ML Things I have done for this SIH project

Three features I have inculcated:

1. **Symptom-based disease prediction** → takes structured symptom input (0/1) and predicts disease.
2. **Hospital recommendation** → suggests hospitals based on user input like district, specialty, cost preference.
3. **Text-based disease prediction (NLP)** → takes free-text symptoms (English or Malayalam) and predicts disease.

1. Disease Risk Prediction Model

Dataset fetched from Kaggle- [https://www.kaggle.com/datasets/dhivyeshrk/diseases-and-symptoms-dataset](https://www.kaggle.com/datasets/dhivyeshrk/diseases-and-symptoms-dataset)
The original dataset had 773 Unique Diseases and 377 One-Hot Encoded Symptoms with 246,000 samples.
But I preprocessed the data shortening it to 6000 rows and top 29 symptoms for better accuracy and model learning.

Because no patient would enter 377 symptoms, also I drop down those diseases not prevalent in Kerala and only kept those which were as our Problem Statement is for Government of Kerala.

Total symptoms: 29

Symptom list:

1. cough
2. fever
3. shortness of breath
4. vomiting
5. sharp chest pain
6. sore throat
7. difficulty breathing
8. coryza
9. chills
10. nausea
11. nasal congestion
12. headache
13. wheezing
14. weakness
15. sharp abdominal pain
16. feeling ill
17. congestion in chest
18. coughing up sputum
19. ache all over
20. suprapubic pain
21. decreased appetite
22. chest tightness
23. diarrhea
24. difficulty in swallowing
25. depression
26. itching of skin
27. rectal bleeding
28. regurgitation

29. regurgitation.1

Although this dataset is highly aligned with real world medical data still it’s just a prototype data as real world data is not available due to privacy concerns.

The algorithm used is Random forest Classifier with an accuracy of 91%.

2. Hospital Recommendation Model

Dataset fetched from- https://www.data.gov.in/

The data was preprocessed to filter out only Kerala hospitals

Result: **890 hospitals**
Added some other columns
**1. Languages** -added a default "Malayalam, English" since these are universally available in Kerala hospitals.

2. **Cost_Category**-

2.1 If Hospital_Category = *Government* → "Free/Low Cost"
2.2 Else → "Paid/Private

3**. Supports_Immigrants**- set to "Yes" assumed all hospitals provide at least some migrant worker support.

The dataset contains 890 sample rows and 16 columns.

� **District filtering** → works well, since every row has a district.

� **Cost filtering** → works, thanks to enrichment (Government = free/low cost, else private).

� **Specialty-based search (TF-IDF)** → partially works, but limited since real specialties are often missing.

- Example: "Cardiology" hospitals may not always be labeled.
- Many hospitals just fall back to their name.

� **Languages / immigrant support** → usable.

Hospital recommendation system is based on a **Content-Based Filtering** approach using **text similarity**.

1. **TF-IDF (Term Frequency – Inverse Document Frequency)**
    - To convert the *hospital specialties* text into numerical vectors.
    - Example: "Cardiology, Neurology" → vector representation.
2. **Cosine Similarity**

- To measure how similar the user's query (e.g., "Cardiology") is to each hospital's specialties.
- Values range from **0 (no match)** to **1 (perfect match)**.

3. **Filtering** (Optional)
   - After computing similarity, you filtered hospitals by **district** and **cost category** if the user provided them.

So the recommendation logic is:

User query → TF-IDF vector → cosine similarity with hospitals → filter by district/cost → rank & recommend.

No machine learning model (like Random Forest or Deep Learning) is being used here — it's purely **NLP-based similarity search**.

We didn't use heavy NLP models (like BERT, GPT, or transformers), we used **traditional NLP techniques** (TF-IDF + cosine similarity) for text-based matching.

In short: **Yes, this is NLP (classical NLP), not deep learning NLP.**

3.Text Based Disease Prediction using NLP

Dataset used- The same dataset which was used in model 1, but preprocessing was done to convert columns into our text based inputs.

**Logistic Regression** used

**Feature Extraction with TF-IDF**

- **TF (Term Frequency):** how often a word appears in a text.
- **IDF (Inverse Document Frequency):** how unique that word is across all documents.

TF-IDF turns each symptom text into a **vector of word importance**.
Example:

- Input: "fever cough sore throat"
- After TF-IDF: [0.5, 0.7, 0.3, ...] (vector of numbers)

**4,853 samples** in training set.

**1,214 samples** in test set.

**Shape (4853, 47)** means:

- 4,853 training samples
- 47 unique features (words/terms) were extracted from your symptom text

**Key Observations:**

1. **Overall accuracy**: 0.94 → very high for a TF-IDF + Logistic Regression model.
2. **High precision & recall** for common diseases like acute bronchitis, atrial flutter, flu, pneumonia, etc.
3. **Very low or zero scores** for rare diseases with **very few samples** (e.g., frostbite, malaria, tuberculosis, typhoid fever) → expected because the model has almost no examples to learn from.
4. **Weighted avg** is 0.93 → reflects good performance across all classes, while **macro avg** 0.68 shows rare classes reduce the average.

Accuracy- 94%