PROJECT

Group - 2

NAME	ROLL NO	ROLE IN PROJECT	REMARKS
PRAKHAR JADAUN	R214220832	Design and implementation	
PRAKRATI SINGH	R214220840	Design and implementation	
ASHUTOSH SINGH SINGH KUSHWAH	R214220293	Design and implementation	

DISEASE PREDICTION AND QUESTION ANSWERING SYSTEM IN THE HEALTHCARE DOMAIN

Healthcare workers and patients may find a disease prediction and question-answering system to be useful tool. Such a system may answer queries about various medical disorders and offer insights into suspected diseases based on symptoms, medical history, and other pertinent data.

DATASET

For Disease Prediction:

The complete Dataset consists of 2 CSV files. One of them is for training and the other is for testing your model.

Each CSV file has 133 columns. 132 of these columns are symptoms that a person experiences and the last column is the prognosis.

These symptoms are mapped to 42 diseases you can classify these sets of symptoms.

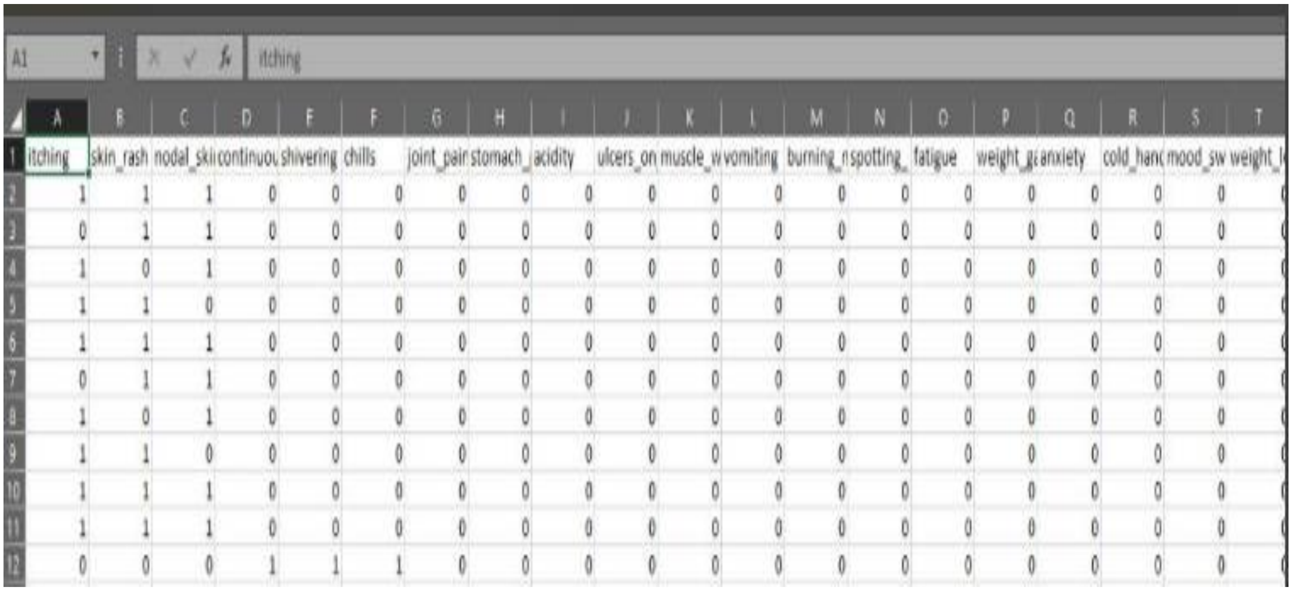
You are required to train your model on training data and test it on testing data.

LINK TO THE DATASET- Testing.csv

For Question Answering System:

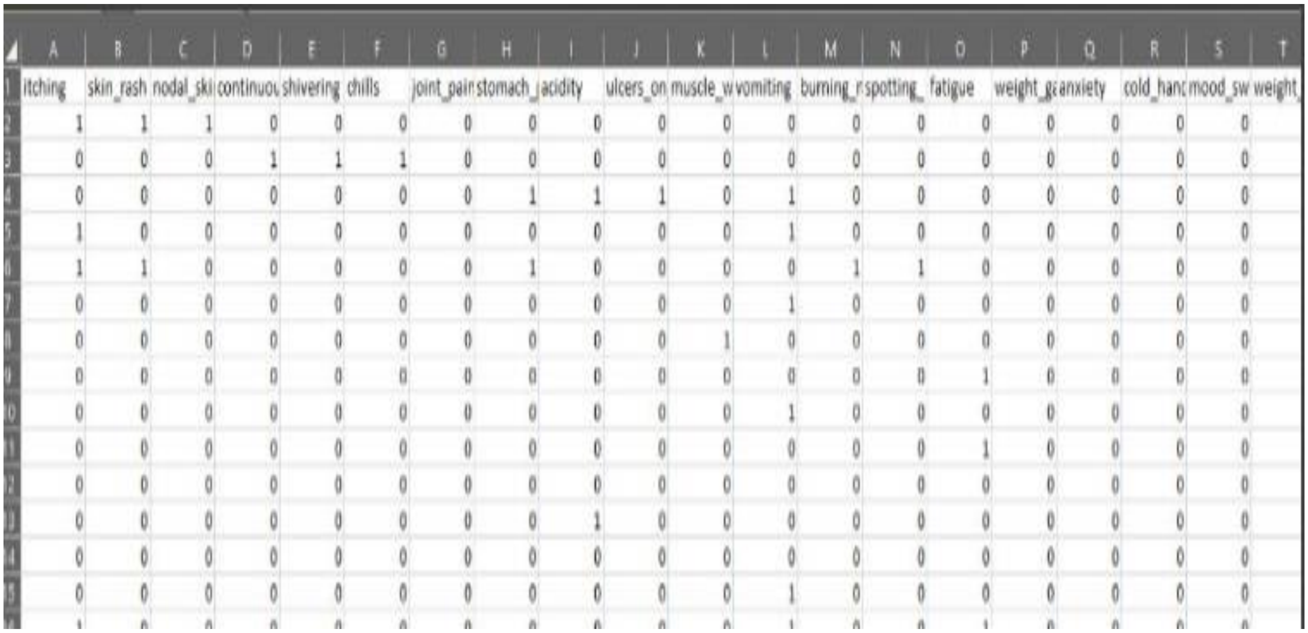
The dataset is divided into six .json files and then merged together to form a larger dataset on which the model can be trained.

TRAINING:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	itching	skin_rash	nodal_ski	continuou	shivering	chills	joint_pain	stomach_	acidity	ulcers_on	muscle_w	vomiting	burning_r	spotting_	fatigue	weight_g	anxiety	cold_hanc	mood_sw	weight_l
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TESTING:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	itching	skin_rash	nodal_ski	continuou	shivering	chills	joint_pain	stomach_	acidity	ulcers_on	muscle_w	vomiting	burning_r	spotting_	fatigue	weight_g	anxiety	cold_hanc	mood_sw	weight_
2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	
5	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
6	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
16	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	

METHODOLOGY

DISEASE PREDICTION

- Column Preprocessing:
  Perform preprocessing steps on the columns or features of your dataset, which may include stemming, stop word removal, and lemmatization. These techniques help standardize and normalize the text data, reducing noise and improving the quality of features.

- Model Selection:
  Choose the Bernoulli Naïve Bayes model for disease prediction. Bernoulli Naïve Bayes is a variant of Naïve Bayes that assumes binary features (present or absent) and is commonly used for text classification tasks.

- Input Vector Preparation:
  Convert the textual input data, such as symptoms or patient information, into a numerical representation. This process involves transforming the text into a suitable format that the model can understand.

- Input Preprocessing:
  Preprocess the input data by removing numbers and special characters, as they may not provide meaningful information for disease prediction. Additionally, apply techniques like stemming, stop word removal, and lemmatization to further refine the input text.

- Conversion to Test Vector:
  Convert the preprocessed input text into a test vector that can be fed into the Bernoulli Naïve Bayes model. The test vector should match the format and structure of the training data to ensure compatibility with the model's requirements.

- Prediction using Bernoulli Naïve Bayes:
  Utilize the trained Bernoulli Naïve Bayes model to make predictions on the test vector. The model uses probabilistic calculations based on the training data to estimate the likelihood of different diseases given the input features.

Check whether the dataset is imbalanced or not.

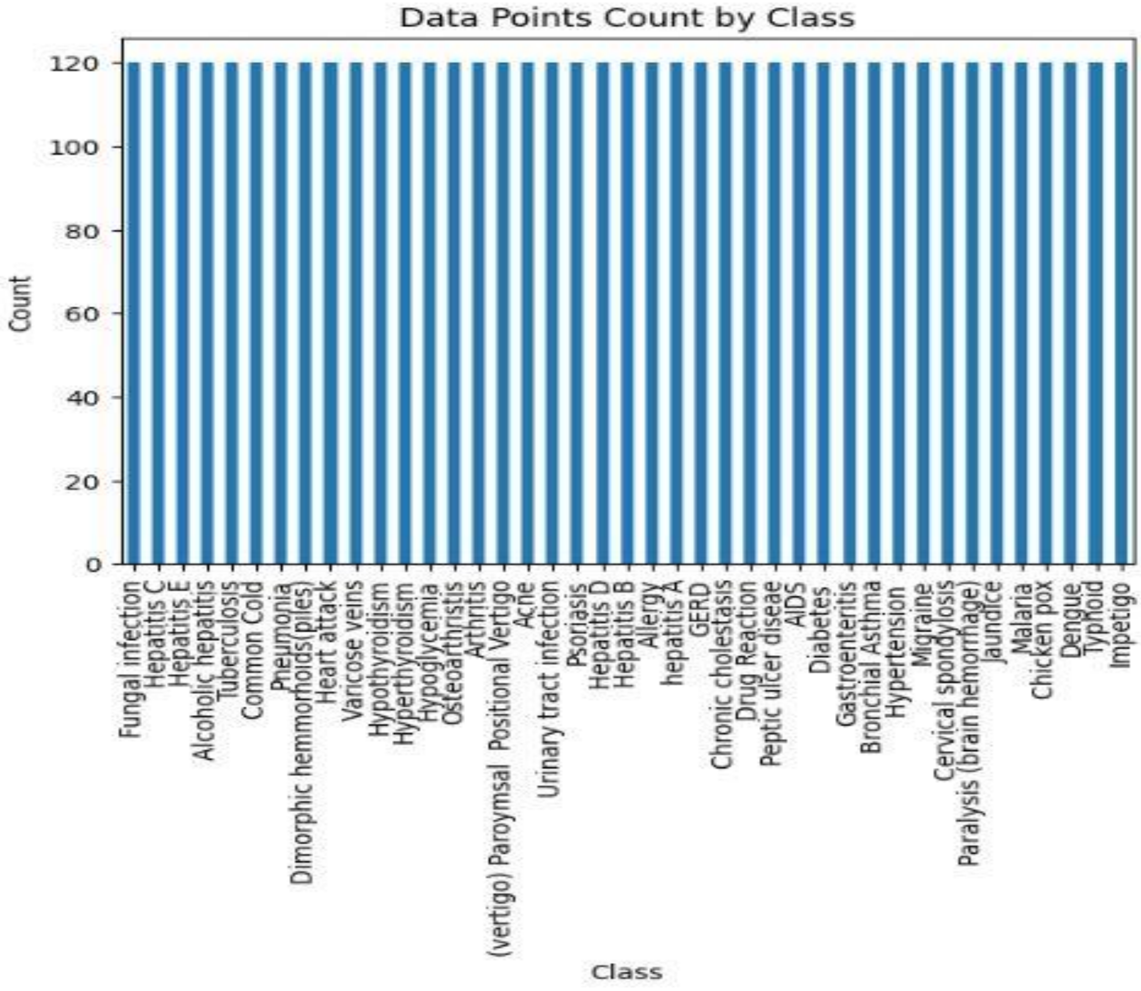
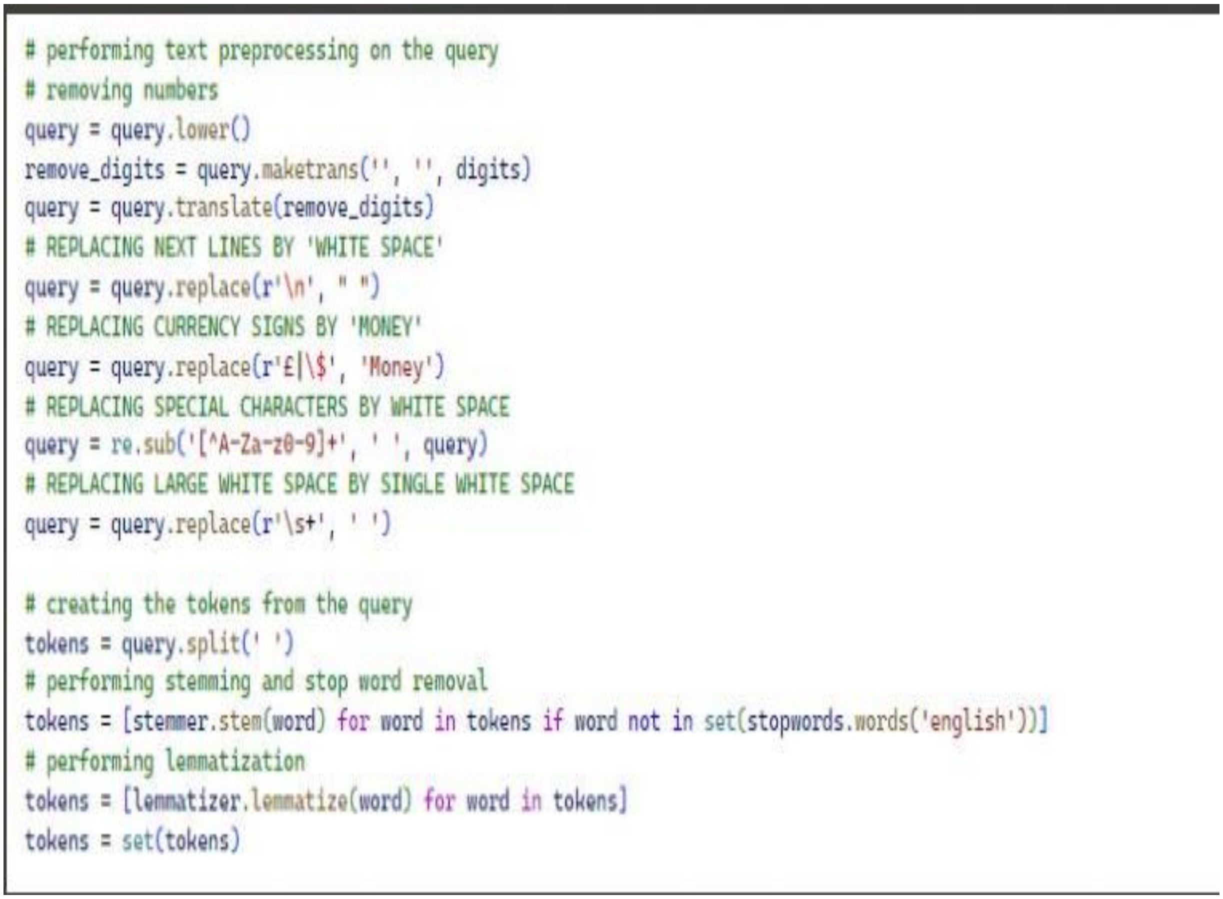


Fig 1: Dataset



```python
# performing text preprocessing on the query
# removing numbers
query = query.lower()
remove_digits = query.maketrans('', '', digits)
query = query.translate(remove_digits)
# REPLACING NEXT LINES BY 'WHITE SPACE'
query = query.replace(r'\n', " ")
# REPLACING CURRENCY SIGNS BY 'MONEY'
query = query.replace(r'£|\$', 'Money')
# REPLACING SPECIAL CHARACTERS BY WHITE SPACE
query = re.sub('[^A-Za-z0-9]+', ' ', query)
# REPLACING LARGE WHITE SPACE BY SINGLE WHITE SPACE
query = query.replace(r'\s+', ' ')

# creating the tokens from the query
tokens = query.split(' ')
# performing stemming and stop word removal
tokens = [stemmer.stem(word) for word in tokens if word not in set(stopwords.words('english'))]
# performing lemmatization
tokens = [lemmatizer.lemmatize(word) for word in tokens]
tokens = set(tokens)
```

Fig 2: Preprocessing

QUESTION ANSWERING SYSTEM

- Merging of Training Data:
  Collect and merge the training data, which consists of question-answer pairs, into a single dataset. The data is representative and covers a wide range of topics within the healthcare domain.

- Preprocessing the "Question" Column:
  Apply preprocessing techniques to the "question" column of the training data. This typically involves stopping word removal, stemming, and lemmatization to standardize and normalize the text data. These steps help in reducing noise and improving the quality of the questions.

- User Query Input:
  Obtain the user query, which is the question asked by the user for which an answer is sought.

- Preprocessing the User Query:
  Preprocess the user query using the same steps performed on the "question" column of the training data. This involves removing stop words, stemming, and lemmatization to prepare the user query for further processing.

- Computing Similarity with Cosine Similarity:
  Compute the similarity between the preprocessed user query and each question in the training data using the cosine similarity metric. Cosine similarity measures the similarity between two vectors by calculating the cosine of the angle between them. It determines how closely the user query matches each question in the training data.

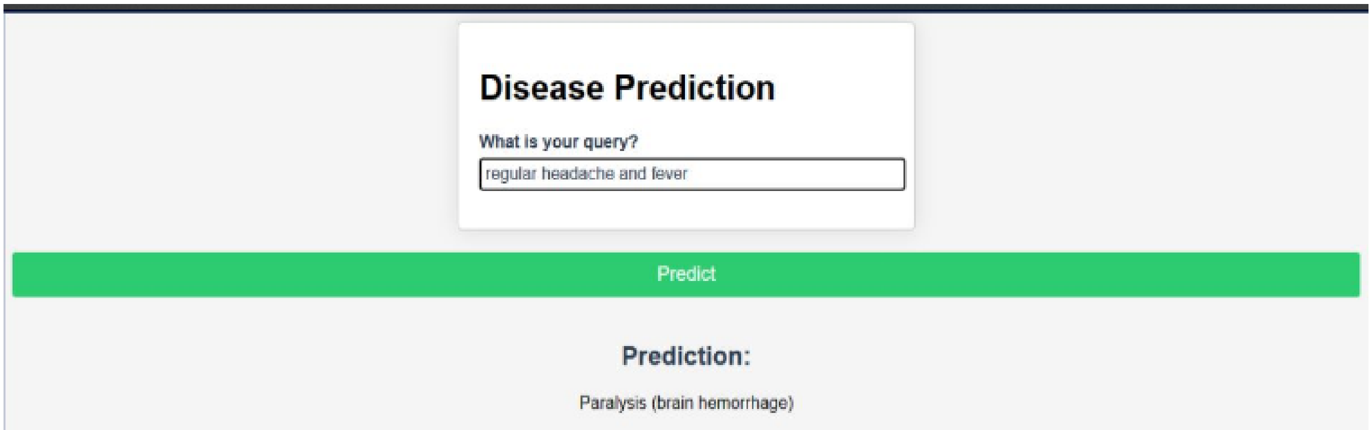
- Obtaining Similarity Scores:
  Retrieve the similarity score for each question in the training data based on the cosine similarity calculation. The similarity score represents the degree of similarity between the user query and each question.

- Retrieving the Answer:
  Identify the question in the training data with the highest similarity score compared to the user query. This indicates that the question in the training data is most similar to the user query.

- Retrieving the Answer Corresponding to the Question:
  Once the question with the highest similarity score is identified, retrieve the corresponding answer from the training data. This answer is considered the most relevant response to the user query.

RESULT

Based on the user input the system is able to predict the disease with which the user might be suffering from.



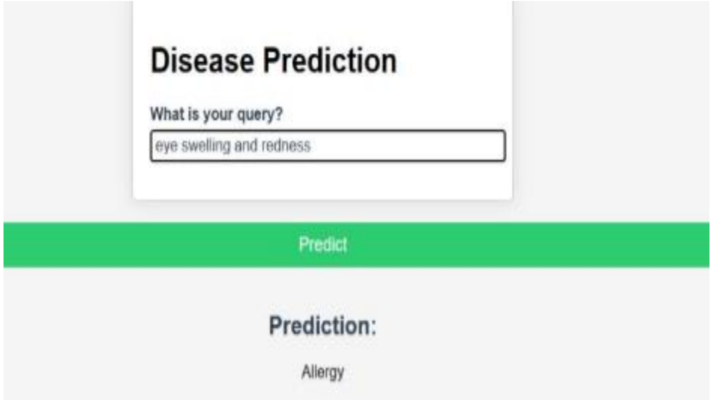
Disease Prediction

What is your query?

regular headache and fever

Prediction:

Paralysis (brain hemorrhage)



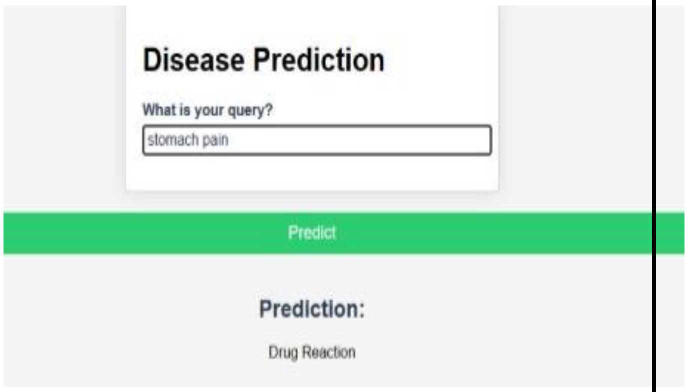
Disease Prediction

What is your query?

eye swelling and redness

Prediction:

Allergy



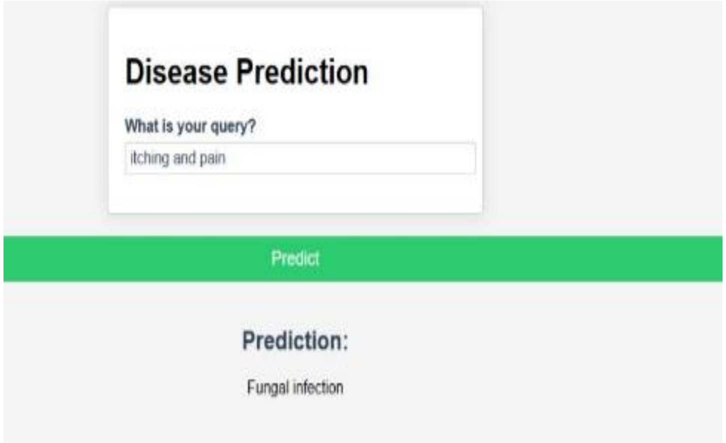
Disease Prediction

What is your query?

stomach pain

Prediction:

Drug Reaction



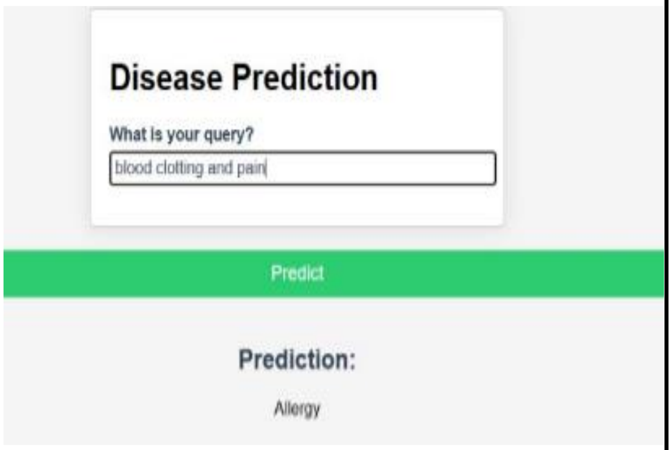
Disease Prediction

What is your query?

itching and pain

Prediction:

Fungal infection



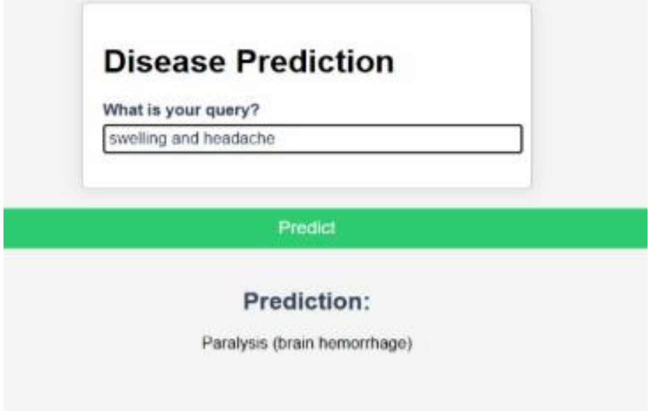
Disease Prediction

What is your query?

blood clotting and pain

Prediction:

Allergy



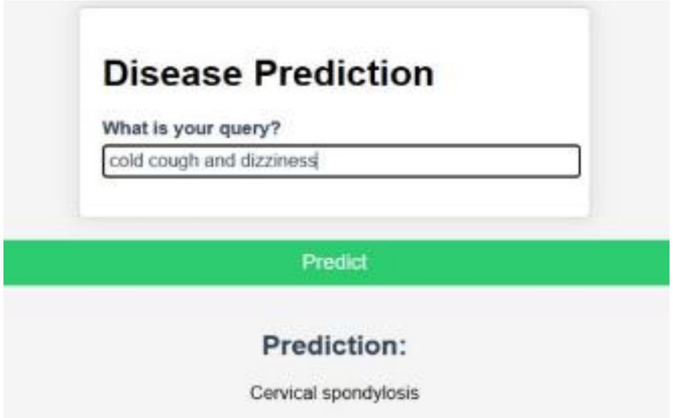
Disease Prediction

What is your query?

swelling and headache

Prediction:

Paralysis (brain hemorrhage)



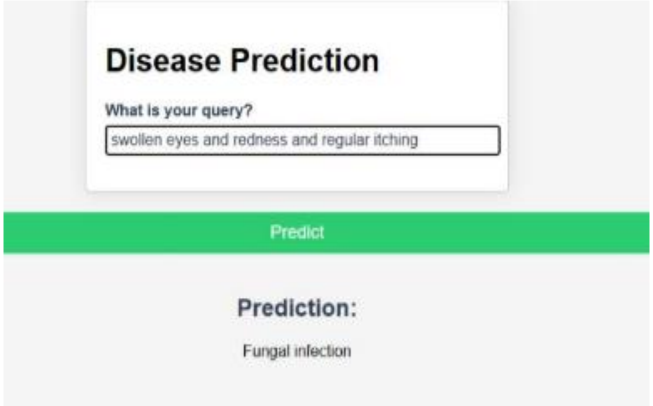
Disease Prediction

What is your query?

cold cough and dizziness

Prediction:

Cervical spondylosis



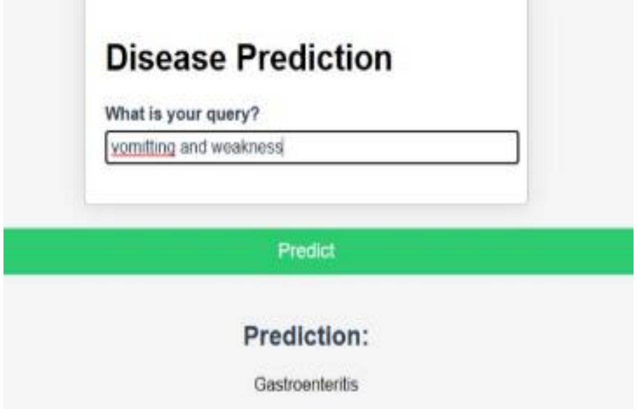
Disease Prediction

What is your query?

swollen eyes and redness and regular itching

Prediction:

Fungal infection



Disease Prediction

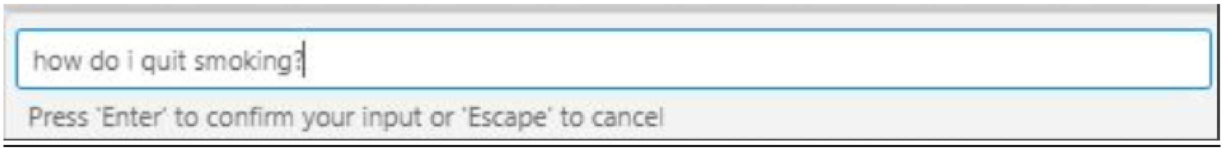
What is your query?

vomitting and weakness

Prediction:

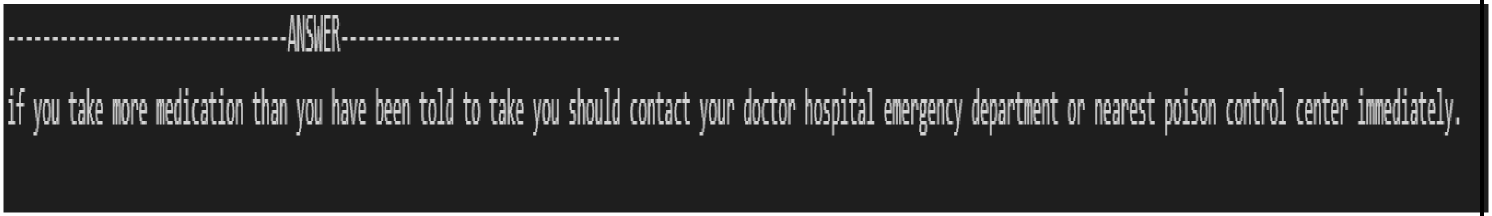
Gastroenteritis

QA SYSTEM OUTPUT:



how do i quit smoking?

Press 'Enter' to confirm your input or 'Escape' to cancel



```
-------------------------------ANSWER-------------------------------
if you take more medication than you have been told to take you should contact your doctor hospital emergency department or nearest poison control center immediately.
```

Challenges or learning throughout the project development

Developing a disease prediction and question-answering system in the healthcare domain can come with various challenges and learning opportunities.

- Availability and Quality of Data: Obtaining comprehensive and reliable healthcare data can be a significant challenge. Access to diverse and well-annotated datasets for training disease prediction models and question-answering systems is crucial.
- Data Preprocessing and Feature Extraction: Preprocessing and feature extraction from healthcare data can be complex due to the unstructured nature of medical records and the presence of noise.
- Model Selection and Performance: Selecting the appropriate machine learning or deep learning models for disease prediction and question answering requires careful consideration.
- System Integration and Deployment: Integrating the disease prediction and question-answering system into existing healthcare workflows, electronic health record systems, or online platforms can be complex. Ensuring seamless integration, scalability, and reliability while considering security and privacy requirements pose additional challenges.
- Continuous Learning and Updates: The healthcare domain is constantly evolving with new research findings, guidelines, and treatment approaches. Ensuring that the system remains up to date with the latest medical knowledge requires continuous learning, regular updates, and monitoring performance to address new diseases, symptoms, and advancements in healthcare practices.

Throughout the project, we encounter various learning opportunities, such as gaining a deeper understanding of healthcare workflows, refining data preprocessing techniques, exploring advanced machine learning or NLP models, and understanding the importance of ethical considerations in healthcare AI applications.

Collaboration with healthcare professionals, conducting user studies, and seeking feedback from stakeholders are also valuable learning opportunities to improve the system's performance, usability, and impact in real-world healthcare settings.