INTRODUCTION

- **This project is a web-based disease prediction system designed to assist in early disease detection.**

- **Early detection plays a crucial role in effective disease management and treatment outcomes. It allows for timely intervention and reduces the risk of complications.**

- **A web-based platform offers accessibility and convenience for users to access the system from anywhere, enabling early detection and informed decision-making regarding their health.**

- **With the increasing availability of digital health data and advancements in machine learning, an automated prediction system can efficiently analyze symptoms and provide potential disease predictions**

TECHNOLOGY STACK

DATAFLOW

01

USER INTERACTION:

Users interact with the frontend interface by selecting symptoms and submitting the form.

02

DATA SUBMISSION:

The front end captures the user-selected symptoms and sends a request to the backend.

03

BACKEND PROCESSING:
The backend receives the symptom data, preprocesses it if necessary, and passes it to the trained machine learning model.

04

DISEASE PREDICTION:

The model predicts the potential disease based on the user's symptoms and returns the result to the backend.

05

RESULT DISPLAY:
The backend sends the predicted disease result back to the frontend for display to the user.

MODEL DEVELOPMENT

- Multinomial Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes that the features (symptoms) are conditionally independent given the class (disease).

- This classifier is suitable for text classification tasks, including disease prediction, where the features (symptoms) can be represented as discrete counts or frequencies.

- Multinomial Naive Bayes calculates the probability of each disease given the symptom inputs and selects the disease with the highest probability as the predicted disease.
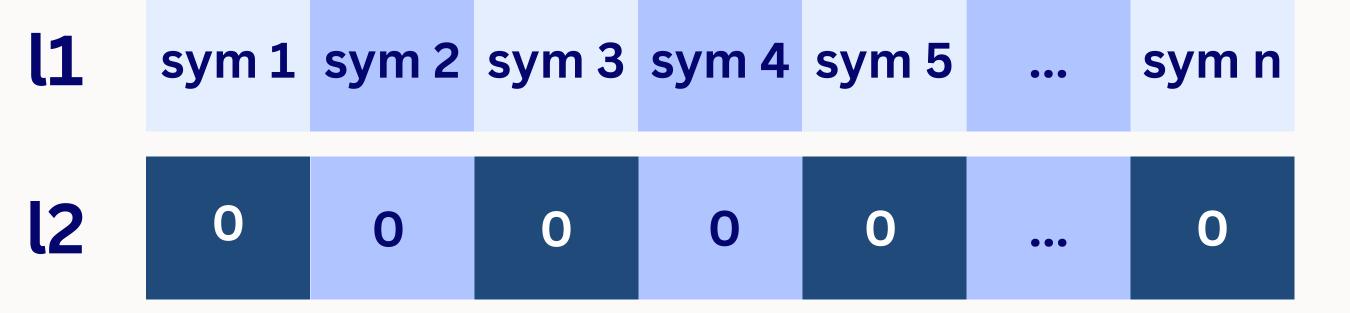
TRAINING PROCESS

- The model is trained using a labeled dataset that includes symptom data and corresponding disease labels.

- The dataset is obtained from a study conducted by Columbian University.

- The labeled data is split into training and testing sets, with a portion reserved for evaluating the model's performance.

- During training, the model learns the statistical patterns and relationships between symptoms and diseases.

WORKING

- Import the required packages.

- Create a list (l1) that contains all the symptoms as the CSV file.

- Create another list that contains the diseases.

- Then, create an empty list L2 having the same length as l1 wherein all the elements are initialized to 0

l1	sym 1	sym 2	sym 3	sym 4	sym 5	...	sym n
l2	0	0	0	0	0	...	0

WORKING

- Perform the same steps for both testing and training datasets.

- Using pandas, read the CSV file

- Replace the disease name with the index, this will work as the label.

- Rest of the data will work as the attributes

s1	s2	s3	s4	s5	...	prognosis
0	1	0	0	1	...	0
1	1	0	0	1	...	1
0	0	1	1	0	...	2