

Group : Mikes

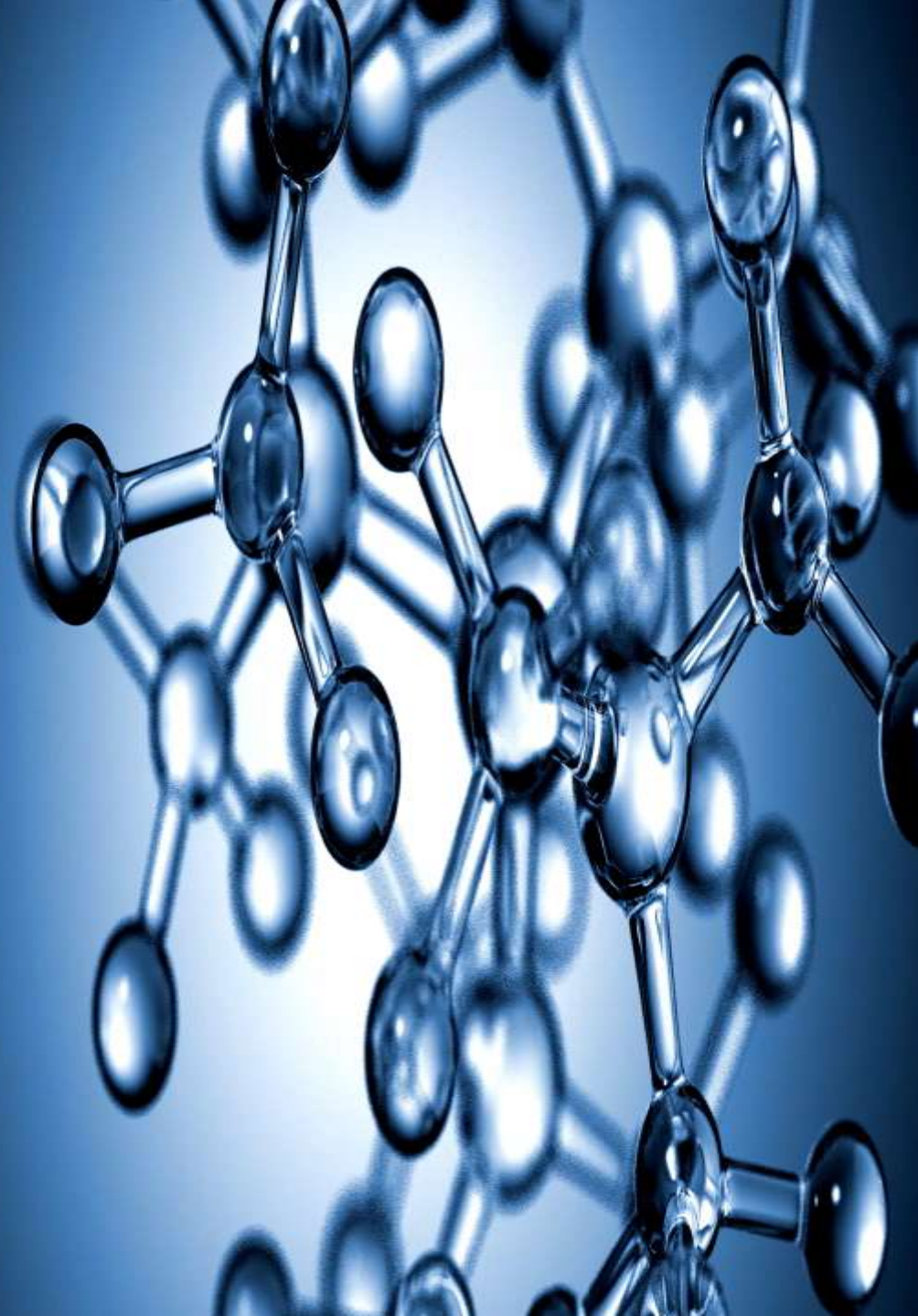
Team Members

Neelima Kommareddy

Sindhu Anugula

Kalwa Sailaxmi

Healthcare Chatbot: Utilizing Artificial Intelligence for Medical Diagnosis



Statement

- ▶ The purpose of this project is to develop an AI-powered healthcare chatbot that can help with diagnosis. In order to give patients a convenient way to submit their symptoms and obtain a preliminary diagnosis, the chatbot will be built. If necessary, the chatbot will also offer suggestions for additional medical assessment. Enhancing the precision and effectiveness of medical diagnosis is the aim of this initiative, particularly when patients do not have quick access to a healthcare provider

Approach

- ▶ **Data Collection** : - Collecting a large dataset of medical records and symptoms to train our AI model.
- ▶ **Model Development** : - Developing a deep learning model that uses natural language processing to analyze patient symptoms and provide accurate diagnoses.
- ▶ **Integration with Chatbot** :-Integrate our AI model with a chatbot interface to provide users with an easy-to-use and accessible platform for medical diagnosis.

Deliverables

- ▶ **Functional Healthcare Chatbot** :- Our team will develop a functional healthcare chatbot that utilizes AI to assist in medical diagnosis.
- ▶ **Integration with Medical Databases** :- The chatbot will be integrated with various medical databases to provide accurate and up-to-date information.
- ▶ **User Testing Results** :- We will provide user testing results to evaluate the effectiveness and efficiency of our chatbot.

Evaluation Methodology

- ▶ **Accuracy Metrics** :- The accuracy of the chatbot's diagnosis will be evaluated using metrics such as precision, recall, and F1 score.
- ▶ **User Feedback** :- User feedback will be collected through surveys and interviews to evaluate the chatbot's usability, effectiveness, and overall satisfaction.
- ▶ **Data Analysis** :- We will analyze the data collected by the chatbot to identify trends and patterns in medical diagnoses.

A large, dark gray puzzle piece is centered on a dark gray background. In the top right corner of the puzzle piece, there is a small red rectangular tab. The puzzle piece has a complex shape with several interlocking points.

Advantages

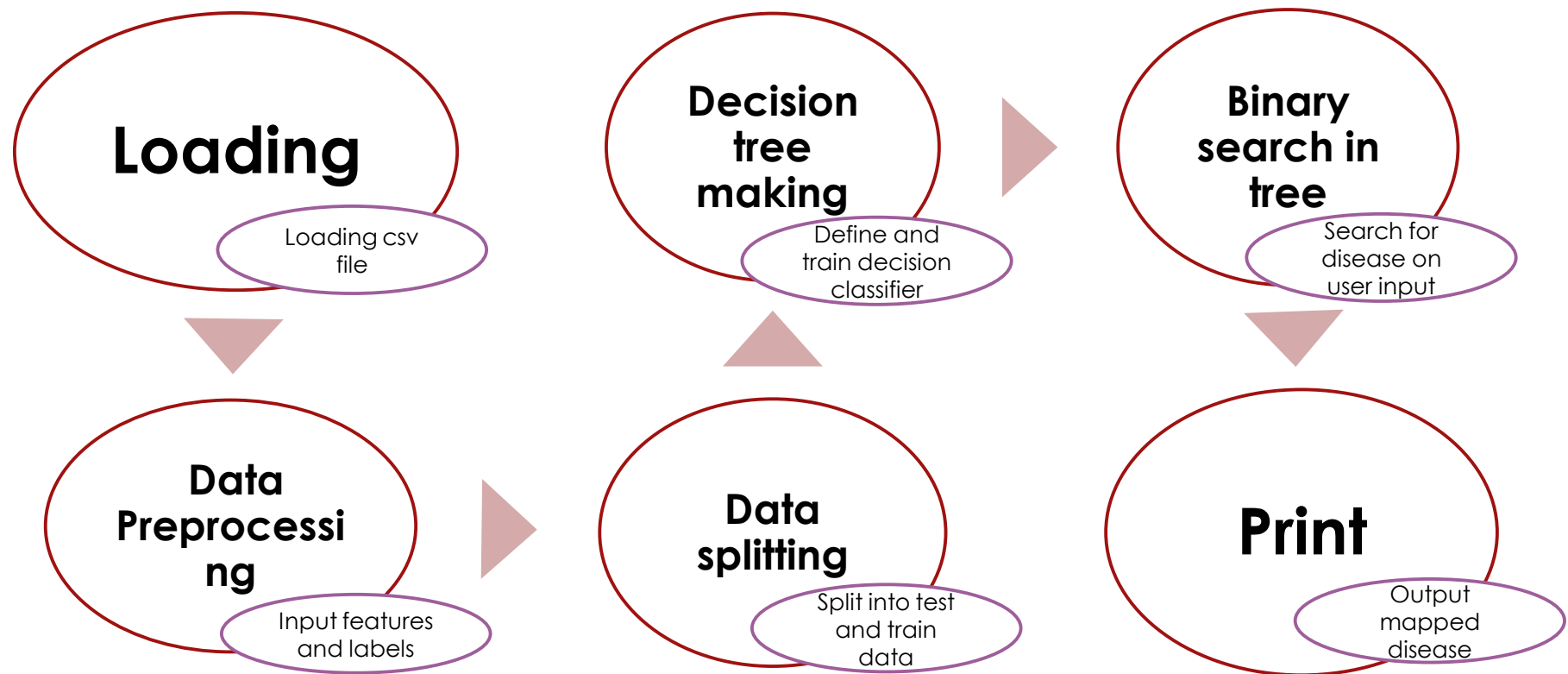
- ▶ 24/7 Accessibility
- ▶ Instant Response
- ▶ Cost-Efficiency
- ▶ Personalized Assistance
- ▶ Reduced Anxiety

Limitations

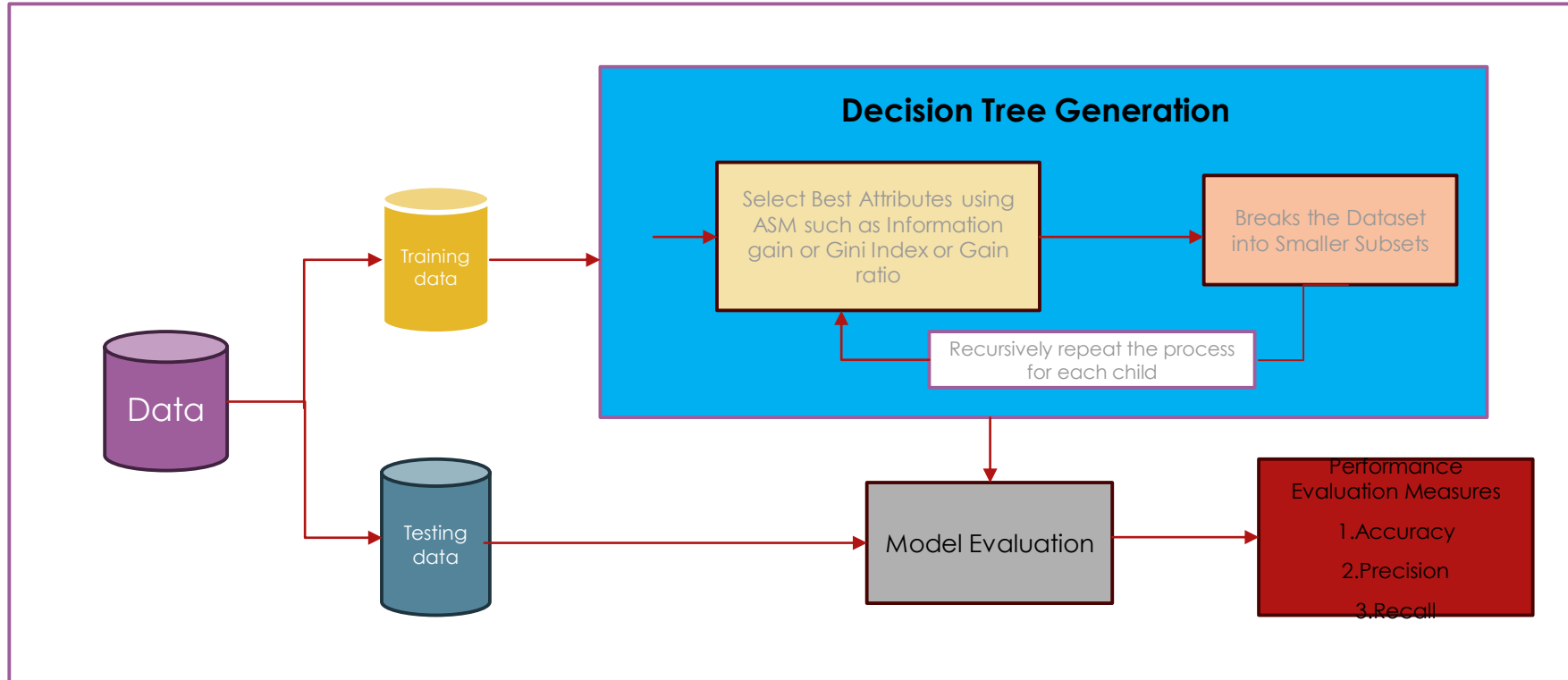
- ▶ **Lack of Empathy**
- ▶ **Limited Understanding of Nuances**
- ▶ **Security and Privacy Concerns**
- ▶ **Dependency on Data Quality**
- ▶ **Inability to Handle Emergency Situations**
- ▶ **User Trust and Adoption**



Overview of the Model



Decision Tree



Factors to Consider

Information gain :- The decrease in entropy is known as information gain. Based on specified attribute values, information gain computes the difference between entropy before split and average entropy after split of the dataset. (In a system, entropy is the randomness)

$$\text{Info}(D) = - \sum_{i=1}^m p_i \log_2 p_i$$

Gini Index :- The Gini Index takes each attribute into account as a binary split. A weighted total of each partition's impurity can be calculated.

$$\text{Gini}(D) = 1 - \sum_{i=1}^m p_i^2$$

Dataset

- ▶ Training dataset : 113 features , 4920 samples
- ▶ Testing dataset: 113 features, 41 samples
- ▶ As a label, the word "prognosis" is utilized.
- ▶ Label Encoder is a program that converts labels into numerical values.

Before Label Encoder	After Label Encoder
Migraine	1
Heart Attack	2
Typhoid	3
Acne	4

Cross validation , Testing and Training

```
clf1 = DecisionTreeClassifier()
```

```
clf2 = clf1.fit(x_train,y_train)
```

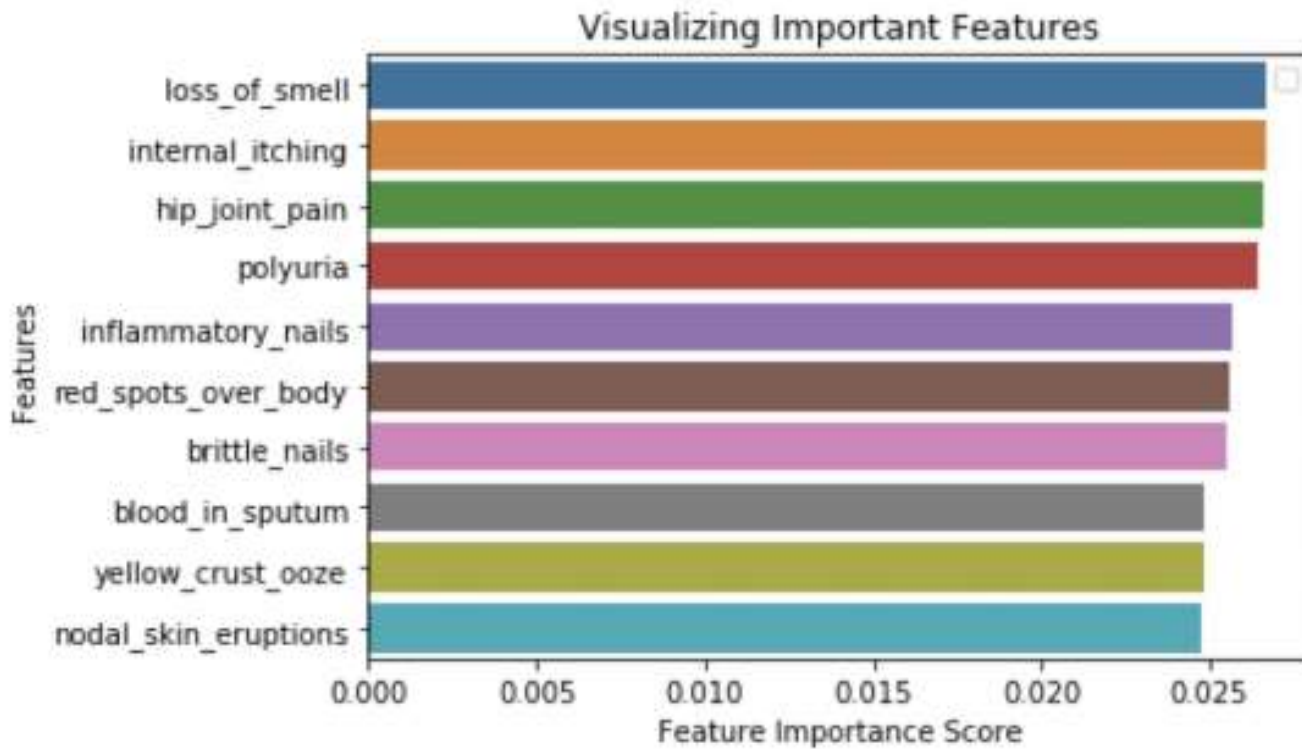
```
print(clf2.score(xtest,ytest))
```

Cross-validation:

```
scr = cross_val_score(clf2, x_test, y_test, cv=3)
```

```
Print(scr.mean())
```


Results



Accuracy of training: 0.97

Accuracy of testing: 0.94

Important Features

Decision Tree

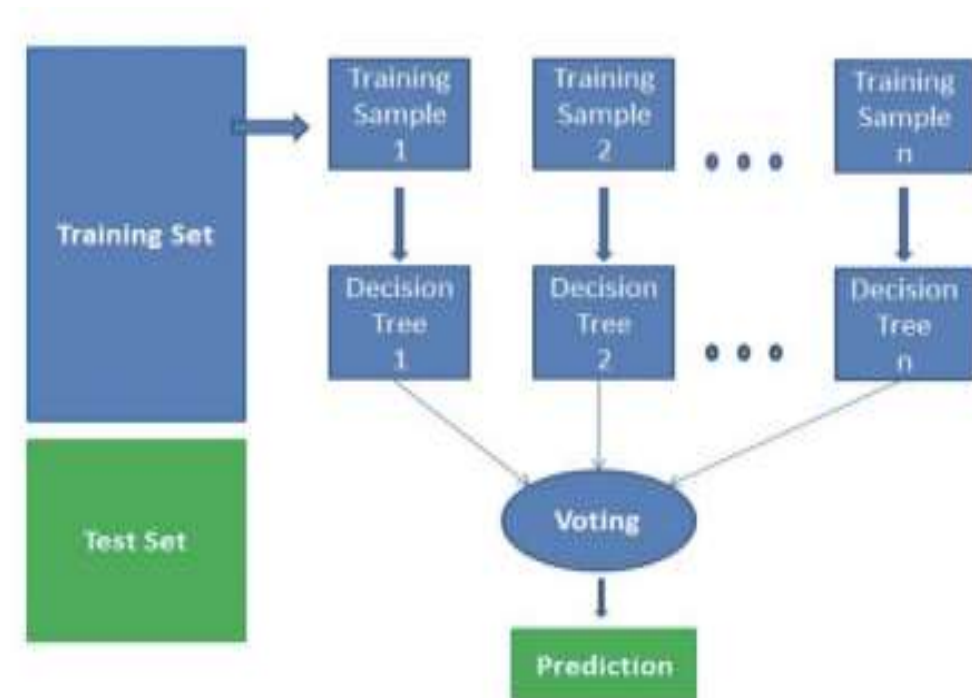


Important Function(code)

- ▶ **Binary search in tree():** A `Decision_tree_bot()` subfunction that does binary search based on user responses to the bot's questions.
- ▶ **Print disease to user():** This function converts the decision tree output into one of the user's diseases.
- ▶ **Decision tree bot():** The Decision Tree Primary Logic Main() : When the program is run, the `main()` function is executed

Future Enhancements

- ▶ Random Forest can be employed to address overfitting issues associated with noisy data in Decision Trees. However, it's important to note that ensemble techniques like Random Forest are more effective when applied to extensive datasets.



Thank You

