Disease Diagnosis Assistant

Amit Pareek, Mayur N Sastry

August 1, 2025

1 Introduction

The Disease Diagnosis Assistant is a simple tool designed to support doctors in diagnosing diseases and recommending treatment steps. By interacting with doctors to gather symptom information, the assistant helps narrow down potential diagnoses and suggests relevant next steps for treatment. The aim is to aid early diagnosis by leveraging a machine learning model for symptom analysis and reliable resources for treatment guidance.

2 Literature Review

Machine learning (ML) has transformed disease prediction by utilizing various algorithms tailored for analyzing patient data. **Random Forest** (RF) is widely favored due to its ability to handle both categorical and continuous data, improving disease classification accuracy through an ensemble of decision trees. Studies show RF can effectively predict diseases based on symptoms and geographical factors (Gaurav et al., 2023).

Support Vector Machines (SVM) are another prominent technique used in disease classification, demonstrating effectiveness in high-dimensional datasets, while optimizing the margin between classes. Research has indicated SVM's successful application in correlating symptoms with diseases.

Long Short-Term Memory (LSTM) networks further enhance disease prediction by handling time-dependent symptom data, making them ideal for chronic condition forecasting. Integrating ML models in healthcare has considerable benefits, such as early diagnosis and personalized treatment plans, ultimately reducing healthcare costs.

However, challenges like data privacy, the need for diverse training datasets, and interpretability of results must be addressed. Future research should focus on enhancing model transparency, integrating multifaceted data, and developing explainable AI solutions to ensure clinician and patient trust in ML-driven predictions. Overall, machine learning models hold great promise in revolutionizing healthcare delivery by enabling accurate, timely disease predictions.

In this project, we focus on building a practical disease diagnosis assistant by leveraging various data-driven approaches like examining symptom co-occurrence and correlation. By analyzing the patterns between symptoms and their relationship with different diseases, our goal is to create an intuitive tool that can provide physicians with reliable suggestions based on patient-reported symptoms. Unlike typical ML-based solutions that directly predict diseases, our approach incorporates statistical insights, such as identifying frequently co-occurring symptoms, calculating correlation matrices, and exploring symptom relationships. This combined methodology aims to enhance diagnostic accuracy and offer an evidence-based recommendation system, ultimately aiding healthcare providers in making more informed clinical decisions.

3 Exploratory Data Analysis: Description of the Dataset

The dataset under analysis consists of **8,835 rows** and **490 columns**, structured to represent the relationship between diseases and their associated

symptoms. This section provides an in-depth description of the dataset's structure, content, and key statistical properties.

3.1 Dataset Structure

The dataset includes:

- **Label Column** (`label_dis`):
  - Represents the name of the disease corresponding to each row.
  - Contains **261 unique diseases**, indicating that the dataset covers a wide range of medical conditions.
  - The disease distribution is highly imbalanced. For example:
    - The most frequent disease is *Myocardial Infarction (Heart Attack)*, which appears in **2,047 rows** ($\sim 23.2\%$ of the dataset).
    - Many diseases are represented by far fewer rows, suggesting variability in data coverage for different conditions.
- **Symptom Columns (489 columns)**:
  - Each symptom is represented by a binary value:
    - **1** indicates the presence of the symptom.
    - **0** indicates its absence.
  - These columns collectively capture various possible symptoms, which form the features used to describe each disease instance.

3.2 Key Statistical Insights

- **Binary Symptom Representation**:
  - All symptom columns are binary (**0** or **1**), reflecting whether the symptom is associated with the given disease instance.

  - The dataset is sparse, as most symptoms have a mean occurrence frequency close to zero. For instance:
    * The symptom *abdominal cramp* appears in only **0.09%** of rows.
    * The symptom *yellowish skin crust* appears in just **0.02%** of rows.

- **Disease Representation**:
  - The dataset's imbalance is notable; some diseases dominate the dataset (e.g., *Myocardial Infarction*), while others are represented by only a few instances.
  - This imbalance may introduce bias in models trained on this data, making it essential to account for it during analysis and model development.

3.3 Sample Data

A sample of the dataset provides insight into its structure: This snippet

label_dis	abdominal cramp	abdominal distention	abnormal behavior	abscess	...
Abscess	0	0	0	0	...
Abscess	0	0	0	0	...
...	...	...	...	...	...

Table 1: Snippet of the Dataset

demonstrates how each row associates a disease (from `label_dis`) with a combination of symptoms, most of which are absent (**0**) in any given row.

3.4 Further Data Analysis

In this section, we analyze the most frequent diseases in our dataset, present the correlation matrix for a subset of symptoms, and examine the most correlated symptoms for a specific disease (influenza).

Top 10 Most Frequent Diseases

The following table lists the top 10 most frequent diseases along with their occurrence counts in the dataset:

Disease	Frequency
Myocardial Infarction (Heart Attack)	2047
Polycystic Ovary Syndrome (PCOS)	511
Anthrax	511
Porphyria	255
Rabies	255
Lead Poisoning	255
Hyperthyroidism	255
Lupus Erythematosus	255
Celiac Disease	127
Goitre	127

Table 2: Top 10 Most Frequent Diseases in the Dataset

Correlation Matrix for a subset of Symptoms

To understand the relationships between few symptoms, we calculated the correlation matrix for the following symptoms:
headache, coughing, dizziness, fatigue, blurry vision

The correlation matrix is shown in Figure 1. The values represent the correlation coefficients, where a higher value indicates a stronger correlation between the symptoms.

Most Correlated Symptoms with Influenza

We also explored the symptoms most correlated with influenza. The plot in Figure 2 visualizes these correlations, highlighting the symptoms with the highest correlation scores.

From the analysis, symptoms like muscle joint pain, cough, and runny nose show the strongest correlation with influenza, indicating their common co-occurrence in patients diagnosed with this disease.

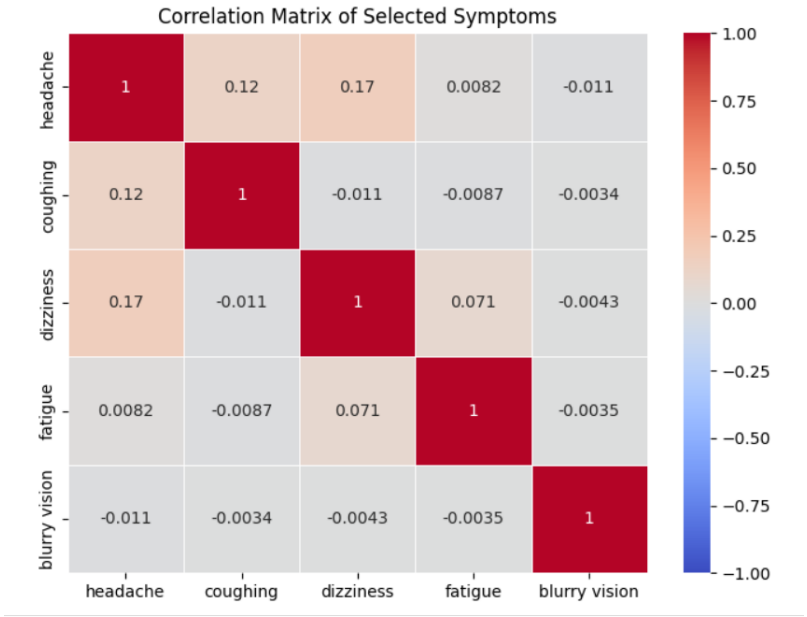

Figure 1: Correlation between a subset of symptoms

4 Process Workflow

1. Symptom Collection and Model Training:

The assistant is trained on a dataset containing symptoms and related diseases using a Random Forest classifier to predict disease probabilities based on symptom patterns.

The Random Forest classifier is an ensemble machine learning model that builds multiple decision trees during training. Each tree makes predictions based on random subsets of the data and features, and the final prediction is determined by aggregating the outputs of all trees. This ensemble approach enhances the model's accuracy and reduces the risk of overfitting, making Random Forest a popular choice for classification tasks.

The predict probabilities method of Random Forest provides the probability

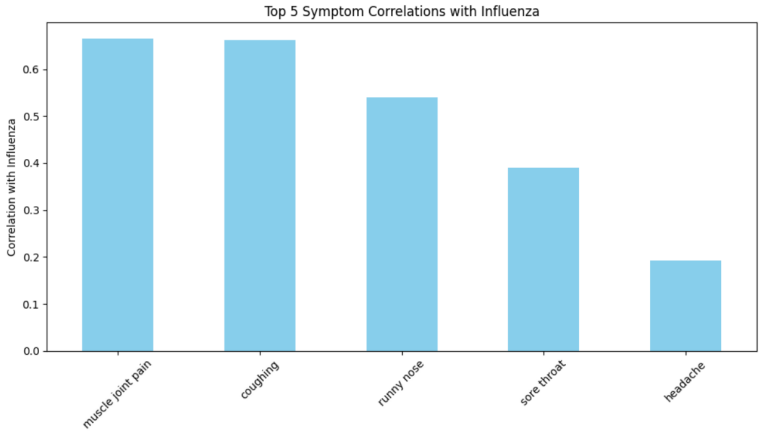

Figure 2: Most Correlated Symptoms with Influenza

of each class for a given input. Instead of a direct class prediction, predict probabilities method outputs a probability distribution across all possible classes, showing how confident the model is for each class label. For instance, if predicting between two diseases, predict probalities method might output values like [0.7, 0.3], indicating a 70 percent likelihood of the first disease and 30 percent for the second. This method is valuable in cases where uncertainty or degree of confidence in the predictions is important, such as in medical diagnosis.

2. Co-occurrence and Correlation:

The diagnostic process begins with the doctor entering a main symptom. The assistant then suggests related symptoms that may be associated with this main symptom, which the doctor can either confirm or rule out. There are two approaches to identify related symptoms: using co-occurrence or using correlation.

Co-occurrence:

The co-occurrence between symptoms measures how often two symptoms

appear together or are absent together across patient records. Let A_{ij} be an indicator variable defined as:

$$A_{ij} = \begin{cases} 1, & \text{if patient } i \text{ has symptom } j \\ 0, & \text{otherwise} \end{cases}$$

The co-occurrence between symptom j and symptom k is calculated as:

$$\text{Cooc}(j,k) = \frac{|\{i : A_{ij} = A_{ik}\}|}{N}$$

Here, N is the total number of patient records in the dataset (in this case, $N = 8835$). This formula measures the proportion of patients who either have both symptoms or neither, capturing the relationship between j and k.

Correlation:
An alternative approach to measuring the relationship between two symptoms is to use the Pearson correlation coefficient. Given a dataset where each column represents a symptom and each entry is either 1 (if the symptom is present) or 0 (if the symptom is absent), the correlation between symptom j and symptom k is calculated as:

$$\text{Corr}(j,k) = \frac{\sum_{i=1}^{N}(A_{ij} - \bar{A}_j)(A_{ik} - \bar{A}_k)}{\sqrt{\sum_{i=1}^{N}(A_{ij} - \bar{A}_j)^2}\sqrt{\sum_{i=1}^{N}(A_{ik} - \bar{A}_k)^2}}$$

where:

- A_{ij} and A_{ik} are the values for patient i for symptoms j and k, respectively.
- $\bar{A}_j$ is the mean value of A_{ij} across all patients, i.e., the proportion of patients with symptom j.
- $\bar{A}_k$ is the mean value of A_{ik} across all patients, i.e., the proportion of patients with symptom k.

The Pearson correlation coefficient Corr(j, k) measures the linear relationship between the presence of symptom j and symptom k, with values ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation).

In this context, we experiment with both methods—co-occurrence and correlation—to identify related symptoms.

3. Predicting the Next Symptom:

Let F denote the function measuring co-occurrence or correlation between symptoms, i.e., $F =$ Cooc or $F =$ Corr.

Given an initial symptom j, the next symptom k can be predicted as the one with the highest value of $F(j, k)$.

Now, suppose we have data on a set of symptoms $j_1, \ldots, j_n$ for a patient. We want to predict the next symptom k. Let R_m denote the patient's response for symptom j_m:

$$R_m = \begin{cases} 1, & \text{if the patient has symptom } j_m \\ 0, & \text{otherwise} \end{cases}$$

Define two sets based on the patient's responses:

- S_1: The set of symptoms for which the patient responded positively, i.e., $S_1 = \{j_m : R_m = 1\}$.
- S_0: The set of symptoms for which the patient responded negatively, i.e., $S_0 = \{j_m : R_m = 0\}$.

To predict the next symptom, we define a *symptom score* that assigns positive weights to correlations with symptoms in S_1 and negative weights to correlations with symptoms in S_0. Mathematically, the symptom score for

a candidate symptom k is given by:

$$\text{SymptomScore}(k) = \sum_{j \in S_1} F(j,k) - \sum_{j \in S_0} F(j,k)$$

The next symptom k is chosen as the one with the highest symptom score.

Example User Interaction with the Symptom Checker (using Correlation)

Below is an example interaction between the user and the Symptom Checker program:

```
Welcome to the Symptom Checker!
What is your main symptom? upper abdomen pain
Do you have belching? (yes/no): yes
Do you have poor appetite? (yes/no): no
Do you have unintended weight loss? (yes/no): no
Do you have vomiting? (yes/no): yes
```

The collected symptoms and their statuses are then summarized as follows:

$$\text{Symptoms Status:} \quad \begin{aligned} &\{\text{'upper abdominal pain'}: 1, \\ &\text{'belching'}: 1, \\ &\text{'poor appetite'}: 0, \\ &\text{'unintended weight loss'}: 0, \\ &\text{'vomiting'}: 1\} \end{aligned}$$

Here, the dictionary shows a key-value pair where the key is the symptom, and the value indicates the symptom status: **1** for "yes" (symptom present) and **0** for "no" (symptom absent).

4. Predicting Disease Probabilities Using Symptom Data:

Given a set of n symptoms and a corresponding input dataset where each symptom is represented as a binary feature (0 or 1), the task is to predict the probabilities of various diseases using a trained Random Forest (RF) classifier.

The process involves the following steps:

4.1 **Input Representation:**

- For each symptom, we represent its presence or absence in the patient's response.
- If the symptom is present (affirmative response), it is marked as 1. If the symptom is absent (negative response), it is typically marked as 0.
- In cases where we want to provide stronger negative evidence for absent symptoms, an additional adjustment can be made by encoding absent symptoms as -1. This allows the model to weigh negative evidence differently from neutral (zero) values.

4.2 **Feature Vector Construction:**

- Based on the input symptom data, we construct a feature vector for the patient. This vector has the same length as the total number of symptoms in the dataset.
- Each element in the vector corresponds to a particular symptom and is set to 1, 0, or -1 depending on the patient's response.

4.3 **Prediction Using Random Forest:**

- The constructed feature vector is fed into the trained Random Forest classifier.
- The RF model, consisting of an ensemble of decision trees, processes the input feature vector and outputs the probability distribution across all possible diseases.

- Each decision tree in the forest contributes to the final probability by voting for one of the disease classes, and the average vote proportion across all trees forms the predicted probabilities.

4.4 **Output:**

- The model returns a set of probabilities for each possible disease, indicating the likelihood of each disease given the input symptom data.
- These probabilities are then sorted in descending order to identify the top predicted diseases. Typically, only the top 3 probabilities are presented to give the most likely diagnoses.

This method leverages the ensemble nature of the Random Forest classifier to handle noisy and varied symptom input data effectively, providing robust predictions even when some symptoms may be missing or ambiguously presented. By taking into account both positive and negative symptom evidence, the classifier can make more informed predictions about the possible diseases.

Example prediction probabilities

Using the previous symptom status dictionary, the Random Forest classifier predicts the top disease probabilities as follows:

	Stomach ulcers	: 0.981
Predicted Disease Probabilities:	Cholera	: 0.003
	Hepatitis A	: 0.003

Based on the given symptoms, the model indicates a high likelihood of **Stomach ulcers** with a probability of 0.981, while the probabilities for Cholera and Hepatitis A are significantly lower.

5. Symptom Mapping:

To account for variations in symptom naming, the assistant uses semantic similarity to match doctor-inputted symptoms to those in the dataset, ensuring accurate data interpretation.

For example, considering the "coughing with blood" symptom in the dataset, we would like "blood while coughing" to have a higher semantic similarity score than merely "coughing".

To do this, we use embeddings generated by a Sentence Transformer model to capture the semantic meaning of each phrase, and cosine similarity then quantifies how closely these meanings align in vector space. This approach allows for a more nuanced similarity measure than simple keyword matching, as it captures context and relationships between words.

Results:
Semantic similarity(coughing with blood, blood while coughing) = 0.972
Semantic similarity(coughing with blood, coughing) = 0.748

6. Treatment Recommendation:

After determining the most probable disease, the assistant uses Retrieval Augmented Generation (RAG) to offer reliable treatment recommendations. This method combines retrieval and generation to give the doctor quick access to accurate, up-to-date medical advice. Here, RAG helps by first searching trusted sources like NHS Inform for treatment options related to the identified condition. The assistant retrieves this relevant information and then uses its language generation capabilities to present the details in a clear, context-appropriate manner, summarizing or rephrasing where needed.

Example Treatment Steps

Based on the prediction from the Random Forest classifier, the patient is most likely suffering from **stomach ulcers**. The treatment approach would depend on the underlying cause of the ulcer. Here are the recommended treatment options:

- If the ulcer is caused by a *Helicobacter pylori* (H. pylori) infection, you would prescribe:
  - A course of antibiotics (e.g., amoxicillin, clarithromycin, or metronidazole)
  - A medication called a proton pump inhibitor (PPI)
- If the ulcer is caused by non-steroidal anti-inflammatory drugs (NSAIDs), you would prescribe:
  - A proton pump inhibitor (PPI)
  - An alternative medication to NSAIDs, such as paracetamol
- In some cases, you may also recommend lifestyle changes, such as quitting smoking, as smoking can increase the risk of developing stomach ulcers and make treatment less effective.

5 Future Improvements

1. Targeting specialized medical domains:
For future improvements, the model could focus on a specialized medical field, such as cardiology, instead of covering a broad range of diseases. By narrowing down to cardiovascular conditions, we can refine the model using a dataset specific to heart-related symptoms like chest pain and shortness of breath. This approach can improve diagnostic accuracy and enable the integration of specialized tests, such as ECGs, enhancing the tool's effectiveness in predicting heart diseases and supporting tailored treatments.

2. Incorporating Additional Health Data:
Including broader health metrics, like age, blood sugar levels, blood pressure, heart rate, and other general health parameters, can provide a more comprehensive basis for predictions. These metrics are crucial because they often correlate strongly with the presence and severity of various diseases. For example, chronic illnesses like diabetes, cardiovascular disease, and hypertension are often linked to these parameters. By integrating this data, the model could improve the accuracy of disease probability predictions, allowing it to identify high-risk cases more precisely.

3. Integrating Explainable AI:
Adding Explainable AI capabilities can make the model's decision-making process more transparent and interpretable for medical professionals. In disease prediction, explainable AI techniques such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) could highlight which symptoms and health parameters contributed most to the prognosis. This transparency can help healthcare providers trust and validate the tool's recommendations, as it provides insights into why certain diseases are being flagged.

6 Conclusion

In this project, a disease diagnosis assistant was developed to help streamline the initial prognosis process for doctors. By using a machine learning model, specifically a Random Forest classifier, the system predicts the probable disease based on symptoms provided by the user. Through the use of semantic similarity models, the assistant can understand various symptom descriptions, even if they differ slightly from those in the dataset. Additionally, treatment steps are generated through retrieval-augmented generation (RAG), accessing reliable resources such as NHS Inform to provide accu-

rate and comprehensive treatment information.

This project has demonstrated the potential for AI-driven tools to support medical professionals in diagnosis and treatment planning, with future improvements likely to increase its effectiveness and trustworthiness in real-world applications.

7 References

1. K. Gaurav, A. Kumar, P. Singh, A. Kumari, M. Kasar, T. Suryawanshi, "Human Disease Prediction using Machine Learning Techniques and Real-life Parameters."
2. NHS Inform, "A to Z Illnesses and Conditions," available at: `https://www.nhsinform.scot/illnesses-and-conditions/a-to-z/`, for treatment information on various common diseases.