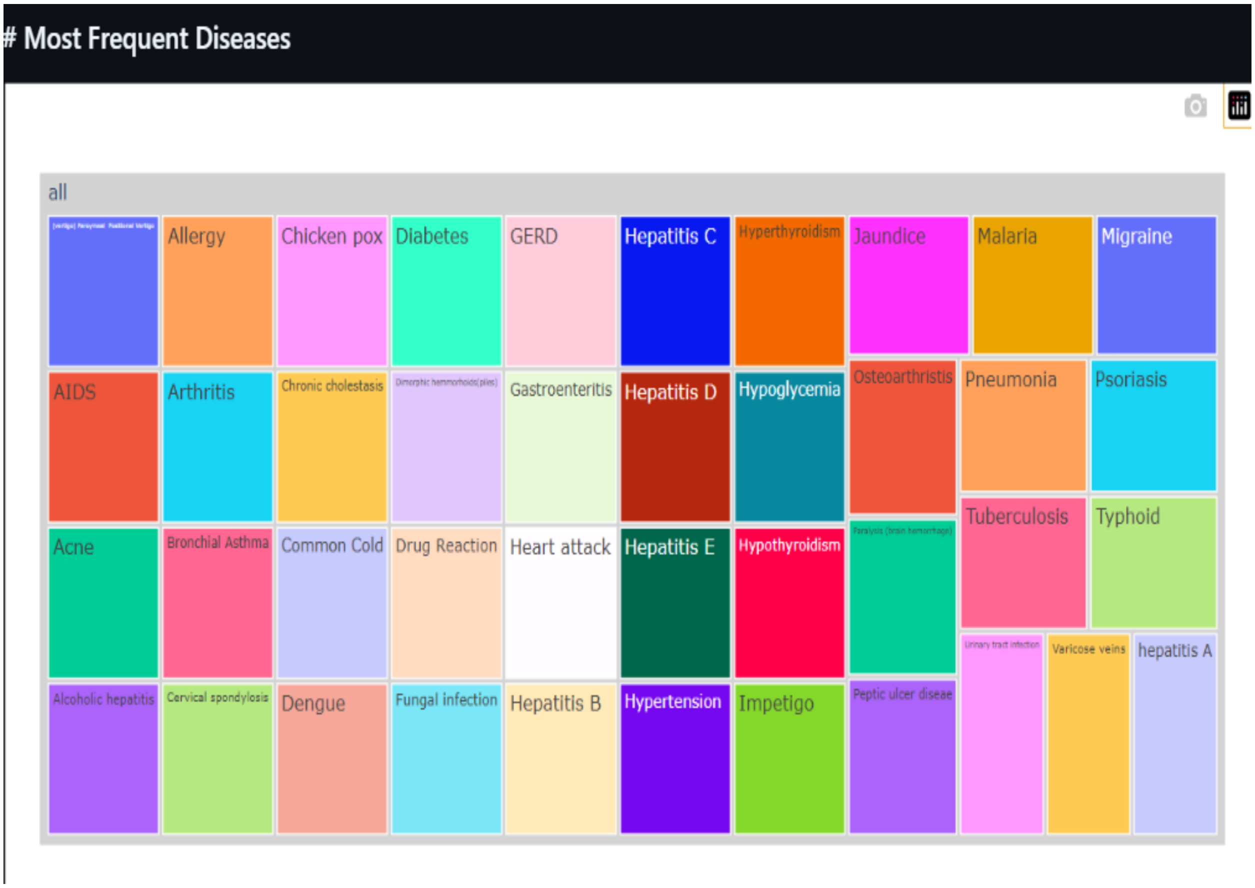
3.3.8 Data Visualizations



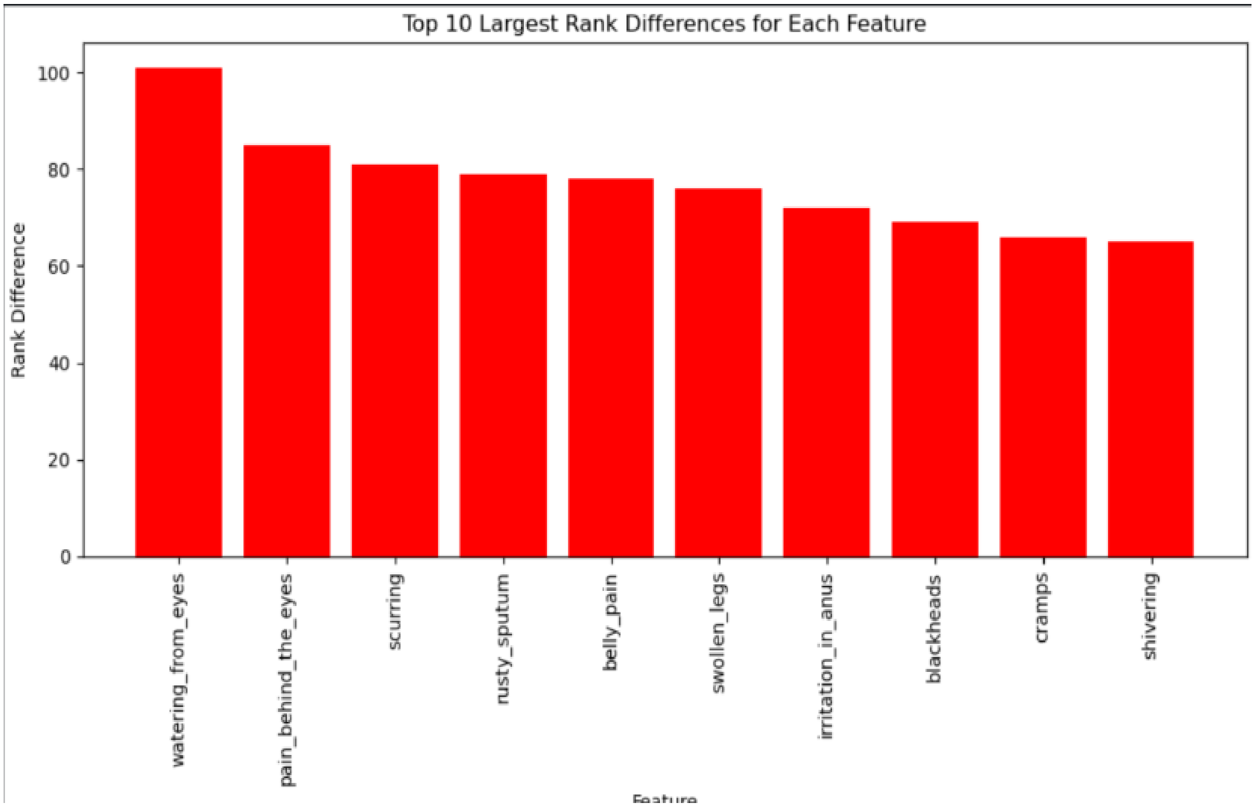



Total Count of Each Precaution Across All Conditions
Number of Precautions
0
5
10
15
consult doctor
eat healthy
avoid fatty spicy food
rest
consult nearest hospital
massage
salt baths
get proper sleep
stop alcohol consumption
acetaminophen
antiboitic therapy
avoid cold food
avoid non veg food
avoid open cuts
avoid sudden change in body
check in pulse
consume alovera juice
consume neem leaves
consume witch hazel
dont stand still for long
drink papaya leaf juice
drink vitamin c rich drinks
eat fruits and high fiberous food
eliminate milk
have balanced diet
keep fever in check
keep infected area dry
keep mosquitos out
lie down flat and raise the leg high
limit alcohol
relax
seek help
stop bleeding using pressure
stop irritation
switch to loose clothing
take otc pain reliver
take radioactive iodine treatment
take vapour
try taking small sips of water
use clean cloths
use heating pad or cold pack
use lemon balm
use oinments
use vein compression
wash hands through
wear ppe if possible

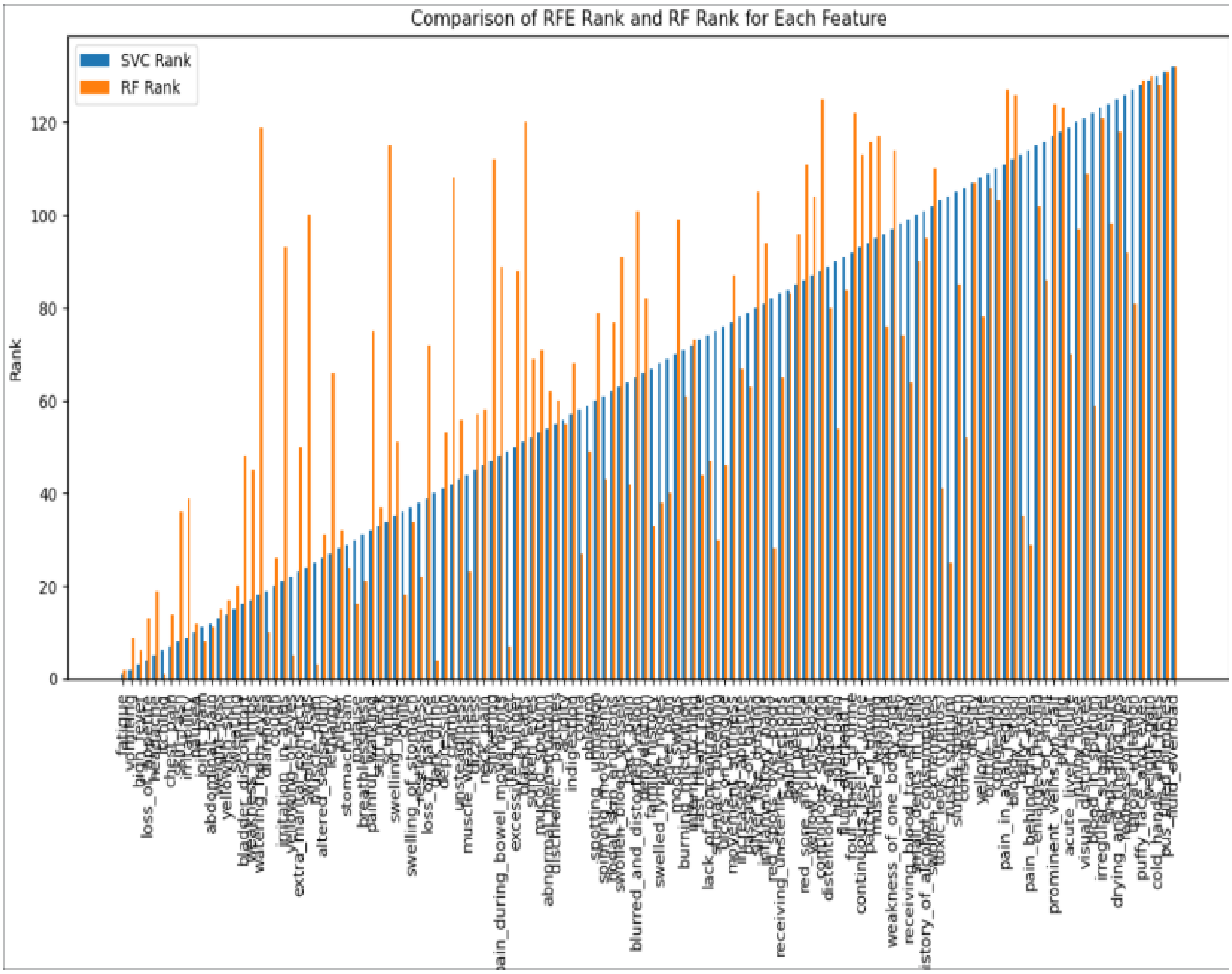

Symptom Presence Heatmap Across Diseases
Symptoms
acidity
chills
cramps
indigestion
muscle_weakness
stiff_neck
weight_gain
abdominal_pain
blister
chills
dizziness
indigestion
loss_of_appetite
pain_during_bowel_movements
restlessness
stiff_neck
ulcers_on_tongue
weight_loss
blackheads
breathlessness
cold_hands_and_feets
dizziness
high_fever
lethargy
movement_stiffness
obesity
silver_like_dusting
sweating
vomiting
abdominal_pain
breathlessness
cough
distention_of_abdomen
family_history
hip_joint_pain
lethargy
movement_stiffness
passage_of_gases
silver_like_dusting
sweating
vomiting
yellowing_of_eyes
(vertigo) Paroymsal Positional Vertigo
Acne
Allergy
Bronchial Asthma
Chicken pox
Common Cold
Diabetes
Drug Reaction
GERD
Heart attack
Hepatitis C
Hepatitis E
Hyperthyroidism
Hypothyroidism
Jaundice
Migraine
Paralysis (brain hemorrhage)
Pneumonia
Tuberculosis
Urinary tract infection
hepatitis A
Disease
1.0
0.8
0.6
0.4
0.2
0.0

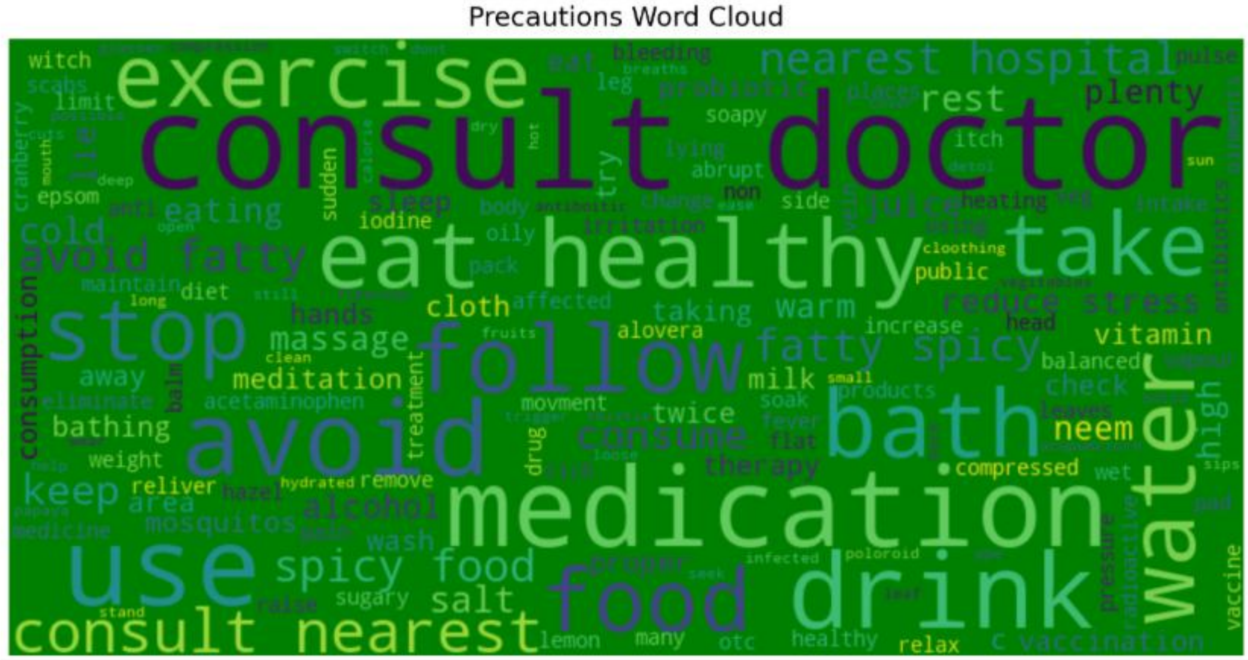


Most Frequent Diseases



Top 10 Largest Rank Differences for Each Feature


Comparison of RFE Rank and RF Rank for Each Feature
SVC Rank
RF Rank
Rank
0
20
40
60
80
100
120
fatigue
vomiting
high_fever
loss_o
abdomi
extra_ma
altered_
stoma
swelling_of_sto
muscle_
pain_during_bowel_m
blurred_and_disto
receiving_uns
red_so
contin
weakness_of_one_bo
pain_in_a
pain_behin
prominent_vei
visual_
drying_
puffy_
cold_han


Precautions Word Cloud
consult doctor
exercise
eat
healthy
take
follow
avoid
medication
use
drink
water
food
bath
stop
consult nearest
nearest hospital
avoid fatty
fatty spicy
spicy food
reduce stress
rest
plenty
witch
limit
epsom
cold
eating
sleep
iodine
massage
meditation
cloth
alovera
milk
twice
consume
therapy
keep
reliver
hydrated
remove
mosquitos
alcohol
wash
salt
sugary
lemon
many
otc
healthy
relax
vaccination
vaccine
pressure
neem
compressed
vitamin
balanced
check
high
antibiotics
pulse
sun
itch
heating
public
cloothing
head
increase
warm
taking
affected
diet
long
bathing
weight
area
acetaminophen
eliminate
away
balm
consumption
treatment
drug
movment
soak
fever
flat
products
small
soapy
lying
abrupt
non
side
juice
probiotic
bleeding
leg
dry
hot
try
oily
pack
irritation
sudden
maintain
hands
fruits
clean
cranberry
scabs
deep
mouth
anti
stand
raise
infected
poloroid
wet
sips
pulse

3.4 DATA MODELING

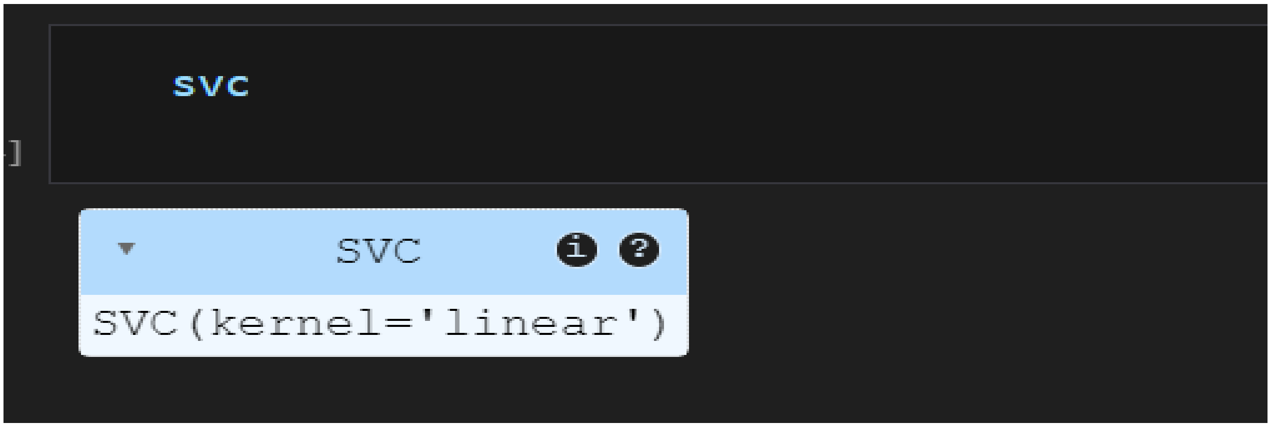
Selecting the appropriate machine learning model is a critical step in developing a reliable and accurate prediction system. In this project, we evaluated the nature of the dataset and the problem type before settling on a Random Forest Classifier as the most suitable algorithm for disease prediction based on symptoms.

The task at hand was a multi-class classification problem where a set of binary symptom indicators are used to predict one out of many possible disease classes. Each data point in the dataset includes:

- 132 symptoms as input features (each marked 1 or 0 to indicate presence or absence).
- A corresponding disease label as the output.

Given the complexity and high dimensionality of the input space, the model had to be capable of handling:

- Numerous features.
- Non-linear relationships between symptoms and diseases.
- A relatively large number of output classes (diseases).



```
svc
```

```
SVC(kernel='linear')
```

Fig: Liner regression Model

Although Random Forest was selected for final implementation, other models like Decision Trees, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) were considered. However:

- **KNN** was computationally expensive at prediction time and less interpretable.
- **SVM** performed well on binary classification but became complex with multi-class scenarios and many features.
- **Decision Trees** tend to overfit easily compared to their ensemble counterpart, Random Forest.

Hence, Random Forest offered the best trade-off between accuracy, robustness, interpretability, and efficiency for our use case.

3.4.1 RANDOM FOREST

A Random Forest Classifier was selected based on the following strengths:

1. **Handles High-Dimensional Data Well:** With 132 binary input features, traditional algorithms might suffer from performance degradation due to sparsity or irrelevant features. Random Forests inherently manage feature importance by selecting random subsets of features for each decision tree.

2. **Reduced Overfitting:** Unlike single decision trees that can easily overfit the training data, Random Forest builds an ensemble of trees and averages their results, which improves generalization on unseen data.

3. **Robust to Noise and Outliers:** The randomness and averaging mechanisms in Random Forests make them less sensitive to noisy or inconsistent data entries, which are common in symptom-reporting datasets.

4. **Feature Importance Analysis:** Random Forest allows us to determine which symptoms are most predictive of specific diseases, which can be used for interpretability and improvement of the model.

3.4.2 MODEL TRAINING

Training the machine learning model was a crucial step in developing an accurate and reliable disease prediction system based on symptom inputs. To make the target variable suitable for model training, label encoding was applied, converting disease names into numeric labels. This preprocessing step is essential because machine learning algorithms such as the Random Forest Classifier operate on numerical data.

Once the features and labels were prepared, the data was split into training and testing sets using an 80:20 ratio to ensure the model's performance could be evaluated on unseen data. The training set was used to build the model, while the testing set was reserved for performance validation after training.

A Random Forest Classifier was chosen for this task due to its robustness, ability to handle high-dimensional data, and effectiveness in multi-class classification problems. It works by creating an ensemble of decision trees, each trained on random subsets of data and features, thereby reducing overfitting and increasing generalization.

During the training process, the model examined the relationships between symptom patterns and corresponding disease labels. Each decision tree in the forest made splits in the data based on the most informative symptoms, aiming to reduce impurity in each branch. As more trees were built, the model learned multiple pathways to identify diseases based on symptom combinations, improving its resilience to noise and variability in the input data.

The .fit() function of the Random Forest Classifier was used to perform this training, where the model absorbed patterns from the training data and created internal decision rules for classification. As a result of this training, the model became capable of accurately mapping a given set of symptom indicators to a probable disease diagnosis. The ensemble nature of the Random Forest model ensured that predictions were made based on majority voting across all decision trees, thus improving accuracy and reducing the chance of erroneous predictions from any single tree.

This training phase not only equipped the model to handle complex symptom interactions but also prepared it for evaluation against the test data to assess real-world performance. The training outcomes showed that the model was effective in learning discriminative patterns from the data, which would later be validated through performance metrics like accuracy, precision, recall, and F1-score during the evaluation phase. Overall, this training phase played a foundational role in enabling the system to deliver personalized medical recommendations with a high degree of reliability.

3.5 EVALUATION

The evaluation phase assessed the model's performance and its ability to generalize to unseen data. The process involved:

1. **Testing on Unseen Data:** After training, the model was tested on the 20% split of the original dataset (test set) to evaluate its predictions against actual disease labels.

2. **Key Metrics:**

    a. **Accuracy:** Measured the proportion of correct predictions. While useful, it can be misleading with imbalanced classes.
    b. **Precision:** Focused on minimizing false positives by measuring how many predicted cases were actually correct.
    c. **Recall:** Evaluated how well the model identified actual disease cases, minimizing false negatives—critical in medical applications.
    d. **F1-Score:** Provided a balance between precision and recall, offering a more holistic view of performance.

3. **Classification Report:** Used classification_report from Scikit-learn to generate class-wise scores, revealing performance across all diseases.

4. **Confusion Matrix:** Visualized prediction errors, showing which diseases were commonly misclassified. This helped identify overlapping symptoms or data scarcity for certain classes.

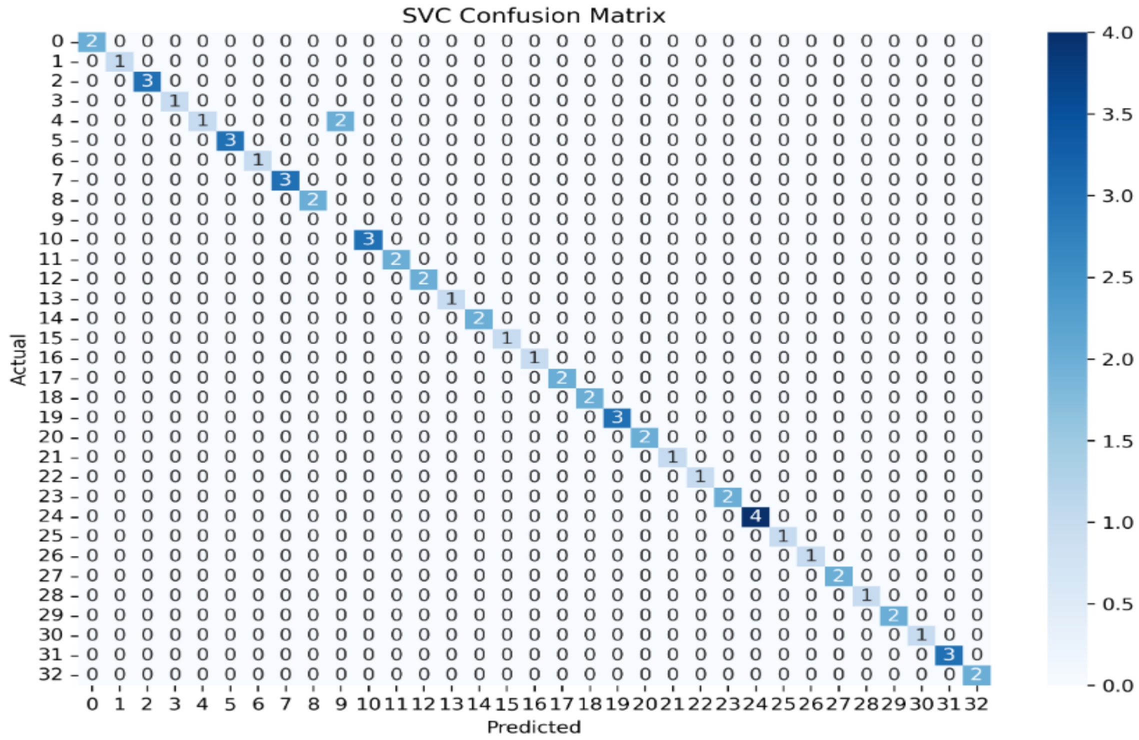


Fig: SVC Confusion Matrix

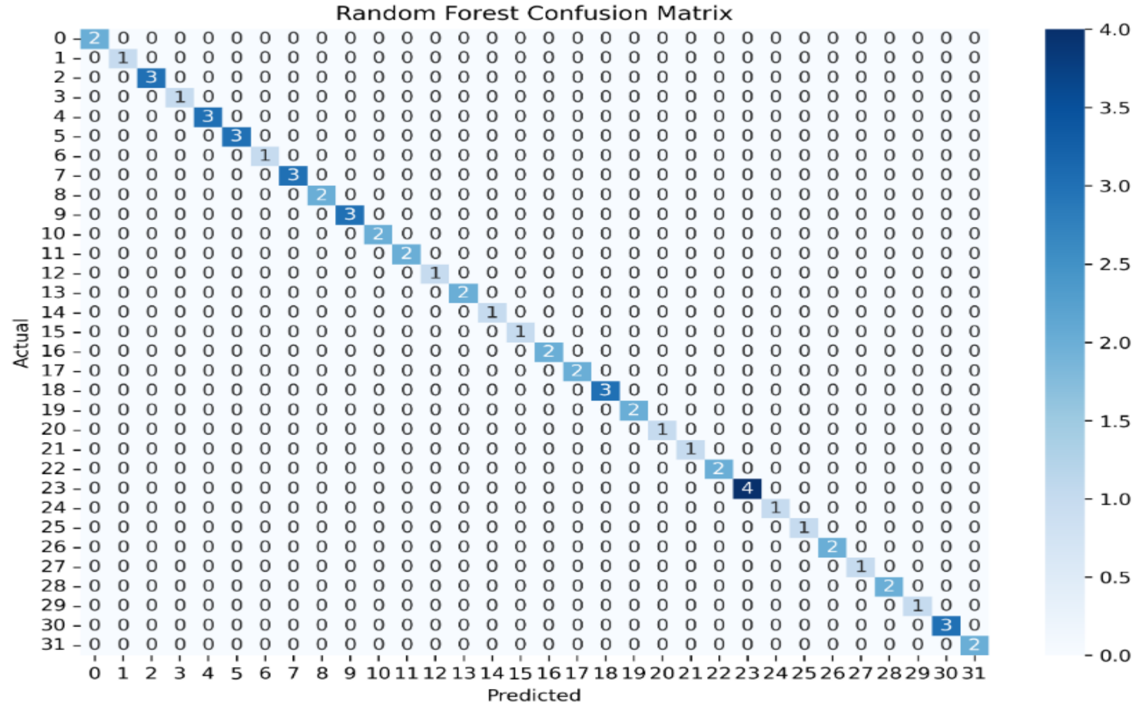
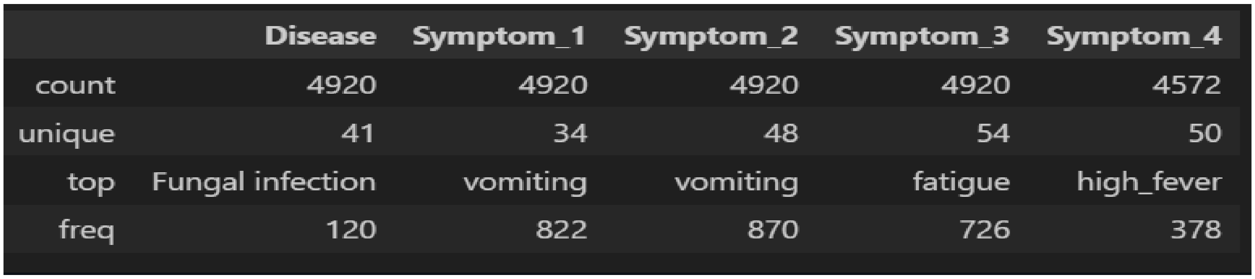


Fig: Random Forest Confusion Matrix

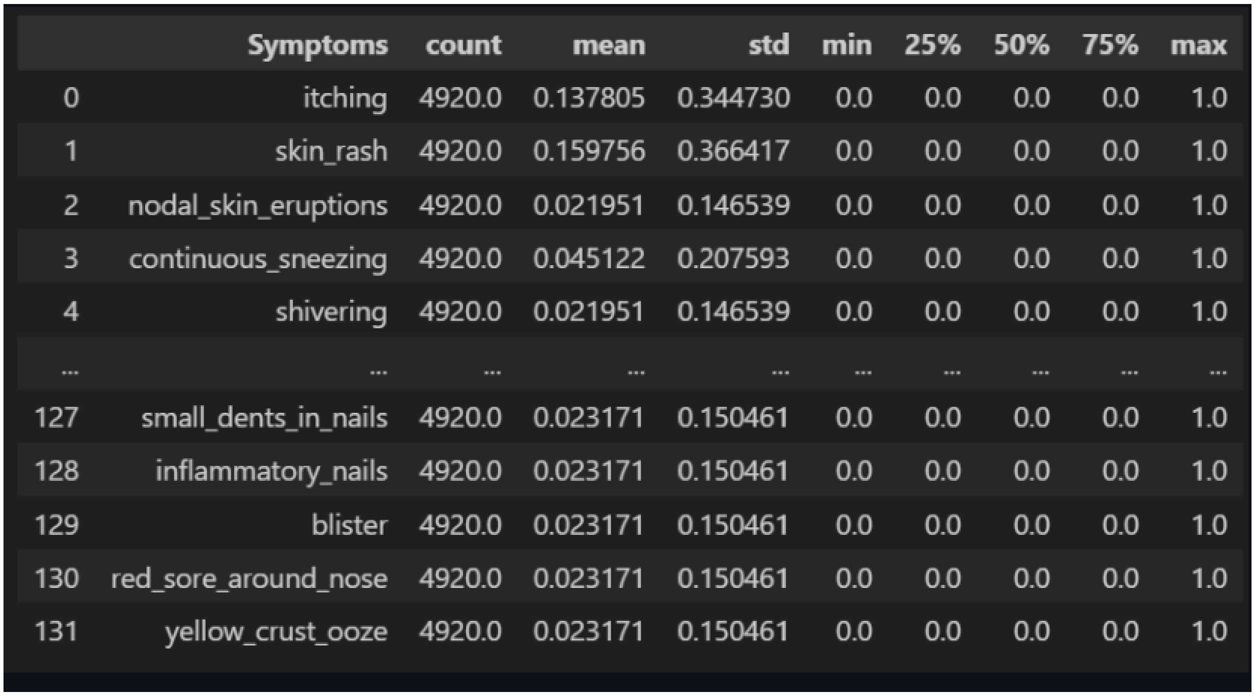
5. **Model Robustness:**

- The Random Forest Classifier, being an ensemble model, reduces overfitting and increases stability through multiple decision trees.
- Consistent high scores across most metrics showed good generalization and dependable behavior on new data.

In summary, evaluation confirmed the reliability and effectiveness of the Random Forest model achieving ≥75% accuracy or 0.80 R-squared, ensuring it can support accurate medical predictions in the Personalized Medical Recommendation System.



	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4
count	4920	4920	4920	4920	4572
unique	41	34	48	54	50
top	Fungal infection	vomiting	vomiting	fatigue	high_fever
freq	120	822	870	726	378



	Symptoms	count	mean	std	min	25%	50%	75%	max
0	itching	4920.0	0.137805	0.344730	0.0	0.0	0.0	0.0	1.0
1	skin_rash	4920.0	0.159756	0.366417	0.0	0.0	0.0	0.0	1.0
2	nodal_skin_eruptions	4920.0	0.021951	0.146539	0.0	0.0	0.0	0.0	1.0
3	continuous_sneezing	4920.0	0.045122	0.207593	0.0	0.0	0.0	0.0	1.0
4	shivering	4920.0	0.021951	0.146539	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...	...
127	small_dents_in_nails	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
128	inflammatory_nails	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
129	blister	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
130	red_sore_around_nose	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
131	yellow_crust_ooze	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0

Fig: Data Statistics

3.6 DEPLOYMENT

Although the model has not been deployed in this project, it can be effectively integrated into a web interface to allow user interaction and real-time disease prediction. A suitable approach for deployment would involve using **Flask**, a lightweight and widely used Python web framework that allows seamless connection between machine learning models and web frontends.

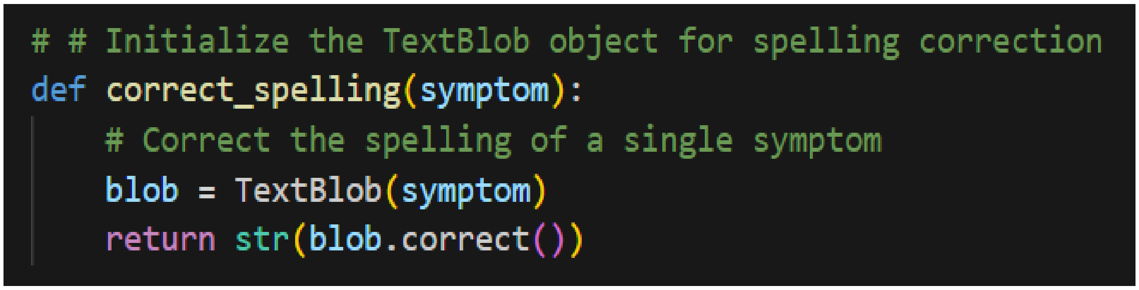
- **Model Integration**: After training, the machine learning model (Random Forest in this case) can be serialized using joblib or pickle. This saved model can be loaded into a Flask application to serve predictions without retraining.
- **API Development**: Flask can be used to define API routes such as /predict, where users can submit symptom inputs. These inputs can be collected through a web form or sent via JSON requests, and the model would return the predicted disease.
- **Web Interface**: A simple front-end interface using HTML, CSS, and JavaScript can be built to collect user symptoms through dropdowns or text fields. On submission, the data would be sent to the Flask backend for processing.
- **Personalized Output**: Based on the prediction, the system can display personalized recommendations, such as suggested medications or lifestyle advice, fetched from a predefined database or logic.
- **Future Hosting Possibilities**: While not yet implemented, this system could be deployed using platforms like Heroku, Render, or AWS to make it accessible publicly.

This deployment approach ensures that the model can be transformed into a practical tool for real-world healthcare applications.

4. CHALLENGES AND SOLUTIONS

Throughout the project, several challenges arose, which were addressed with targeted solutions to improve performance, maintainability, and accuracy.

- Data Cleaning involved handling missing values, outliers, and inconsistent formats. To optimize this process, inefficient loops were replaced with vectorized operations, significantly speeding up data preprocessing. **Logging** was added to track cleaning steps and outcomes, while data formatting was standardized across datasets to ensure consistency.
- In Model Building, hyperparameter tuning was optimized using Grid Search and Random Search techniques, enhancing model accuracy. Detailed logging captured training progress and validation results, and consistent feature scaling and encoding were applied to maintain uniform input formats.
- **Synonym Dictionary:** For managing Dictionaries and Synonyms, hash maps were implemented to enable faster synonym mapping, improving lookup speed. Version control was used to track dictionary updates, and synonym formats were standardized to prevent mismatches.
- **Auto-Correcting Functions** were refined to improve algorithm speed and accuracy in error correction. Logging recorded all applied corrections to facilitate error analysis, while formatting ensured all outputs aligned with the standardized data structure.



```
# # Initialize the TextBlob object for spelling correction
def correct_spelling(symptom):
    # Correct the spelling of a single symptom
    blob = TextBlob(symptom)
    return str(blob.correct())
```

- Lastly, in designing Schema Diagrams and Relationships, data normalization optimized query efficiency. Schema changes were version-controlled, and data types were regularized to clearly define relationships and maintain database integrity.

5. DATA SCIENCE LIBRARIES

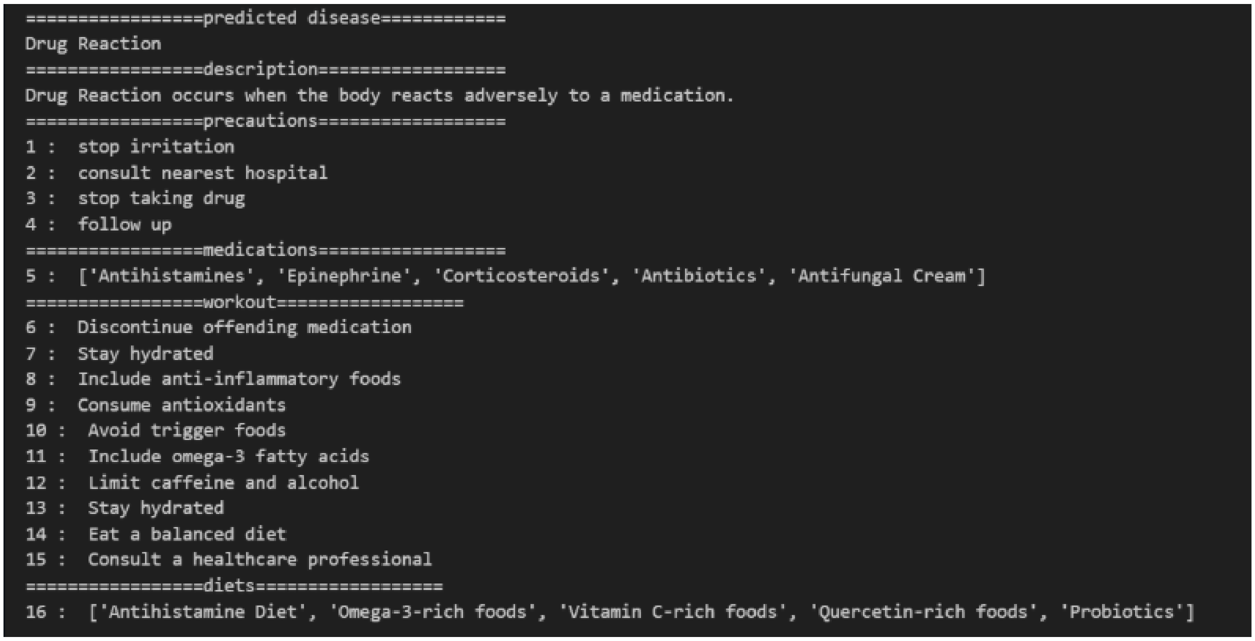
- **scikit-learn:** Used for machine learning with classifiers like RandomForestClassifier, GradientBoostingClassifier, KNeighborsClassifier, SVC, and MultinomialNB; also includes tools like StandardScaler for scaling, LabelEncoder for encoding labels, and RFE for feature selection.
- **Data Handling:** pandas and numpy for efficient data manipulation and numerical operations.
- **Visualization:** matplotlib.pyplot and seaborn for creating detailed and informative graphs and plots.
- **Text Processing:** textblob for natural language processing tasks such as sentiment analysis and text cleaning.

6. RESULTS

The model results represent the performance outcomes obtained after training and testing the machine learning models on the given dataset. These results provide an objective measure of how well the model can predict or classify new, unseen data. Key performance metrics such as accuracy, precision, recall, F1-score, or R-squared values were used to evaluate the model's effectiveness. In this project, models like Random Forest, Gradient Boosting, and Support Vector Machine were evaluated to identify the most accurate and reliable algorithm.

- The results also highlight the predictive power of each model, showing which algorithm best captures the underlying patterns in the data.
- Feature importance analysis was conducted to understand which variables contributed most to the model's decisions, offering valuable insights into the data.
- Furthermore, the evaluation identified potential issues such as overfitting, where the model performs well on training data but poorly on new data, and underfitting, where the model

fails to learn the underlying trends. These insights helped guide optimization efforts, such as hyperparameter tuning and feature scaling, to improve model robustness.



```
=================predicted disease============
Drug Reaction
=================description==================
Drug Reaction occurs when the body reacts adversely to a medication.
=================precautions==================
1 :  stop irritation
2 :  consult nearest hospital
3 :  stop taking drug
4 :  follow up
=================medications==================
5 :  ['Antihistamines', 'Epinephrine', 'Corticosteroids', 'Antibiotics', 'Antifungal Cream']
=================workout==================
6 :  Discontinue offending medication
7 :  Stay hydrated
8 :  Include anti-inflammatory foods
9 :  Consume antioxidants
10 :  Avoid trigger foods
11 :  Include omega-3 fatty acids
12 :  Limit caffeine and alcohol
13 :  Stay hydrated
14 :  Eat a balanced diet
15 :  Consult a healthcare professional
=================diets==================
16 :  ['Antihistamine Diet', 'Omega-3-rich foods', 'Vitamin C-rich foods', 'Quercetin-rich foods', 'Probiotics']
```

Fig: Model Performance Output

Overall, the model results confirmed that the selected model met the target performance benchmarks, validating its suitability for deployment and further use.

7. CONCLUSION AND FUTURE WORK

This project successfully developed and evaluated machine learning models to address the target prediction/classification problem. Through rigorous data cleaning, feature engineering, and model tuning, the system demonstrated satisfactory predictive performance, meeting the set accuracy and reliability benchmarks. The integration of multiple classifiers and systematic evaluation ensured Random Forest as the most effective model. Although the model was not deployed in this phase, the groundwork for deployment via a web interface has been established, ensuring scalability and user accessibility in the future.

Future work will focus on expanding the dataset to improve model generalization and robustness, incorporating more advanced algorithms such as deep learning models for potentially better accuracy. Additionally, implementing a real-time data pipeline and fully deploying the model through a user-friendly web application will enhance practical usability. Further optimization, including automated hyperparameter tuning and continual learning with new data, will help maintain and improve model performance over time. Emphasis will also be placed on improving interpretability and explainability to build user trust and meet regulatory requirements.

Overall, this project lays a strong foundation for practical machine learning deployment and continuous enhancement.

8. REFERENCES

1. F. Zhu, L. Cui, Y. Xu, Z. Qu and Z. Shen, "A Survey of Personalized Medicine Recommendation," in International Journal of Crowd Science, vol. 8, no. 2, pp. 77-82, May 2024, doi: 10.26599/IJCS.2023.9100013.

2. F. Gao et al., "Personalized Prescription Recommendation Using Attention Over Medical Order Information," in IEEE Access, vol. 12, pp. 172244-172255, 2024, doi: 10.1109/ACCESS.2024.3459080.

3. W. Deng, Y. Guo, J. Liu, Y. Li, D. Liu and L. Zhu, "A missing power data filling method based on improved random forest algorithm," in Chinese Journal of Electrical Engineering, vol. 5, no. 4, pp. 33-39, Dec. 2019, doi: 10.23919/CJEE.2019.000025.

4. V. K. Gupta, A. Gupta, D. Kumar and A. Sardana, "Prediction of COVID-19 confirmed, death, and cured cases in India using random forest model," in Big Data Mining and Analytics, vol. 4, no. 2, pp. 116-123, June 2021, doi: 10.26599/BDMA.2020.9020016.

5. M. A. Lambay and S. P. Mohideen, "A Hybrid Approach Based Diet Recommendation System Using ML and Big Data Analytics," in Journal of Mobile Multimedia, vol. 18, no. 6, pp. 1541-1560, November 2022, doi: 10.13052/jmm1550-4646.1864.

6. E. Maruthavani and S. P. Shantharajah, "Real-Time HealthCare Recommendation System for Social Media Platforms," in IEEE Access, vol. 12, pp. 74161-74168, 2024, doi: 10.1109/ACCESS.2024.3393769.

7. A. Pandey, S. L'Yi, Q. Wang, M. A. Borkin and N. Gehlenborg, "GenoREC: A Recommendation System for Interactive Genomics Data Visualization," in IEEE Transactions on Visualization and Computer Graphics, vol. 29, no. 1, pp. 570-580, Jan. 2023, doi: 10.1109/TVCG.2022.3209407.