# DISEASE PREDICTION SYSTEM

**MOTIVATION**: Hospitals nowadays have exceedingly long wait times and people end up spending hours at the opd for very minute injuries or diseases. This system can help overcome this by providing people with an accessible platform to provide solutions for their symptoms. Over the past decade, the use of specific disease prediction tools and the concerning health has increased due to a variety of diseases and low doctor-patient ratio. Thus, this system can provide immediate and accurate disease prediction depending on the person's symptoms and can also suggest diets, medicines, and precautions to them.

## IMPLEMENTATION:

In this project, the standard libraries for database analysis and model creation are used.

1. Numpy: Numpy is the core library of scientific computing in Python. It provides powerful tools to deal with various multi-dimensional arrays in Python. It is a general-purpose array processing package. It is mainly used to handle arrays. Data manipulation occurring in arrays while performing various operations needs to give the desired results while predicting outputs requires such high operational capabilities.

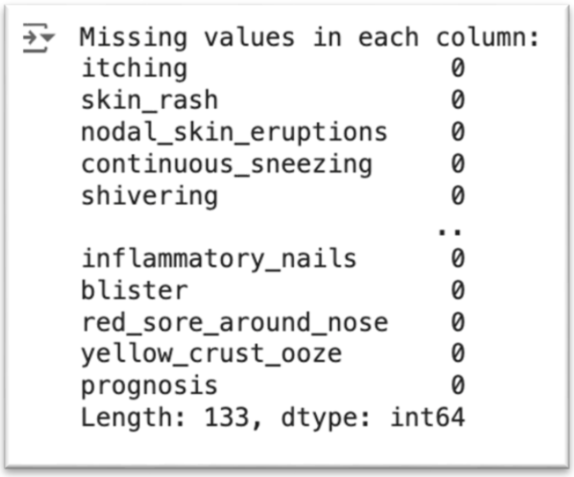
2.Pandas: Panda's data frame is used extensively in this project to use datasets needed for training and testing the algorithms. Dataframes make it easier to work with attributes and results. Several of its inbuilt functions such as replace were used in our project for data manipulation and preprocessing.

3.Sklearn: Sklearn is an open-source Python library with implements a vast range of machine learning, pre-processing, cross-validation and visualization algorithms. Using sklearn we get the advantage of using the inbuilt classification algorithms like Naive Bayes,knn,Gradient boosting etc.

**PREPROCESSING:**

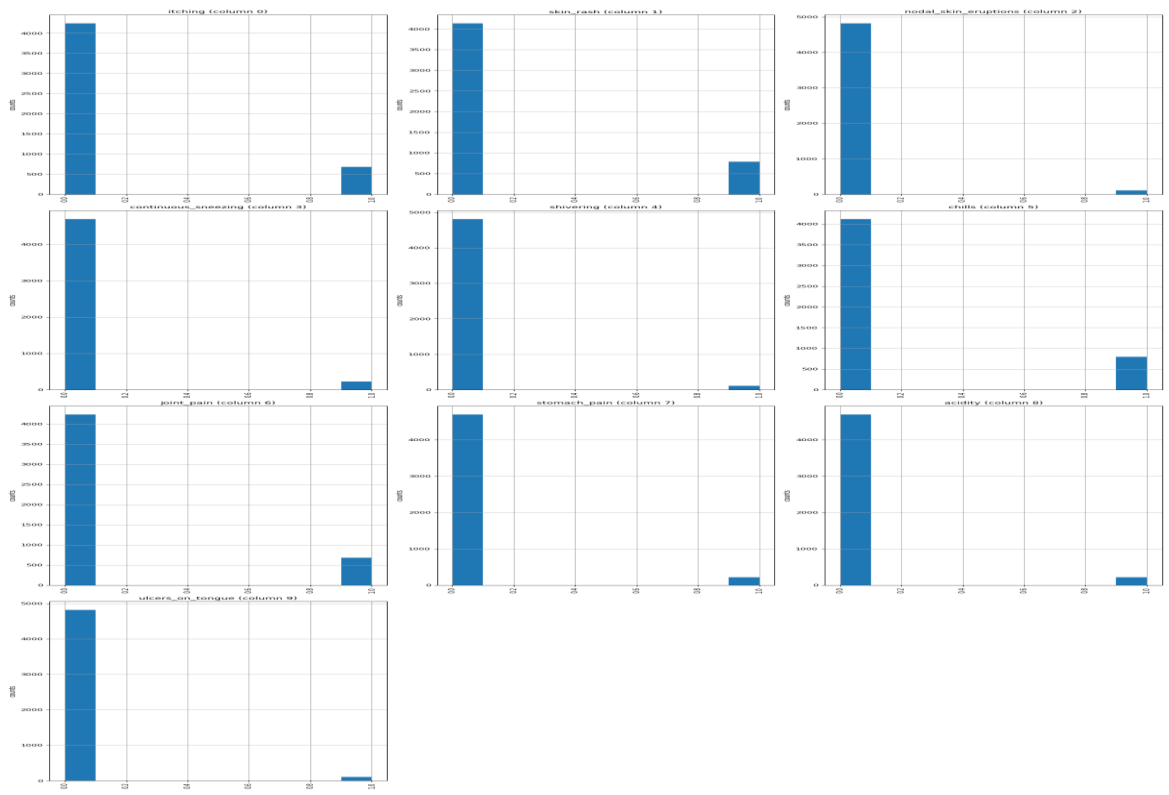
- Checked for missing values but there were no missing values.

```
# Check for missing values
print ("Missing values in each column:")
print(df.isnull(). sum ())
```



```
Missing values in each column:
itching                 0
skin_rash               0
nodal_skin_eruptions    0
continuous_sneezing     0
shivering               0
                       ..
inflammatory_nails      0
blister                 0
red_sore_around_nose    0
yellow_crust_ooze       0
prognosis               0
Length: 133, dtype: int64
```

- Histogram:



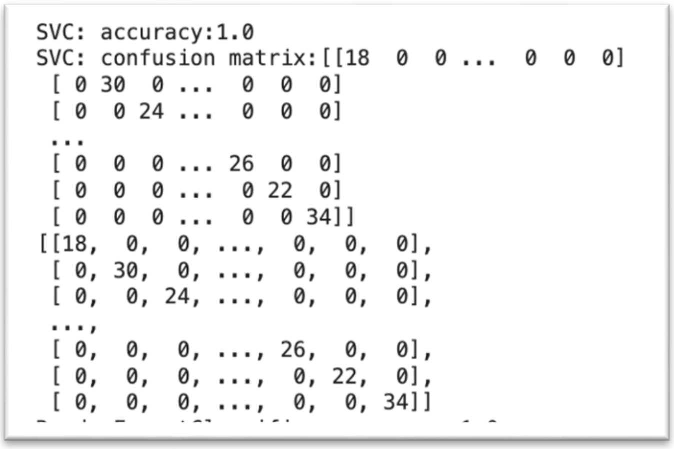
- Upon more analysis, the data is seen to be completely preprocessed.

## MODEL CONSTRUCTION:

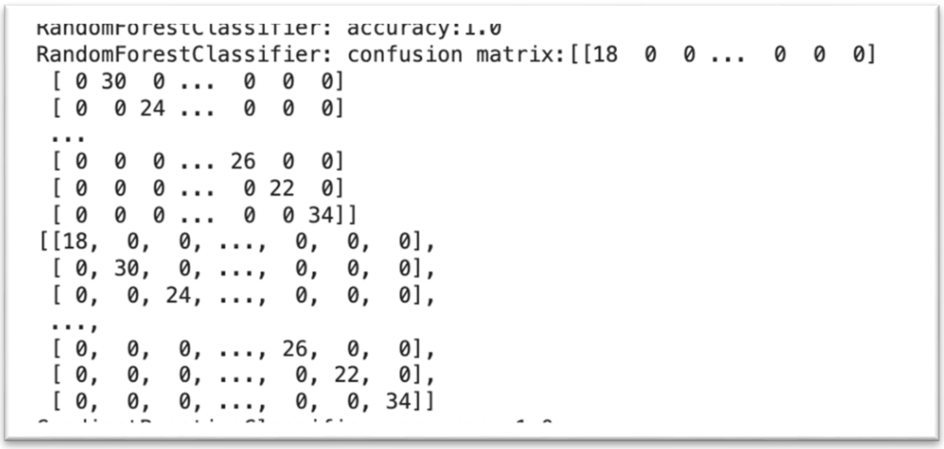
- Split the data into training and test and used label encoder to encode the prognosis column.
- Trained the models and compared their accuracy scores.

```
#creating dictionary to store models
models= {
"SVC":SVC (kernel="linear"),
"RandomForestClassifier":RandomForestClassifier(n_estimators=100, random_state=123),
"GradientBoostingClassifier":GradientBoostingClassifier(n_estimators=100, random_state=34),
"MultinomialNB":MultinomialNB(),
"KNeighborsClassifier":KNeighborsClassifier(n_neighbors=5)
}
```

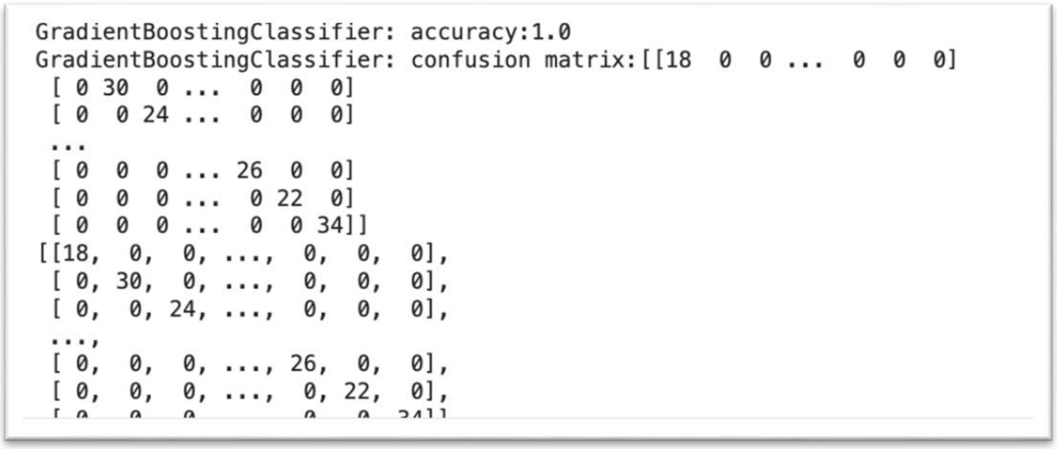
- All of the models had a good accuracy score of 1.



```
SVC: accuracy:1.0
SVC: confusion matrix:[[18  0  0 ...  0  0  0]
 [ 0 30  0 ...  0  0  0]
 [ 0  0 24 ...  0  0  0]
 ...
 [ 0  0  0 ... 26  0  0]
 [ 0  0  0 ...  0 22  0]
 [ 0  0  0 ...  0  0 34]]
[[18,  0,  0, ...,  0,  0,  0],
 [ 0, 30,  0, ...,  0,  0,  0],
 [ 0,  0, 24, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 26,  0,  0],
 [ 0,  0,  0, ...,  0, 22,  0],
 [ 0,  0,  0, ...,  0,  0, 34]]
```

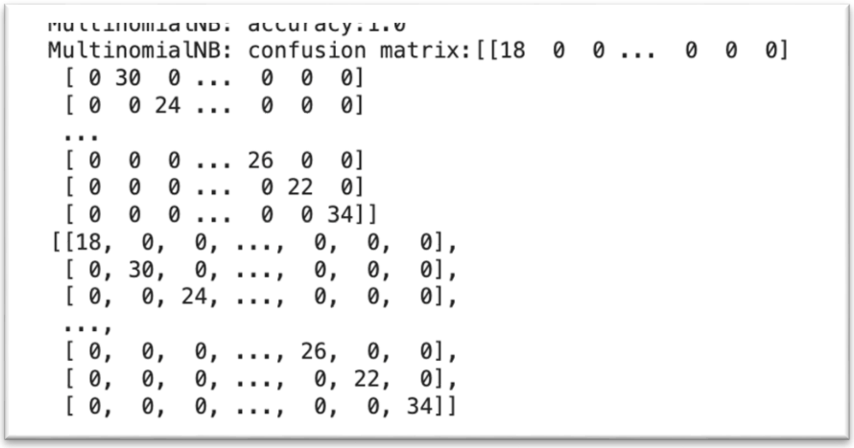


```
RandomForestClassifier: accuracy:1.0
RandomForestClassifier: confusion matrix:[[18  0  0 ...  0  0  0]
 [ 0 30  0 ...  0  0  0]
 [ 0  0 24 ...  0  0  0]
 ...
 [ 0  0  0 ... 26  0  0]
 [ 0  0  0 ...  0 22  0]
 [ 0  0  0 ...  0  0 34]]
[[18,  0,  0, ...,  0,  0,  0],
 [ 0, 30,  0, ...,  0,  0,  0],
 [ 0,  0, 24, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 26,  0,  0],
 [ 0,  0,  0, ...,  0, 22,  0],
 [ 0,  0,  0, ...,  0,  0, 34]]
```

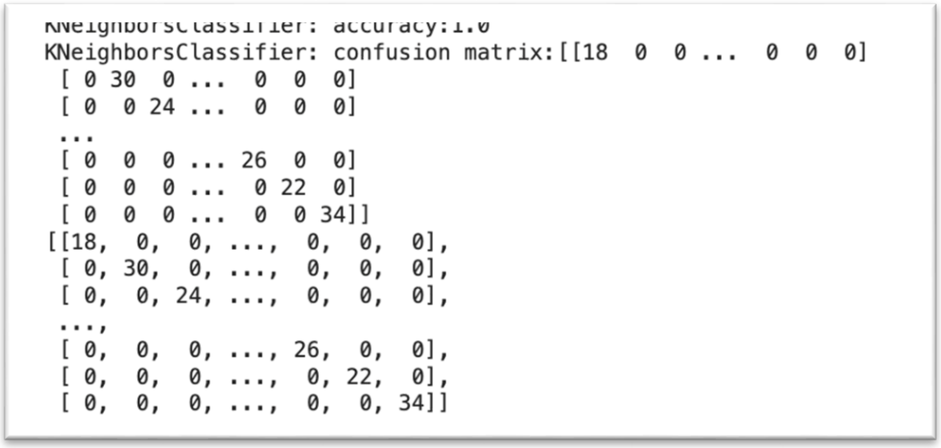


```
GradientBoostingClassifier: accuracy:1.0
GradientBoostingClassifier: confusion matrix:[[18  0  0 ...  0  0  0]
 [ 0 30  0 ...  0  0  0]
 [ 0  0 24 ...  0  0  0]
 ...
 [ 0  0  0 ... 26  0  0]
 [ 0  0  0 ...  0 22  0]
 [ 0  0  0 ...  0  0 34]]
[[18,  0,  0, ...,  0,  0,  0],
 [ 0, 30,  0, ...,  0,  0,  0],
 [ 0,  0, 24, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 26,  0,  0],
 [ 0,  0,  0, ...,  0, 22,  0],
 [ 0,  0,  0, ...,  0,  0, 34]]
```

## THE SYSTEM:



```
MultinomialNB: accuracy:1.0
MultinomialNB: confusion matrix:[[18  0  0 ...  0  0  0]
 [ 0 30  0 ...  0  0  0]
 [ 0  0 24 ...  0  0  0]
 ...
 [ 0  0  0 ... 26  0  0]
 [ 0  0  0 ...  0 22  0]
 [ 0  0  0 ...  0  0 34]]
[[18,  0,  0, ...,  0,  0,  0],
 [ 0, 30,  0, ...,  0,  0,  0],
 [ 0,  0, 24, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 26,  0,  0],
 [ 0,  0,  0, ...,  0, 22,  0],
 [ 0,  0,  0, ...,  0,  0, 34]]
```



```
KNeighborsClassifier: accuracy:1.0
KNeighborsClassifier: confusion matrix:[[18  0  0 ...  0  0  0]
 [ 0 30  0 ...  0  0  0]
 [ 0  0 24 ...  0  0  0]
 ...
 [ 0  0  0 ... 26  0  0]
 [ 0  0  0 ...  0 22  0]
 [ 0  0  0 ...  0  0 34]]
[[18,  0,  0, ...,  0,  0,  0],
 [ 0, 30,  0, ...,  0,  0,  0],
 [ 0,  0, 24, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 26,  0,  0],
 [ 0,  0,  0, ...,  0, 22,  0],
 [ 0,  0,  0, ...,  0,  0, 34]]
```

- I chose SVC as the final model for the prediction.
- This code snippet uses the **pickle** module in Python to serialize and save a machine learning model, specifically a Support Vector Classifier (SVC), to a

```
pickle.dump(svc,open("svc.pkl",'wb'))
```

- After we save the model, we can then use it to make our predictions.

## THE SYSTEM:

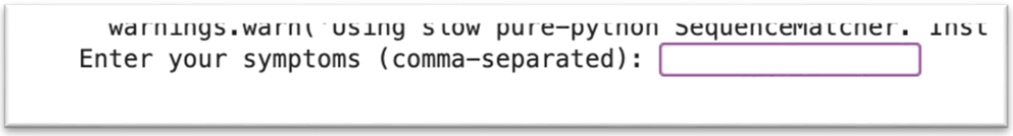
- We then call all the other datasets like diets,symptoms etc.

```
def helper(dis):
desc = description[description['Disease'] == predicted_disease]['Description']
desc = " ". join ([w for w in desc])

pre = precautions[precautions['Disease'] == dis] [['Precaution_1', 'Precaution_2', 'Precaution_3',
'Precaution_4']]
pre = [col for col in pre.values]

med = medications[medications['Disease'] == dis] ['Medication']
med = [med for med in med.values]

die = diets[diets['Disease'] == dis] ['Diet']
die = [die for die in die.values]

wrkout = workout[workout['disease'] == dis] ['workout']



return desc,pre,med,die,wrkout

symptoms_dict = {'itching': 0, 'skin_rash': 1, 'nodal_skin_eruptions': 2, 'continuous_sneezing': 3,
'shivering': 4, 'chills': 5, 'joint_pain': 6, 'stomach_pain': 7, 'acidity': 8, 'ulcers_on_tongue': 9...}
diseases_list = {15: 'Fungal infection', 4: 'Allergy', 16: 'GERD', 9: 'Chronic cholestasis', 14: 'Drug Reaction',
33: 'Peptic ulcer diseae', 1: ...}

# Model Prediction function
def get_predicted_value(patient_symptoms):
input_vector = np.zeros(len(symptoms_dict))
for item in patient_symptoms:
input_vector[symptoms_dict[item]] = 1
return diseases_list[svc.predict([input_vector]) [0]]
```

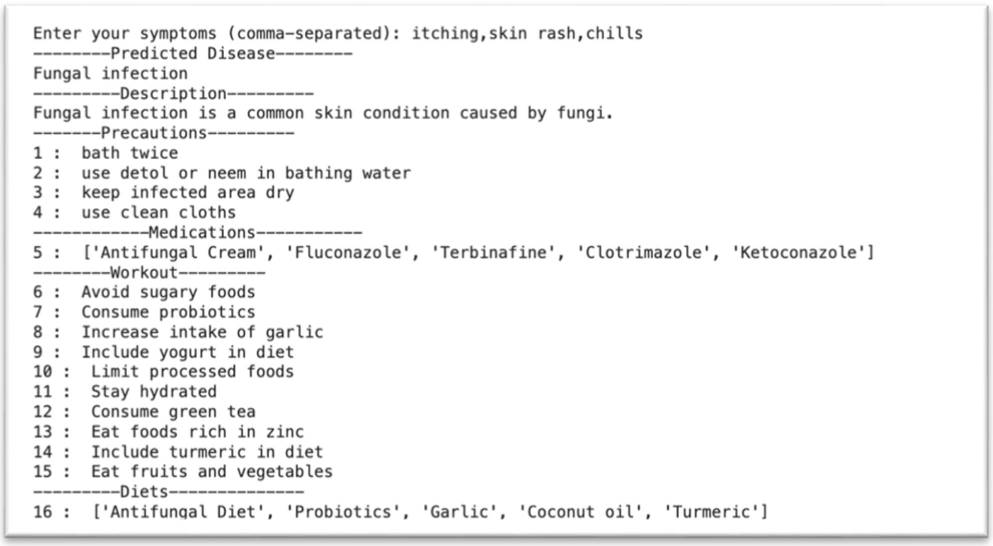
- The disease prediction system takes in a patient's symptoms as input, which are then converted into a numerical representation using a dictionary that maps symptom names to numerical indices. This numerical representation is fed into a trained Support Vector Classifier (SVC) model, which predicts a disease index based on the input symptoms. The predicted disease index is then used to look up the corresponding disease name in a dictionary, and the disease name is returned as the predicted diagnosis. This process allows the system to accurately predict a patient's disease based on their symptoms, providing a valuable tool for medical professionals and patients alike.

- The numbers to the diseases in the disease dict are given by label encoding which makes it easier for the computer to access it.
- The symptom that is taken as the input is confirmed against a predefined dictionary of valid symptoms(symptoms_dict).
- Once the symptoms are confirmed, the system processes them to predict a disease. The predicted disease is then used to retrieve more information such as a description, precautions, medications, workout routines, and diets associated with the disease. The system outputs this information to the user, providing a comprehensive diagnosis and guidance for managing their condition.

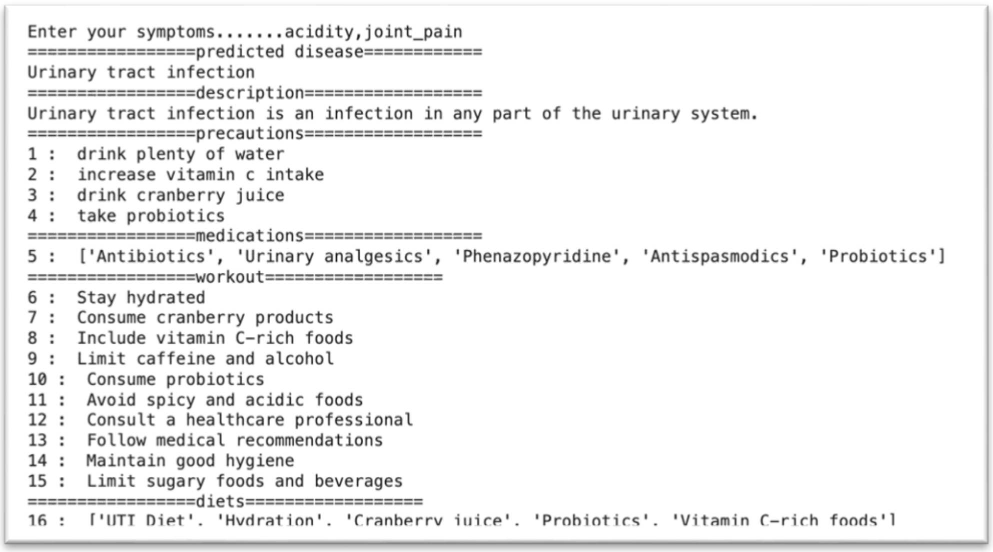
## **OUTPUT:**



```
  warnings.warn("Using slow pure-python SequenceMatcher. Inst
Enter your symptoms (comma-separated):
```



```
Enter your symptoms (comma-separated): itching,skin rash,chills
--------Predicted Disease--------
Fungal infection
---------Description---------
Fungal infection is a common skin condition caused by fungi.
-------Precautions---------
1 :  bath twice
2 :  use detol or neem in bathing water
3 :  keep infected area dry
4 :  use clean cloths
------------Medications-----------
5 :  ['Antifungal Cream', 'Fluconazole', 'Terbinafine', 'Clotrimazole', 'Ketoconazole']
--------Workout---------
6 :  Avoid sugary foods
7 :  Consume probiotics
8 :  Increase intake of garlic
9 :  Include yogurt in diet
10 :  Limit processed foods
11 :  Stay hydrated
12 :  Consume green tea
13 :  Eat foods rich in zinc
14 :  Include turmeric in diet
15 :  Eat fruits and vegetables
---------Diets--------------
16 :  ['Antifungal Diet', 'Probiotics', 'Garlic', 'Coconut oil', 'Turmeric']
```



```
Enter your symptoms.......acidity,joint_pain
=================predicted disease============
Urinary tract infection
=================description=================
Urinary tract infection is an infection in any part of the urinary system.
=================precautions==================
1 :  drink plenty of water
2 :  increase vitamin c intake
3 :  drink cranberry juice
4 :  take probiotics
=================medications==================
5 :  ['Antibiotics', 'Urinary analgesics', 'Phenazopyridine', 'Antispasmodics', 'Probiotics']
=================workout=================
6 :  Stay hydrated
7 :  Consume cranberry products
8 :  Include vitamin C-rich foods
9 :  Limit caffeine and alcohol
10 :  Consume probiotics
11 :  Avoid spicy and acidic foods
12 :  Consult a healthcare professional
13 :  Follow medical recommendations
14 :  Maintain good hygiene
15 :  Limit sugary foods and beverages
=================diets==================
16 :  ['UTI Diet', 'Hydration', 'Cranberry juice', 'Probiotics', 'Vitamin C-rich foods']
```

- Upon entering the symptoms, you get the predicted disease along with precautions and medications.

**REFERENCES:**

1.Datasets from: https://github.com/itachi9604/healthcare-chatbot

2.https://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

3. "Disease Prediction Using Machine Learning Over Big Data"Vinitha S, Sweetlin S, Vinusha H and Sajini S (2018).

4. https://www.karunya.edu/aqar/2020-21/QIF/Criteria_1/1.3.4/520.pdf

5. https://www.geeksforgeeks.org/disease-prediction-using-machine-learning/