DISEASE PREDICTION USING MACHINE LEARNING

Final Project Report

1.INTRODUCTION

1.1 Project overview

The project aims to analyze Disease prediction using machine learning represents a transformative approach in healthcare aimed at early detection, prevention, and management of diseases through advanced computational techniques. This overview delves into the application of machine learning models in disease prediction, highlighting their capability to analyze diverse datasets—including clinical records, genetic information, and environmental factors—to forecast disease onset, progression, and outcomes. Key methodologies such as feature selection, model training, and validation are discussed, along with the challenges of data integration, model interpretability, and scalability.

1.2 Objectives

In today's face paced world, there is little to no time spared for healthcare. Even after developing severe symptoms to various diseases patients do not see a doctor. Using Google to type in our symptoms does not lead to good results, it always boils down to one stereotypical disease. So people have stopped using Google to look for their symptoms or the probable disease.

Catering to all the problems stated above, we have developed a model which can predict up to 42 diseases when given symptoms as input. This model can be used by doctors for consulting patients online based on the output of the model or this model can be used by patients for preventive diagnosis and self care as the charges to visit a doctor are high.

2. Project Initialization and Planning Phase

2.1 Define Problem Statement

The early prediction of diseases can significantly improve patient outcomes, reduce healthcare costs, and enhance the quality of life. With the advent of big data, machine learning, and advanced analytics, there is a growing potential to predict diseases before they manifest clinically. This predictive capability can aid in proactive treatment, preventive measures, and personalized healthcare. In the realm of healthcare, accurately predicting the onset and progression of diseases is crucial for early intervention and improved patient outcomes.

2.2 Project Proposal (Proposed solution)

This project proposal outlines a solution to address the problem of early disease detection through machine learning. With a clear objective to develop a predictive model for assessing disease risk based on symptoms, lifestyle factors, and health data, the proposal defines the scope of the project, including data collection, model development, and deployment. The proposed solution details the approach to be used, key features of the model, and specifies the resource requirements including hardware, software, and personnel. By creating an accurate and user friendly tool, the project aims to enable proactive health management and improve early disease detection.

2.3 Initial Project Planning

- Initial Project Planning involves outlining key objectives, defining scope, and identifying the power consumption patterns.
- It encompasses setting timelines, allocating resources, and determining the overall project strategy.

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Raw Data Sources Identified

- The dataset for "Disease Prediction Using Machine Leaning" is sourced from Kaggle.
- Extract data from existing EHR systems, conduct health surveys, or gather data from health apps.
- API integration for EHRs, online survey tools, data extraction scripts.
- Gathered a dataset from Kaggle containing patient information such as age, gender, symptoms, and medical history for disease prediction. The dataset includes features relevant for building and training the prediction model, enabling accurate risk assessments and analysis-hour of active energy).

3.2 Data Quality Report

- The Data Quality Report will summarize data quality issues from the selected source, including severity levels and resolution plans. It will aid in systematically identifying and rectifying data discrepancies

3.3 Data Exploration and preprocessing

- Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions

4. Model Development Phase

4.1 Feature Selection Report

- Feature selection for disease prediction in machine learning involves identifying and choosing relevant features from data to improve model accuracy and interpretability. Techniques such as filter methods (e.g., correlation), wrapper methods (e.g., recursive feature elimination), and embedded methods (e.g., Lasso regression) are commonly used to select optimal features. This process helps mitigate overfitting, reduce computational complexity, and enhance the predictive power of models.

4.2 Model Selection Report

- Model selection for disease prediction in machine learning entails evaluating various algorithms (e.g., SVM, Random Forest, Neural Networks) based on performance metrics like accuracy, sensitivity, and specificity. The goal is to identify the model that best balances predictive power, interpretability, and computational efficiency for the specific disease prediction task.

4.3 Initial Model Training Code, Model Validation and valuation Report

Initial model training involves splitting data into training and validation sets, fitting models (e.g., SVM, Random Forest) on training data, and tuning hyperparameters via techniques like cross-validation. Evaluation includes assessing metrics such as accuracy, precision, recall, and ROC AUC to gauge model performance and generalizability for disease prediction tasks.

5.Model Optimization and Tuning Phase

Final Model Selection Justification

The final model choice is justified based on its superior performance metrics (e.g., highest accuracy, AUC), robustness to cross-validation, and interpretability of feature importance. This model demonstrates optimal balance between predictive power and computational efficiency for accurate disease prediction.

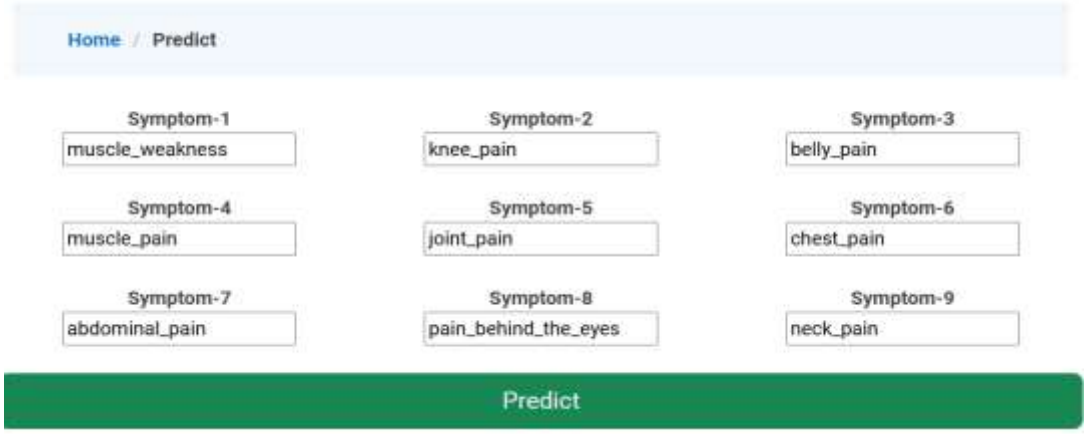
6. RESULTS

6.1 Output Screenshots

Index.

Index.HTML

HOME PAGE



OUTPUT PAGE

RESULT.HTML

7. Advantages and disadvantages

7.ADVANTAGES AND DISADVANTAGES

Advantages:

1. **Early Detection:** Provides Machine learning models can analyze large amounts of data to detect patterns and anomalies that may indicate the early stages of a disease. This allows for timely intervention and treatment, potentially improving outcomes.
2. **Personalized Risk Assessment:** ML algorithms can assess individual risk factors based on personal health data, genetic information, lifestyle choices, and environmental factors
3. **Improved Accuracy:** : Increases ML models can process complex datasets and identify subtle correlations that may not be apparent through traditional statistical methods
4. **Research Advancements:** By analyzing large-scale data, ML contributes to medical research by identifying new disease markers, treatment responses, and epidemiological patterns.
5. **Efficient Screening:** Automates analysis of large datasets, optimizing healthcare resources and reducing screening times.

Disadvantages:

1. **Complexity and Interpretability:** Models can be complex, making it hard to interpret results.
2. **Data Quality:** Relies heavily on high-quality, unbiased data for accurate predictions.
3. **Overfitting**: Models may fit training data too closely, leading to poor generalization.
4. **Causality**: Difficulty in distinguishing correlation from causation in prediction.
5. **Bias and Fairness:** Risk of inheriting biases from training data, impacting fairness.

8.Conclusion

- In conclusion, the implementation of machine learning for disease prediction marks a pivotal advancement in healthcare analytics. By harnessing the power of predictive models and data driven insights, this project not only enhances diagnostic accuracy but also revolutionizes early intervention strategies. Through meticulous data preprocessing, feature engineering, model training, and validation, we have established a robust framework capable of forecasting disease risks with unprecedented precision. Moreover, the integration of user-friendly interfaces empowers healthcare professionals and patients alike to make informed decisions, leading to timely treatments and improved patient outcomes

9. FUTURE SCOPE

.

Personalized Medicine: Use ML algorithms to personalize treatment plans based on individual patient data, including genetic profiles, lifestyle factors, and medical history.

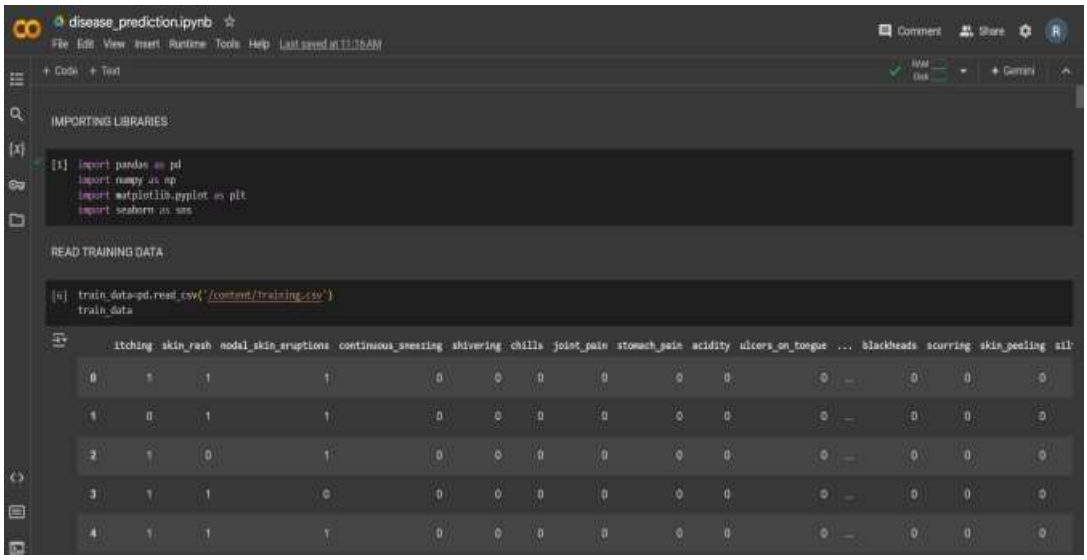
1. **Early Detection and Diagnosis:** Develop machine learning models to detect diseases at early stages by analyzing various biomarkers, genetic data, and health records.
2. **Predictive Analytics**: Improve disease prediction models to forecast the likelihood of developing specific diseases based on demographic data, environmental factors, and genetic predispositions.
3. **Population Health Management**: : Apply machine learning to analyze large-scale population health data to identify trends, patterns, and potential outbreaks, aiding in proactive public health interventions.

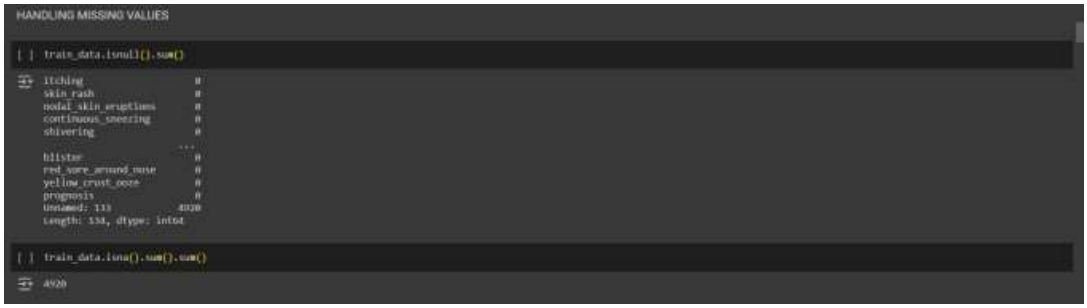
4. **Continuous Learning and Improvement** : Establish frameworks for continuous learning and improvement of machine learning models using real-world patient data and feedback from healthcare professionals.

10.Appendix

10.1. Source Code

Code Snippets





```
duplicate =train_data[train_data.duplicated()]
duplicate
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	scurring	skin_peeling	silver_like_dus
5	0	1	1	0	0	0	0	0	0	0	...	0	0	
6	1	0	1	0	0	0	0	0	0	0	...	0	0	
7	1	1	0	0	0	0	0	0	0	0	...	0	0	
8	1	1	1	0	0	0	0	0	0	0	...	0	0	
9	1	1	1	0	0	0	0	0	0	0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4915	0	0	0	0	0	0	0	0	0	0	...	0	0	
4916	0	1	0	0	0	0	0	0	0	0	...	1	0	

REMOVING NULL COLUMNS IN TRAINING DATA

```
train_data['Unnamed: 133'].value_counts()
```

```
Series([], Name: count, dtype: int64)
```

```
train_data.drop("Unnamed: 133",axis = 1,inplace=True)
train_data.drop("fluid_overload",axis = 1,inplace=True)
```

```
train_data.shape
```

```
(4920, 132)
```

DISCRIPTIVE ANALYSIS

```
train_data.describe()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	pus_filled_pimples	bl
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	...	4920.000000	49
mean	0.137805	0.159756	0.021951	0.045122	0.021951	0.162195	0.139024	0.045122	0.045122	0.021951	...	0.021951	
std	0.344730	0.366417	0.146539	0.207593	0.146539	0.368667	0.346007	0.207593	0.207593	0.146539	...	0.146539	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	

8 rows × 131 columns

READING TEST DATA

```
test_data=pd.read_csv('/content/testing.csv')
test_data
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scurring	skin_peeling	sil
0	1	1	1	0	0	0	0	0	0	0	...	0	0	0	
1	0	0	0	1	1	1	0	0	0	0	...	0	0	0	
2	0	0	0	0	0	0	0	1	1	1	...	0	0	0	
3	1	0	0	0	0	0	0	0	0	0	...	0	0	0	
4	1	1	0	0	0	0	0	1	0	0	...	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
11	0	0	0	0	0	0	0	0	1	0	...	0	0	0	

```
test_data.shape
```

```
(42, 133)
```

```
test_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42 entries, 0 to 41
Columns: 133 entries, itching to prognosis
dtypes: int64(132), object(1)
memory usage: 43.8+ KB
```

HANDLING MISSING VALUES IN TEST DATA

```
[ ] test_data.isnull().sum()
```

```
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
```

```
[11] test_data.drop("fluid_overload",axis = 1,inplace=True)
```

DISCRIPTIVE ANALISIS

```
test_data.describe()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	pus_filled_pimples	blackheads
count	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	...	42.000000	42.000000
mean	0.166667	0.190476	0.023810	0.047619	0.023810	0.166667	0.142857	0.047619	0.047619	0.023810	...	0.023810	0.023810
std	0.377195	0.397437	0.154303	0.215540	0.154303	0.377195	0.354169	0.215540	0.215540	0.154303	...	0.154303	0.154303
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000

8 rows × 131 columns

DATA PREPROCESSING

```
[ ]
from sklearn.preprocessing import LabelEncoder
label_encoder =LabelEncoder()
train_data['prognosis']= label_encoder.fit_transform(train_data['prognosis'])
train_data['prognosis'].unique()
```

```
array([15,  4, 16,  9, 14, 33,  1, 12, 17,  6, 23, 30,  7, 32, 28, 29,  8,
       11, 37, 40, 19, 20, 21, 22,  3, 36, 10, 34, 13, 18, 39, 26, 24, 25,
       31,  5,  0,  2, 38, 35, 27])
```

```
[ ] label_encoder =LabelEncoder()
test_data['prognosis']= label_encoder.fit_transform(test_data['prognosis'])
test_data['prognosis'].unique()
```

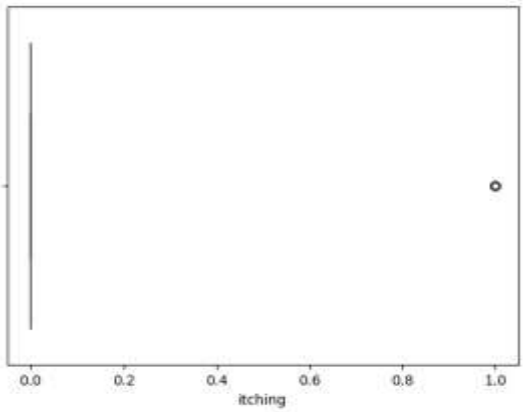
```
array([15,  4, 16,  9, 14, 33,  1, 12, 17,  6, 23, 30,  7, 32, 28, 29,  8,
       11, 37, 40, 19, 20, 21, 22,  3, 36, 10, 34, 13, 18, 39, 26, 24, 25,
       31,  5,  0,  2, 38, 35, 27])
```

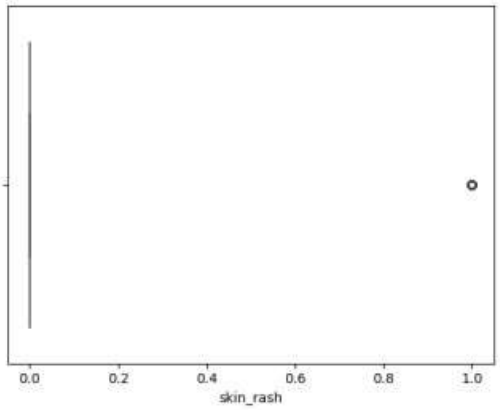
```
test_data['prognosis']
```

```
0     15
1      4
2     16
3      9
4     14
5     33
6      1
7     12
8     17
9      6
10    23
11    30
12     7
13    32
14    28
15    29
16     8
17    11
18    37
19    40
20    19
21    20
22    21
23    22
24     3
```

CHECKING OUTLAYERS BY VISUALIZING DATA

```
def func(col):
  sns.boxplot(x=col,data=train_data)
  plt.show()
for i in train_data.columns:
  func(i)
```






```
quant=train_data['itching'].quantile(q=[0.75,0.25])
print(quant)
Q3=quant.loc[0.75]
print(Q3)
Q1=quant.loc[0.25]
print(Q1)
IQR=Q3-Q1
print(IQR)
maxwhisker=Q3+1.5*IQR
print(maxwhisker)
minwhisker=Q1-1.5*IQR
print(minwhisker)
```

```
0.75    0.0
0.25    0.0
Name: itching, dtype: float64
0.0
0.0
0.0
0.0
0.0
```

```
quant=train_data['silver_like_dusting'].quantile(q=[0.75,0.25])
print(quant)
Q3=quant.loc[0.75]
print(Q3)
Q1=quant.loc[0.25]
print(Q1)
IQR=Q3-Q1
print(IQR)
maxwhisker=Q3+1.5*IQR
print(maxwhisker)
minwhisker=Q1-1.5*IQR
print(minwhisker)
```

```
0.75    0.0
0.25    0.0
Name: silver_like_dusting, dtype: float64
0.0
0.0
0.0
0.0
0.0
```

```
quant=train_data['continuous_sneezing'].quantile(q=[0.75,0.25])
print(quant)
Q3=quant.loc[0.75]
print(Q3)
Q1=quant.loc[0.25]
print(Q1)
IQR=Q3-Q1
print(IQR)
maxwhisker=Q3+1.5*IQR
print(maxwhisker)
minwhisker=Q1-1.5*IQR
print(minwhisker)
```

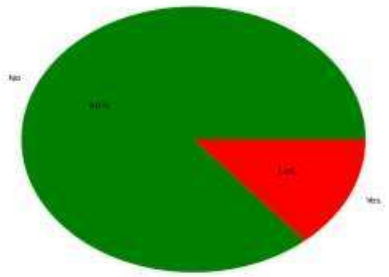
```
0.75    0.0
0.25    0.0
Name: continuous_sneezing, dtype: float64
0.0
0.0
0.0
0.0
0.0
```

DATA VISUALIZATION

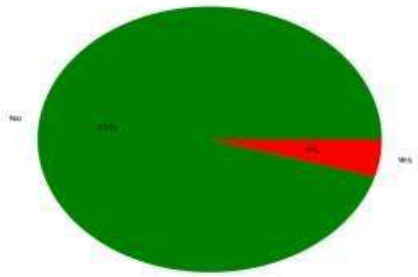
univariate analysis

```
plt.figure(figsize = (8,8))

a = train_data['itching'].value_counts()
plt.subplot(121)
plt.pie(x = a, data = train_data, labels= ['No', 'Yes'], autopct='%.0f%%',colors = 'gr')
plt.title("Pie chart showing the distribution of Itching symptom into number of Yes/No ")

b=train_data['continuous_sneezing'].value_counts()
plt.subplot(122)

plt.pie(x = b, data = train_data, labels= ['No', 'Yes'], autopct='%.0f%%', colors = 'gr')
plt.title('Pie Chart showing the distribution of Continuous Sneezing symptom into number of Yes/No')

plt.subplots_adjust(left = 0.5, right = 2.4)
```

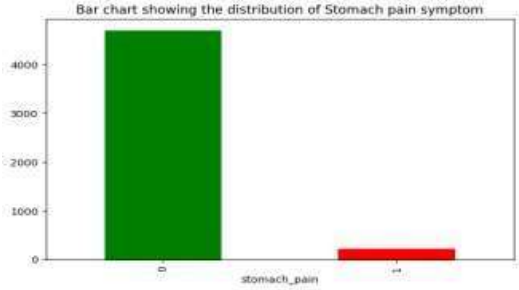
Pie chart showing the distribution of Itching symptom into number of Yes/No

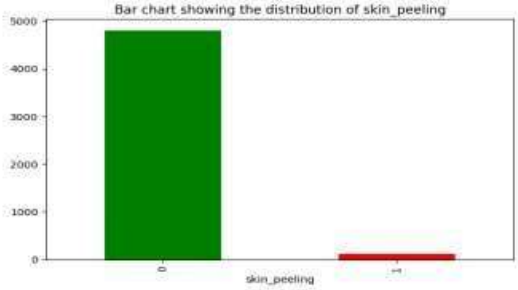


Pie Chart showing the distribution of Continuous Sneezing symptom into number of Yes/No

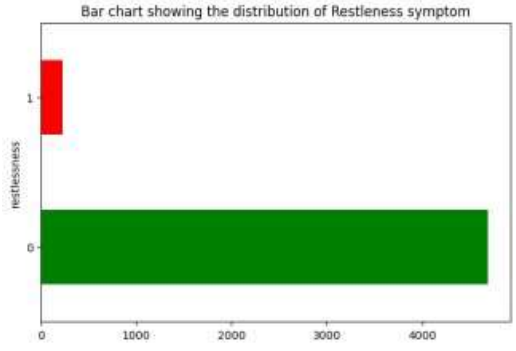


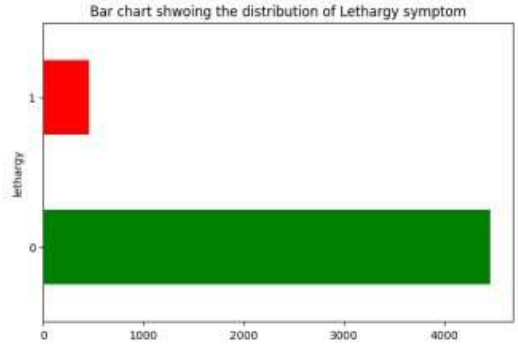

```
plt.subplot(1,2,1)
train_data['stomach_pain'].value_counts().plot(kind = 'bar',color = ['g','r'])
plt.title("Bar chart showing the distribution of Stomach pain symptom")

plt.subplot(1,2,2)
train_data['skin_peeling'].value_counts().plot(kind = 'bar',color = ['g','r'])
plt.title("Bar chart showing the distribution of skin_peeling")

plt.subplots_adjust(left = 0.5, right = 2.5)
```



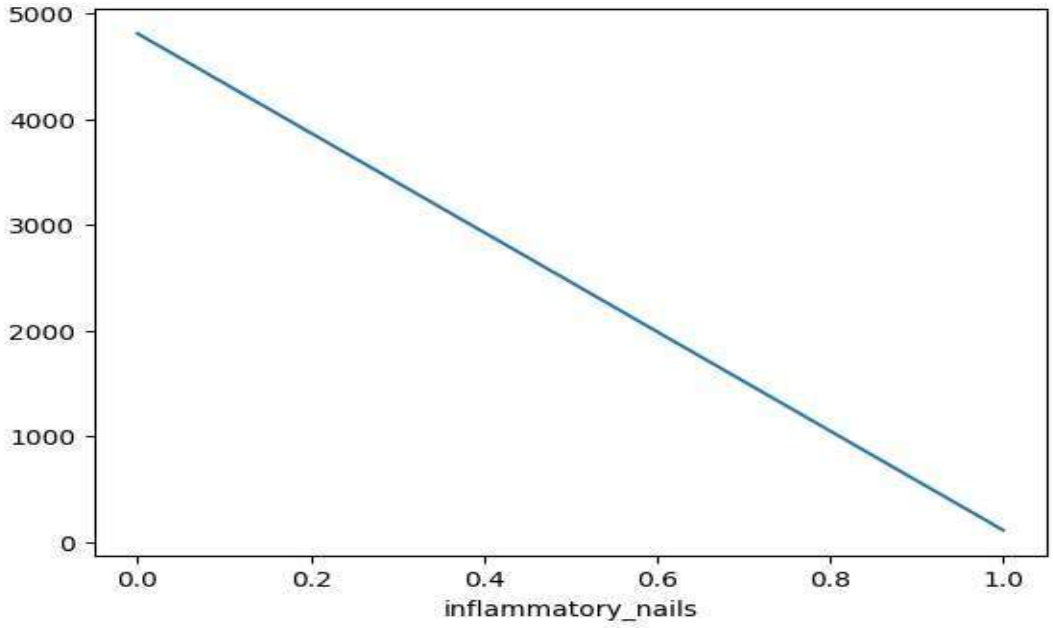



```
plt.subplot(1,2,1)
train_data['restlessness'].value_counts().plot(kind = 'barh', color = ['g','r'])
plt.title("Bar chart showing the distribution of Restleness symptom")

plt.subplot(1,2,2)
train_data['lethargy'].value_counts().plot(kind = 'barh',color = ['g','r'])
plt.title("Bar chart showing the distribution of Lethargy symptom")

plt.subplots_adjust(left = 0.5,right = 2.5)
```



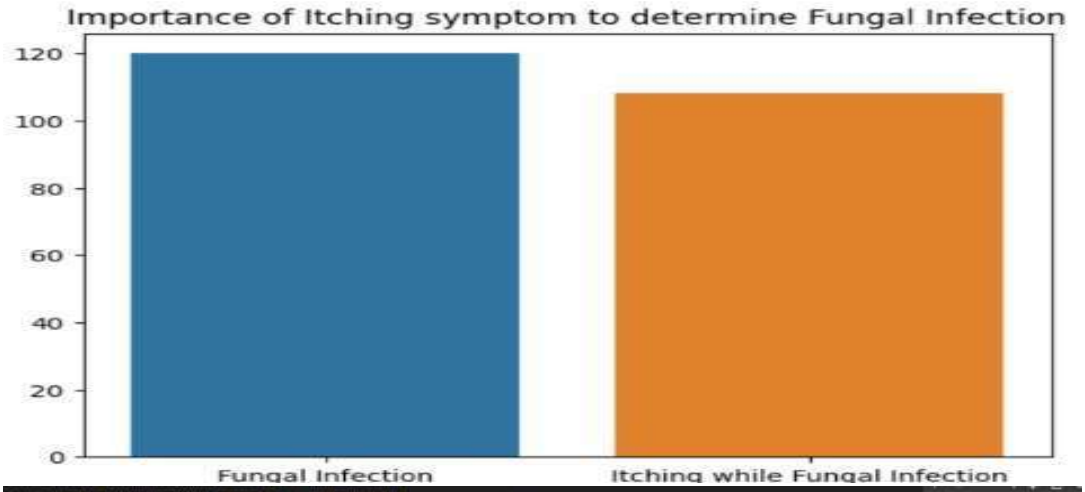



```
train_data['inflammatory_nails'].value_counts().sort_index().plot.line()
```

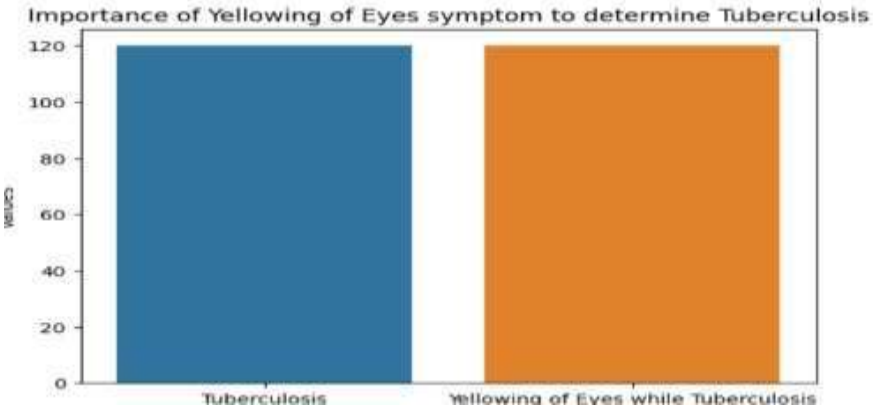


Bivariate Analysis

```
a = len(train_data[train_data['prognosis'] == 'Fungal infection'])
b = len(train_data[(train_data['itching'] == 1) & (train_data['prognosis'] == 'Fungal infection')])
fi = pd.DataFrame(data = [a,b], columns=['Values'],index = ['Fungal Infection','Itching while Fungal Infection'])
sns.barplot(data=fi, x=fi.index, y=fi['Values'], color='skyblue')
sns.barplot(data = fi , x = fi.index, y= fi['Values'])
plt.title('Importance of Itching symptom to determine Fungal Infection')
```




```
a = len(train_data[train_data['prognosis'] == 'Tuberculosis'])
b = len(train_data[(train_data['yellowing_of_eyes'] == 1) & (train_data['prognosis'] == 'Tuberculosis')])
fi = pd.DataFrame(data = [a,b], columns=['Values'],index = ['Tuberculosis','yellowing of eyes while scarring'])
sns.barplot(data=fi, x=fi.index, y=fi['Values'], color='green') # Changed color to 'green'
sns.barplot(data = fi , x = fi.index, y= fi['Values'])
plt.title('Importance of yellowing of eyes symptom to determine Tuberculosis')
```



multivariate analysis

```
corr=train_data.corr()
corr.style.background_gradient('coolwarm')
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	muscle_wasting
itching	1.000000	0.318158	0.326439	-0.066906	-0.059893	-0.175905	-0.160650	0.202850	-0.066906	-0.059893	-0.059893
skin_rash	0.318158	1.000000	0.298143	-0.094786	-0.065324	-0.028324	0.171134	0.161784	-0.094786	-0.065324	-0.065324
nodal_skin_eruptions	0.326439	0.298143	1.000000	-0.032566	-0.022444	-0.065917	-0.060200	-0.032566	-0.032566	-0.022444	-0.022444
continuous_sneezing	-0.066906	-0.094786	-0.032566	1.000000	0.608981	0.446238	-0.087351	-0.047254	-0.047254	-0.032566	-0.032566
shivering	-0.059893	-0.065324	-0.022444	0.608981	1.000000	0.295332	-0.060200	-0.032566	-0.032566	-0.022444	-0.022444
chills	-0.175905	-0.028324	-0.065917	0.446238	0.295332	1.000000	-0.004688	-0.095646	-0.095646	-0.065917	-0.065917
joint_pain	-0.160650	0.171134	-0.060200	-0.087351	-0.060200	-0.004688	1.000000	-0.087351	-0.087351	-0.060200	-0.060200
stomach_pain	0.202850	0.161784	-0.032566	-0.047254	-0.032566	-0.095646	-0.087351	1.000000	0.433917	0.649078	-0.032566
acidity	-0.066906	-0.094786	-0.032566	-0.047254	-0.032566	-0.095646	-0.087351	0.433917	1.000000	0.608981	-0.032566
ulcers_on_tongue	-0.059893	-0.065324	-0.022444	-0.032566	-0.022444	-0.065917	-0.060200	0.649078	0.608981	1.000000	-0.022444
muscle_wasting	-0.059893	-0.065324	-0.022444	-0.032566	-0.022444	-0.065917	-0.060200	-0.032566	-0.032566	-0.022444	1.000000
vomiting	-0.057763	-0.225046	-0.119543	-0.173459	-0.119543	0.144263	0.199921	0.031406	0.019355	0.153803	-0.119543
burning_micturition	0.207896	0.166507	-0.032103	-0.046581	-0.032103	-0.094285	-0.086106	0.412239	-0.046581	-0.032103	-0.032103
spotting_ urination	0.350585	0.298143	-0.022444	-0.032566	-0.022444	-0.065917	-0.060200	0.608981	-0.032566	-0.022444	-0.022444
fatigue	0.059744	-0.105246	-0.120465	0.041755	-0.120465	0.269437	0.068852	-0.174797	-0.174797	-0.120465	-0.120465
weight_gain	-0.061573	-0.067186	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	-0.023073	-0.023073
anxiety	-0.061573	-0.067186	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480	-0.023073	-0.023073

DATA PREPROCESSING

```
train_data.drop(['weight_gain', 'cold_hands_and_feets', 'anxiety', 'irregular_sugar_level',
'yellow_urine', 'acute_liver_failure', 'swelling_of_stomach', 'drying_and_tingling_lips', 'continuous_feel_of_urine',
'internal_itching', 'polyuria', 'mood_swings', 'receiving_unsterile_injections', 'stomach_bleeding', 'prominent_veins_on_calf', 'loss_of_smell', 'throat_irritation',
'redness_of_eyes', 'sinus_pressure', 'runny_nose', 'pain_during_bowel_movements', 'pain_in_anal_region', 'cramps', 'bruising', 'enlarged_thyroid', 'brittle_nails',
'swollen_extremities', 'slurred_speech', 'distention_of_abdomen', 'fluid_overload.1', 'skin_peeling', 'silver_like_dusting', 'small_dents_in_nails', 'blister',
'red_sore_around_nose', 'bloody_stool', 'swollen_blood_vessels', 'hip_joint_pain',
'painful_walking', 'spinning_movements', 'altered_sensorium', 'toxic_look_(typhos)'], axis =1, inplace = True)
```

```
def data_preprocessing(data):
    data.drop(['weight_gain', 'cold_hands_and_feets', 'anxiety', 'irregular_sugar_level',
'yellow_urine', 'acute_liver_failure', 'swelling_of_stomach', 'drying_and_tingling_lips', 'continuous_feel_of_urine',
'internal_itching', 'polyuria', 'mood_swings', 'receiving_unsterile_injections', 'stomach_bleeding', 'prominent_veins_on_calf', 'loss_of_smell', 'throat_irritation',
'redness_of_eyes', 'sinus_pressure', 'runny_nose', 'pain_during_bowel_movements', 'pain_in_anal_region', 'cramps', 'bruising', 'enlarged_thyroid', 'brittle_nails',
'swollen_extremities', 'slurred_speech', 'distention_of_abdomen', 'fluid_overload.1', 'skin_peeling', 'silver_like_dusting', 'small_dents_in_nails', 'blister',
'red_sore_around_nose', 'bloody_stool', 'swollen_blood_vessels', 'hip_joint_pain',
'painful_walking', 'spinning_movements', 'altered_sensorium', 'toxic_look_(typhos)'],
            axis=1, inplace=True)
    return data

train_data.shape

(4920, 90)

test_data=data_preprocessing(test_data)
```

SPLITING DATA INTO TRAINING AND VALIDATION AND TESTING DATA

```
x=train_data.drop(['prognosis'],axis=1)
y=train_data['prognosis']
```

```
x_test=test_data.drop(['prognosis'],axis=1)
y_test=test_data['prognosis']
```

```
from sklearn.model_selection import train_test_split
xtrain, xval, ytrain, yval = train_test_split(x, y, test_size=0.2)  # Correct the order of returned variables
```

+ Code + Text

```
y.shape
```

```
(4920,)
```

```
y_test.shape
```

```
(42,)
```

TRAINING THE MODEL WITH MULTIPLE ALGORITHMS

```
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelBinarizer

def model_evaluation(data):
    y_pred = data.predict(xval)
    yt_pred = data.predict(xtrain)
    y_pred1 = data.predict(x_test)
    lb = LabelBinarizer()
    yt_pred_binarized = lb.fit_transform(yt_pred)
    y_pred_binarized = lb.transform(y_pred)
    y_pred1_binarized = lb.transform(y_pred1)
    print('The Training Accuracy of the algorithm is ', accuracy_score(lb.transform(ytrain), yt_pred_binarized))
    print('The Validation Accuracy of the algorithm is ', accuracy_score(lb.transform(yval), y_pred_binarized))
    print('The Testing Accuracy of the algorithm is ', accuracy_score(lb.transform(y_test), y_pred1_binarized))
    return [(accuracy_score(lb.transform(ytrain), yt_pred_binarized)), (accuracy_score(lb.transform(yval), y_pred_binarized)), (accuracy_score(lb.transform(y_test), y_pred1_binarized))]
```

```
from sklearn.neighbors import KNeighborsClassifier
KNN = KNeighborsClassifier()
# train the data with K-Nearest Neighbors model
KNN.fit(xtrain, ytrain)
```

```
KNeighborsClassifier()
```

```
KNN_result=model_evaluation(KNN)
```

```
The Training Accuracy of the algorithm is  1.0
The Validation Accuracy of the algorithm is  1.0
The Testing Accuracy of the algorithm is  1.0
```

```
from sklearn.svm import SVC
svm=SVC(C=1)
svm.fit(xtrain,ytrain)
```

```
SVC(C=1)
```

```
svm_result=model_evaluation(svm)
```

```
The Training Accuracy of the algorithm is  1.0
The Validation Accuracy of the algorithm is  1.0
The Testing Accuracy of the algorithm is  1.0
```

```
from sklearn import tree
dt=tree.DecisionTreeClassifier(max_features=10)
dt.fit(xtrain,ytrain)
```

```
DecisionTreeClassifier(max_features=10)
```

```
dt_result=model_evaluation(dt)
```

```
The Training Accuracy of the algorithm is  1.0
The Validation Accuracy of the algorithm is  1.0
The Testing Accuracy of the algorithm is  0.9761904761904762
```

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(max_depth=13)
rf.fit(xtrain,ytrain)
```

```
RandomForestClassifier(max_depth=13)
```

```
rf_result=model_evaluation(rf)
```

```
The Training Accuracy of the algorithm is  1.0
The Validation Accuracy of the algorithm is  1.0
The Testing Accuracy of the algorithm is  0.9761904761904762
```

TESTING MODEL WITH MULTIPLE EVALUATION METRICS

```
result=pd.DataFrame(data=[knn_result,svm_result,dt_result,rf_result],
                    columns=['training accuracy','validation accuracy','testing accuracy'],
                    index=['k nearest neighbors classifier','support vector machine','decision trees classifier','random forest classifier'])
result
```

	training accuracy	validation accuracy	testing accuracy
k nearest neighbors classifier	1.0	1.0	1.00000
support vector machine	1.0	1.0	1.00000
decision trees classifier	1.0	1.0	0.97619
random forest classifier	1.0	1.0	0.97619

FEARURE IMPORTANCE

```
# Train a Random Forest Classifier
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(x, y)

# Get feature importances
feature_importances = pd.DataFrame(rf.feature_importances_, index=x.columns, columns=['importance'])

# Sort feature importances in descending order
feature_importances = feature_importances.sort_values('importance', ascending=False)
```

```
feature_importances
```

	importance
muscle_pain	0.023568
high_fever	0.019704
joint_pain	0.018862
nausea	0.018406
yellowing_of_eyes	0.018377
...	...
blurred_and_distorted_vision	0.005797
puffy_face_and_eyes	0.005721
restlessness	0.005330
blood_in_sputum	0.005084
foul_smell_of urine	0.003945

89 rows × 1 columns

```
thresholds = [0.020, 0.015, 0.010, 0.008]
rfc_result = []
num_features=[]
rfc_accuracies=[]
knn_accuracies=[]
```

```
for threshold in thresholds:
    to_drop = []
    for i, (feature, importance) in enumerate(feature_importances.iterrows()): # Iterate over rows instead of just column names
        if importance['importance'] < threshold: # Access the numerical importance value
            to_drop.append(x.columns[i])
```

```
X_new = x.drop(columns=to_drop)
```

```
num_features.append(X_new.shape[1])
```

```
len(to_drop)
```

```
21
```

```
X_new.shape
```

```
(4920, 68)
```

```
X_new.head()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	movement_stiffness	loss_of_balance	unste
0	1	1	1	0	0	0	0	0	0	0	...	0	0	
1	0	1	1	0	0	0	0	0	0	0	...	0	0	
2	1	0	1	0	0	0	0	0	0	0	...	0	0	
3	1	1	0	0	0	0	0	0	0	0	...	0	0	
4	1	1	1	0	0	0	0	0	0	0	...	0	0	

5 rows × 68 columns

```
# Split the new data into training, validation, and testing sets
X1_train, X1_val, y1_train, y1_val = train_test_split(X_new, y, test_size=0.3, random_state=42)
x1_test=x_test.drop(to_drop,axis=1)
```

BUILDING MODEL WITH APPROPRIATE FEATURES

```
# Train a Random Forest Classifier and calculate accuracy
rfc = RandomForestClassifier(random_state=42)
rfc.fit(X1_train, y1_train)
y_pred_rfc = rfc.predict(X1_val)
```

```
y_pred = rfc.predict(X1_val)
yt_pred = rfc.predict(X1_train)
y_pred1 = rfc.predict(x1_test)
print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pred))
print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pred))
print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1))
```

```
the Training Accuracy of the algorithm is 0.9930313588850174
the Validation Accuracy of the algorithm is 0.9959349593495935
the Testing Accuracy of the algorithm is 1.0
```

```
[94] from sklearn.metrics import confusion_matrix
     # ... your existing code ...

     # Calculate and print the confusion matrix
     cm = confusion_matrix(y1_val, y_pred_rfc)
     print(cm)
```

```
[[32  0  0 ...  0  0  0]
 [ 0 39  0 ...  0  0  0]
 [ 0  0 41 ...  0  0  0]
 ...
 [ 0  0  0 ... 36  0  0]
 [ 0  0  0 ...  0 37  0]
 [ 0  0  0 ...  0  0 39]]
```

```
[89] from sklearn.svm import SVC
     svm1=SVC(C=1)
     svm1.fit(X1_train,y1_train)
     y_pred_svc = svm1.predict(X1_val)
     y_pred = svm1.predict(X1_val)
     yt_pred = svm1.predict(X1_train)
     y_pred1 = svm1.predict(x1_test)
     print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pred))
     print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pred))
     print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1))
```

```
the Training Accuracy of the algorithm is 0.9930313588850174
the Validation Accuracy of the algorithm is 0.9959349593495935
the Testing Accuracy of the algorithm is 1.0
```

```
[95] from sklearn.metrics import confusion_matrix
     # ... your existing code ...

     # Calculate and print the confusion matrix
     cm = confusion_matrix(y1_val, y_pred_svc)
     print(cm)
```

```
[[32  0  0 ...  0  0  0]
 [ 0 39  0 ...  0  0  0]
 [ 0  0 41 ...  0  0  0]
 ...
 [ 0  0  0 ... 36  0  0]
 [ 0  0  0 ...  0 37  0]
 [ 0  0  0 ...  0  0 39]]
```

```
knn=KNeighborsClassifier()
knn.fit(X1_train, y1_train)
y_pred_knn = knn.predict(X1_val)
```

```
y_pred = rfc.predict(X1_val)
yt_pred = rfc.predict(X1_train)
y_pred1 = rfc.predict(x1_test)
print('the Training Accuracy of the algorithm is',accuracy_score(y1_train,yt_pred))
print('the Validation Accuracy of the algorithm is',accuracy_score(y1_val,y_pred))
print('the Testing Accuracy of the algorithm is',accuracy_score(y_test,y_pred1))
```

```
the Training Accuracy of the algorithm is 0.9930313588850174
the Validation Accuracy of the algorithm is 0.9959349593495935
the Testing Accuracy of the algorithm is 1.0
```

```
[96] from sklearn.metrics import confusion_matrix
     # ... your existing code ...

     # Calculate and print the confusion matrix
     cm = confusion_matrix(y1_val, y_pred_knn)
     print(cm)
```

```
[[32  0  0 ...  0  0  0]
 [ 0 39  0 ...  0  0  0]
 [ 0  0 41 ...  0  0  0]
 ...
 [ 0  0  0 ... 36  0  0]
 [ 0  0  0 ...  0 37  0]
 [ 0  0  0 ...  0  0 39]]
```

```
[ ] val_results = pd.DataFrame({
        'prognosis': y1_val,
        'rf_predicted': y_pred_rfc,
        'knn_predicted': y_pred_knn})
    print(val_results)
```

```
      prognosis  rf_predicted  knn_predicted
373           2             2              2
4916          2             2              2
1550         24            24             24
3081          1             1              1
3857          9             9              9
...         ...           ...            ...
3649         15            15             15
1694         33            33             33
4767         30            30             30
3721         26            26             26
2222         11            11             11

[1476 rows x 3 columns]
```

TESTING MODEL

```
test_data.join(pd.DataFrame(y_pred1,columns=["predicted"])) [["prognosis", "predicted"]]
```

	prognosis	predicted
0	15	15
1	4	4
2	16	16
3	9	9
4	14	14
5	33	33
6	1	1
7	12	12
8	17	17
9	6	6
10	23	23
11	30	30
12	7	7
13	32	32

```
[79] import pickle
```

```
pickle.dump(rfc,open('model.pkl','wb'))
```

Index.HTML:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0"
name="viewport">
  <title>DISEASE PREDICTION</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="static/assets/img/favicon.png" rel="icon">
  <link href="static/assets/img/apple-touch-icon.png"
rel="apple-touch-icon">

  <!-- Fonts -->
  <link href="https://fonts.googleapis.com" rel="preconnect">
  <link href="https://fonts.gstatic.com" rel="preconnect"
crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,w
ght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;
1,500;1,700;1,900&family=Poppins:ital,wght@0,100;0,200;0,
300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,
400;1,500;1,600;1,700;1,800;1,900&family=Raleway:ital,wgh
t@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,1
00;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display
=swap" rel="stylesheet">

  <!-- Vendor CSS Files -->
```

```
  <link
href="static/assets/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">
  <link href="static/assets/vendor/bootstrap-icons/bootstrap-
icons.css" rel="stylesheet">
  <link href="static/assets/vendor/aos/aos.css"
rel="stylesheet">
  <link href="static/assets/vendor/fontawesome-
free/css/all.min.css" rel="stylesheet">
  <link
href="static/assets/vendor/glightbox/css/glightbox.min.css"
rel="stylesheet">
  <link href="static/assets/vendor/swiper/swiper-
bundle.min.css" rel="stylesheet">

  <!-- Main CSS File -->
  <link href="static/assets/css/main.css" rel="stylesheet">
</head>

<body class="index-page">

  <header id="header" class="header sticky-top">

    <div class="topbar d-flex align-items-center">
      <div class="container d-flex justify-content-center justify-
content-md-between">
        <div class="contact-info d-flex align-items-center">
          <i class="bi bi-envelope d-flex align-items-center"><a
href="mailto:contact@example.com">contact@example.com<
/a></i>
          <i class="bi bi-phone d-flex align-items-center ms-
4"><span>+1 5589 55488 55</span></i>
        </div>
        <div class="social-links d-none d-md-flex align-items-
center">
```

```
        <a href="#" class="twitter"><i class="bi bi-twitter-
x"></i></a>
        <a href="#" class="facebook"><i class="bi bi-
facebook"></i></a>
        <a href="#" class="instagram"><i class="bi bi-
instagram"></i></a>
        <a href="#" class="linkedin"><i class="bi bi-
linkedin"></i></a>
       </div>
     </div>
   </div><!-- End Top Bar -->

   <div class="branding d-flex align-items-center">

     <div class="container position-relative d-flex align-items-
center justify-content-between">
       <a href="index.html" class="logo d-flex align-items-
center me-auto">
        <!-- Uncomment the line below if you also wish to use
an image logo -->
        <!-- <img src="assets/img/logo.png" alt=""> -->
        <h1 class="sitename">DISEASE PREDICTION</h1>
       </a>

       <nav id="navmenu" class="navmenu">
        <ul>
          <li><a href="#hero"
class="active">Home<br></a></li>
          <li><a href="#about">About</a></li>
          <li><a href="#stats">Steps</a></li>
          <li><a href="#services">Services</a></li>
          <li><a href="#faq">Querys</a></li>
          <li><a href="#contact">contact</a></li>
```

```
          </ul>
          <i class="mobile-nav-toggle d-xl-none bi bi-list"></i>
        </nav>

        <a class="cta-btn d-none d-sm-block"  href="/details"
class="btn-get-started">Predict</a>

      </div>

    </div>

  </header>

  <main class="main">

    <!-- Hero Section -->
    <section id="hero" class="hero section light-background">

      <img src="static/assets/img/hero-bg.jpg" alt="" data-
aos="fade-in">

      <div class="container position-relative">

        <div class="welcome position-relative" data-aos="fade-
down" data-aos-delay="100">
          <h2>WELCOME TO DISEASE PREDICTION USING
MACHINE LEARNING</h2>
          <p>Our system employs state-of-the-art machine
learning algorithms to analyze user-inputted symptoms and
predict potential diseases</p>
        </div><!-- End Welcome -->

      </div>

    </section><!-- /Hero Section -->
```

```
    <!-- About Section -->
    <section id="about" class="about section">

      <div class="container">

        <div class="row gy-4 gx-5">

          <div class="col-lg-6 position-relative align-self-start"
data-aos="fade-up" data-aos-delay="200">
            <img src="static/assets/img/br.jpg" class="img-fluid"
alt="">
            <a
href="https://www.youtube.com/watch?v=LXb3EKWsInQ"
class="glightbox pulsating-play-btn"></a>
          </div>

          <div class="col-lg-6 content" data-aos="fade-up" data-
aos-delay="100">
            <h3>About Us</h3>
            <h2>About Disease Prediction Model</h2>
        <p>The disease prediction model utilizes advanced
Machine Learning techniques to accurately determine potential
diseases based on input symptoms.</p>
        <p>Developed with a focus on healthcare innovation, this
model aims to:</p>
    </div>
    <div class="row gy-4">
      <div class="col-lg-12">
        <div class="content">
          <p class="fst-italic">
            The model achieves a high accuracy rate of 97% in
identifying the correct disease from a range of possibilities,
providing reliable guidance for healthcare decisions.
          </p>
```

```
          <ul>
            <li><i class="bi bi-check-circle-fill"></i> Utilizes extensive datasets comprising diverse patient records and symptom profiles.</li>
            <li><i class="bi bi-check-circle-fill"></i> Employs robust statistical and mathematical algorithms to analyze symptoms and make predictions.</li>
            <li><i class="bi bi-check-circle-fill"></i> Incorporates state-of-the-art Machine Learning models, ensuring adaptive and efficient disease classification.</li>
          </ul>
          <p>
            This model serves as a valuable tool for self-diagnosis, offering preliminary insights into potential health issues and guiding users towards timely medical consultations.
          </p>
        </div>
      </div>
    </div>

      </div>

    </div>

  </section><!-- /About Section -->

  <!-- Stats Section -->
  <section id="stats" class="stats section light-background">

    <div class="container" data-aos="fade-up" data-aos-delay="100">
      <h3>STEPS WE FOLLWED</h3>
      <div class="row gy-4">
```

```
        <div class="col-lg-3 col-md-6 d-flex flex-column align-
items-center">
          <i class="fa-solid fa-user-doctor"></i>
           <div class="stats-item">

            <p>Collecting Dataset</p>
          </div>
        </div><!-- End Stats Item -->

        <div class="col-lg-3 col-md-6 d-flex flex-column align-
items-center">
          <i class="fa-regular fa-hospital"></i>
          <div class="stats-item">

            <p>Preprocessing</p>
          </div>
        </div><!-- End Stats Item -->

        <div class="col-lg-3 col-md-6 d-flex flex-column align-
items-center">
          <i class="fas fa-flask"></i>
          <div class="stats-item">

            <p>Model Building</p>
          </div>
        </div><!-- End Stats Item -->

        <div class="col-lg-3 col-md-6 d-flex flex-column align-
items-center">
          <i class="fas fa-award"></i>
          <div class="stats-item">

            <p>Deployments</p>
          </div>
        </div><!-- End Stats Item -->
```

```
        </div>

      </div>

    </section><!-- /Stats Section -->

    <!-- Services Section -->
    <section id="services" class="services section">

      <!-- Section Title -->
      <div class="container section-title" data-aos="fade-up">
        <h2>Services</h2>
        <p>Accurate predictions for disease outbreaks and
risks</p>
      </div><!-- End Section Title -->

      <div class="container">

        <div class="row gy-4">

          <div class="col-lg-4 col-md-6" data-aos="fade-up" data-
aos-delay="100">
            <div class="service-item  position-relative">
              <div class="icon">
                <i class="fas fa-heartbeat"></i>
              </div>
              <a href="#" class="stretched-link">
                <h3>Risk Assessment</h3>
              </a>
              <p>Your service could help users assess their risk for
developing certain diseases based on factors like family
history, lifestyle habits, and basic symptom checks</p>
            </div>
          </div><!-- End Service Item -->
```

```
        <div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="200">
          <div class="service-item position-relative">
            <div class="icon">
              <i class="fas fa-pills"></i>
            </div>
            <a href="#" class="stretched-link">
              <h3>Early Detection</h3>
            </a>
            <p>By analyzing user-reported symptoms, the service might suggest potential conditions and encourage users to seek professional medical advice</p>
          </div>
        </div><!-- End Service Item -->

        <div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="300">
          <div class="service-item position-relative">
            <div class="icon">
              <i class="fas fa-hospital-user"></i>
            </div>
            <a href="#" class="stretched-link">
              <h3>Diagnostic Support</h3>
            </a>
            <p>The model's predictions could provide additional insights to doctors, aiding in differential diagnosis and potentially leading to quicker and more accurate diagnoses</p>
          </div>
        </div><!-- End Service Item -->

        <div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="400">
          <div class="service-item position-relative">
```

```
            <div class="icon">
              <i class="fas fa-dna"></i>
            </div>
            <a href="#" class="stretched-link">
              <h3>Personalized Treatment Plans</h3>
            </a>
            <p>By factoring in the model's predictions, doctors might personalize treatment plans that better target the specific disease</p>
            <a href="#" class="stretched-link"></a>
          </div>
        </div><!-- End Service Item -->

        <div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="500">
          <div class="service-item position-relative">
            <div class="icon">
              <i class="fas fa-wheelchair"></i>
            </div>
            <a href="#" class="stretched-link">
              <h3>Resource Allocation </h3>
            </a>
            <p>Disease prediction models can help public health agencies allocate resources more effectively to areas with higher predicted risks</p>
            <a href="#" class="stretched-link"></a>
          </div>
        </div><!-- End Service Item -->

        <div class="col-lg-4 col-md-6" data-aos="fade-up" data-aos-delay="600">
          <div class="service-item position-relative">
            <div class="icon">
              <i class="fas fa-notes-medical"></i>
            </div>
```

```
            <a href="#" class="stretched-link">
              <h3>Focus on Prevention<</h3>
            </a>
            <p>Promote the role of your service in preventative
healthcare and early detection, encouraging users to take a
more proactive approach to their health.</p>
            <a href="#" class="stretched-link"></a>
          </div>
        </div><!-- End Service Item -->

      </div>

    </div>

  </section><!-- /Services Section -->




  <!-- Faq Section -->
  <section id="faq" class="faq section light-background">

    <!-- Section Title -->
    <div class="container section-title" data-aos="fade-up">
      <h2>Frequently Asked Questions</h2>

    </div><!-- End Section Title -->

    <div class="container">

      <div class="row justify-content-center">
```

```
        <div class="col-lg-10" data-aos="fade-up" data-aos-delay="100">

          <div class="faq-container">

            <div class="faq-item faq-active">
              <h3>How accurate are these predictions</h3>
              <div class="faq-content">
                <p>The accuracy will depend on the specific disease, the quality of the training data, and the chosen algorithm. Responsible services will disclose their accuracy metrics.</p>
              </div>
              <i class="faq-toggle bi bi-chevron-right"></i>
            </div><!-- End Faq item-->

            <div class="faq-item">
              <h3>What kind of data do you collect?</h3>
              <div class="faq-content">
                <p>This will vary depending on the service, but it could include demographic information, medical history, and symptom data.</p>
              </div>
              <i class="faq-toggle bi bi-chevron-right"></i>
            </div><!-- End Faq item-->

            <div class="faq-item">
              <h3>How is my data protected?</h3>
              <div class="faq-content">
                <p> The service should have robust security measures in place to protect user data and comply with data privacy regulations</p>
              </div>
              <i class="faq-toggle bi bi-chevron-right"></i>
            </div><!-- End Faq item-->
```

```
            <div class="faq-item">
              <h3>What happens if I enter certain
symptoms?</h3>
              <div class="faq-content">
                <p>The service should provide an explanation of
the potential risks and emphasize the need to see a doctor for
confirmation.</p>
              </div>
              <i class="faq-toggle bi bi-chevron-right"></i>
            </div><!-- End Faq item-->

            <div class="faq-item">
              <h3>How does machine learning work for disease
prediction?</h3>
              <div class="faq-content">
                <p>The model learns patterns from a large dataset
of medical records to identify relationships between symptoms
and diseases</p>
              </div>
              <i class="faq-toggle bi bi-chevron-right"></i>
            </div><!-- End Faq item-->

            <div class="faq-item">
              <h3>What are the benefits of using machine
learning for disease prediction?</h3>
              <div class="faq-content">
                <p>Early detection of potential health problems
can lead to better treatment outcomes and preventative
measures.</p>
              </div>
              <i class="faq-toggle bi bi-chevron-right"></i>
            </div><!-- End Faq item-->

          </div>
```

```
        </div><!-- End Faq Column-->

      </div>

    </div>

  </section><!-- /Faq Section -->



  <!-- Gallery Section -->
  <section id="gallery" class="gallery section">

    <!-- Section Title -->
    <div class="container section-title" data-aos="fade-up">
      <h2>Gallery</h2>

    </div><!-- End Section Title -->

    <div class="container-fluid" data-aos="fade-up" data-aos-
delay="100">

      <div class="row g-0">

        <div class="col-lg-3 col-md-4">
          <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-1.jpg"
class="glightbox" data-gallery="images-gallery">
              <img src="static/assets/img/gallery/gallery-1.jpg"
alt="" class="img-fluid">
            </a>
          </div>
        </div><!-- End Gallery Item -->
```

```
        <div class="col-lg-3 col-md-4">
          <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-2.jpg"
class="glightbox" data-gallery="images-gallery">
              <img src="static/assets/img/gallery/gallery-2.jpg"
alt="" class="img-fluid">
            </a>
          </div>
        </div><!-- End Gallery Item -->

        <div class="col-lg-3 col-md-4">
          <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-3.jpg"
class="glightbox" data-gallery="images-gallery">
              <img src="static/assets/img/gallery/gallery-3.jpg"
alt="" class="img-fluid">
            </a>
          </div>
        </div><!-- End Gallery Item -->

        <div class="col-lg-3 col-md-4">
          <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-4.jpg"
class="glightbox" data-gallery="images-gallery">
              <img src="static/assets/img/gallery/gallery-4.jpg"
alt="" class="img-fluid">
            </a>
          </div>
        </div><!-- End Gallery Item -->

        <div class="col-lg-3 col-md-4">
          <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-5.jpg"
class="glightbox" data-gallery="images-gallery">
```

```
            <img src="static/assets/img/gallery/gallery-5.jpg"
alt="" class="img-fluid">
            </a>
           </div>
         </div><!-- End Gallery Item -->

         <div class="col-lg-3 col-md-4">
           <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-6.jpg"
class="glightbox" data-gallery="images-gallery">
              <img src="static/assets/img/gallery/gallery-6.jpg"
alt="" class="img-fluid">
            </a>
           </div>
         </div><!-- End Gallery Item -->

         <div class="col-lg-3 col-md-4">
           <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-7.jpg"
class="glightbox" data-gallery="images-gallery">
              <img src="static/assets/img/gallery/gallery-7.jpg"
alt="" class="img-fluid">
            </a>
           </div>
         </div><!-- End Gallery Item -->

         <div class="col-lg-3 col-md-4">
           <div class="gallery-item">
            <a href="static/assets/img/gallery/gallery-8.jpg"
class="glightbox" data-gallery="images-gallery">
              <img src="static/assets/img/gallery/gallery-8.jpg"
alt="" class="img-fluid">
            </a>
           </div>
         </div><!-- End Gallery Item -->
```

```
      </div>

    </div>

  </section><!-- /Gallery Section -->

  <!-- Contact Section -->
  <section id="contact" class="contact section">

    <!-- Section Title -->
    <div class="container section-title" data-aos="fade-up">
      <h2>Contact</h2>
      <p>Vaagdevi Engineering College is an engineering
college in Bollikunta, Warangal, Telangana, India.
        VHRX+2XR, Khammam - Warangal Hwy, Road,
Bollikunta, Telangana 50600 Phone: 0870 286 518</p>
    </div><!-- End Section Title -->

    <div class="mb-5" data-aos="fade-up" data-aos-
delay="200">

    </div><!-- End Google Maps -->

    <div class="container" data-aos="fade-up" data-aos-
delay="100">

      <div class="row gy-4">

        <div class="col-lg-4">
          <div class="info-item d-flex" data-aos="fade-up" data-
aos-delay="300">
            <i class="bi bi-geo-alt flex-shrink-0"></i>
            <div>
              <h3>Location</h3>
```

```
            <p>VHRX+2XR, Khammam - Warangal Hwy,
Road, Bollikunta, Telangana 50600</p>
          </div>
        </div><!-- End Info Item -->

        <div class="info-item d-flex" data-aos="fade-up" data-
aos-delay="400">
          <i class="bi bi-telephone flex-shrink-0"></i>
          <div>
            <h3>Call Us</h3>
            <p>+1 5589 55488 55</p>
          </div>
        </div><!-- End Info Item -->

        <div class="info-item d-flex" data-aos="fade-up" data-
aos-delay="500">
          <i class="bi bi-envelope flex-shrink-0"></i>
          <div>
            <h3>Email Us</h3>
            <p>info@example.com</p>
          </div>
        </div><!-- End Info Item -->

      </div>

      <div class="col-lg-8">
        <form action="static/forms/contact.php"
method="post" class="php-email-form" data-aos="fade-up"
data-aos-delay="200">
          <div class="row gy-4">

            <div class="col-md-6">
              <input type="text" name="name" class="form-
control" placeholder="Your Name" required="">
            </div>
```

```
            <div class="col-md-6 ">
                    <input type="email" class="form-control"
name="email" placeholder="Your Email" required="">
            </div>

            <div class="col-md-12">
              <input type="text" class="form-control"
name="subject" placeholder="Subject" required="">
            </div>

            <div class="col-md-12">
              <textarea class="form-control" name="message"
rows="6" placeholder="Message" required=""></textarea>
            </div>

            <div class="col-md-12 text-center">
              <div class="loading">Loading</div>
              <div class="error-message"></div>
              <div class="sent-message">Your message has been
sent. Thank you!</div>

              <button type="submit">Send Message</button>
            </div>

          </div>
        </form>
      </div><!-- End Contact Form -->

    </div>

  </div>

</section><!-- /Contact Section -->
```

```
  </main>

  <footer id="footer" class="footer light-background">

    <div class="container footer-top">
      <div class="row gy-4">
        <div class="col-lg-4 col-md-6 footer-about">
          <a href="index.html" class="logo d-flex align-items-
center">
            <span class="sitename">Vaagdevi Engineering
College</span>
          </a>
          <div class="footer-contact pt-3">
            <p>VHRX+2XR, Khammam - Warangal Hwy, Road,
Bollikunta, Telangana 50600</p>

            <p class="mt-3"><strong>Phone:</strong> <span>+1
5589 55488 55</span></p>
            <p><strong>Email:</strong>
<span>info@example.com</span></p>
          </div>
          <div class="social-links d-flex mt-4">
            <a href=""><i class="bi bi-twitter-x"></i></a>
            <a href=""><i class="bi bi-facebook"></i></a>
            <a href=""><i class="bi bi-instagram"></i></a>
            <a href=""><i class="bi bi-linkedin"></i></a>
          </div>
        </div>



  </footer>

  <!-- Scroll Top -->
```

```
  <a href="#" id="scroll-top" class="scroll-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>

  <!-- Preloader -->
  <div id="preloader"></div>

  <!-- Vendor JS Files -->
  <script src="static/assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script src="static/assets/vendor/php-email-form/validate.js"></script>
  <script src="static/assets/vendor/aos/aos.js"></script>
  <script src="static/assets/vendor/glightbox/js/glightbox.min.js"></script>
  <script src="static/assets/vendor/purecounter/purecounter_vanilla.js"></script>
  <script src="static/assets/vendor/swiper/swiper-bundle.min.js"></script>

  <!-- Main JS File -->
  <script src="static/assets/js/main.js"></script>

</body>

</html>
```

Result.html:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Results</title>
    <style>
        body {
            font-family: 'Helvetica Neue', Arial, sans-serif;
            background-image: url('static/assets/img/result.png');
            background-size: cover;
            background-position: center;
            background-repeat: no-repeat;
            height: 100vh;
            margin: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            color: #444;
        }

        .container {
            background: rgba(255, 255, 255, 0.9);
            padding: 30px;
            border-radius: 15px;
            box-shadow: 0 4px 20px rgba(0, 0, 0, 0.2);
            text-align: center;
            max-width: 600px;
            width: 100%;
        }

        .container h2 {
            color: #007bff;
            font-size: 2rem;
            margin-bottom: 20px;
        }
```

```css
.container h3 {
    font-weight: bold;
    font-size: 1.5rem;
    margin: 20px 0;
}

.prediction-text {
    color: #000; /* Black color for prediction text */
}

.container hr {
    border: 1px solid #ddd;
    margin: 20px 0;
}

.home-link {
    position: absolute;
    top: 20px;
    left: 20px;
    background: #007bff;
    color: #fff;
    padding: 10px 20px;
    text-decoration: none;
    border-radius: 5px;
    font-weight: bold;
}

.home-link:hover {
    background: #0056b3;
}

.back-link {
    display: inline-block;
    margin-top: 20px;
```

```
            padding: 10px 20px;
            background: #28a745;
            color: #fff;
            text-decoration: none;
            border-radius: 5px;
            font-weight: bold;
        }

        .back-link:hover {
            background: #218838;
        }

        @media (max-width: 768px) {
            .container {
                width: 90%;
                padding: 20px;
            }

            .container h2 {
                font-size: 1.5rem;
            }

            .container h3 {
                font-size: 1.25rem;
            }
        }
    </style>
</head>

<body>
    <a href="/" class="home-link">Home</a>
    <div class="container">
        <h2>Results</h2>
        <hr>
```

```
        <h3>The probable diagnosis is: <span class="prediction-
text">{{ prediction_text }}</span></h3>
        <hr>
        <p><a href="/details" class="back-link">Try Another
Prediction</a></p>
    </div>
</body>

</html>
```

App.py:

```
from flask import Flask, render_template, request
import numpy   as np
import pickle

app = Flask(__name__)

# Load the model
model = pickle.load(open('6.Disease prediction executable
files/model.pkl', 'rb'))

# Define routes
@app.route("/")
def home():
    return render_template('index.html')

@app.route('/details')
def details():
    return render_template('details.html')

@app.route('/predict', methods=['POST'])
def predict():
    # Define the list of symptoms
```

```
    symptoms = ['itching', 'continuous sneezing', 'joint pain',
'vomiting',
        'spotting_ urination', 'fatigue', 'weight loss', 'lethargy',
        'high fever', 'sunken eyes', 'sweating', 'headache', 'dark
urine',
        'nausea', 'loss of appetite', 'pain behind the eyes',
'abdominal pain',
        'diarrhoea', 'mild fever', 'yellowing of eyes', 'swelled
lymph nodes',
        'malaise', 'phlegm', 'congestion', 'chest pain', 'fast heart
rate',
        'irritation in anus', 'swollen legs', 'puffy face and eyes',
        'excessive hunger', 'muscle weakness', 'movement
stiffness',
        'weakness of one body side', 'bladder discomfort',
'depression',
        'irritability', 'muscle pain', 'red spots over body',
        'abnormal_menstruation', 'increased appetite', 'mucoid
sputum',
        'rusty sputum', 'lack of concentration', 'receiving blood
transfusion',
        'coma', 'history of alcohol consumption', 'blood in sputum',
        'palpitations', 'inflammatory nails', 'yellow crust ooze']

    if request. method == 'POST':
        input = [str(x) for x in request.form.values()]

        b = [0] * 50
        for x in range(0, 50):
            for y in input:
                if symptoms[x] == y:
                    b[x] = 1
        b = np.array(b)
        b = b.reshape(1, 50)
        prediction = model.predict(b)
```

```
        prediction = prediction[0]
        return render_template('results.html', prediction_text="
{}".format(prediction))

if __name__ == "__main__":
    app.run()
```

10.2 GitHub and project Demo link:

Github link: **Click Here**

Project Demo link: **Click Here**