



## **CAPSTONE PROJECT REPORT**

**SUBJECT:** Big Data Analytics

**SOURCE CODE:** 19CSE357

### **SUBMITTED TO:**

Dr. Vidhyasagar

CSE Department

### **SUBMITTED BY:**

**TEAM MEMBERS:** (CSE-B)

Devakrishnan T U – 20117

Jogi Lokesh Kumar – 20129

Manguluru Hanumesh – 20127

Kolagani Sankar Brahmachari – 20135

# **DISEASE PREDICTION USING DJANGO AND MONGODB**

## **INTRODUCTION**

Disease prediction using machine learning models has become an essential area of research in healthcare, and Django and MongoDB provide a robust and scalable platform to build and deploy disease prediction applications. The project aims to develop a disease prediction application that uses machine learning to predict the likelihood of a patient developing a particular disease based on their medical history. The application is user-friendly, efficient, and complies with healthcare industry data privacy and security standards. Additionally, the application provides a dashboard for healthcare providers to monitor the predictions, is easy to deploy on a web server, and the model should be easily updatable with new data.

The goal of this application is to provide a quick and efficient way for individuals to assess potential health issues based on their symptoms. The application collects data related to diseases and symptoms from various sources, pre-processes the data, and trains a machine-learning model to predict potential diseases based on input symptoms.

Django is an excellent choice for developing web applications due to its robustness, scalability, and flexibility. It provides a user-friendly interface for users to input their symptoms and receive disease predictions. MongoDB is a NoSQL database that can handle unstructured data, making it an ideal choice for storing data related to diseases and symptoms.

The disease prediction application using Django and MongoDB has the potential to revolutionize the healthcare industry. It provides a quick and efficient way for individuals to assess their health, which could lead to early detection and treatment of diseases. This application could also reduce the workload of healthcare professionals, allowing them to focus on more critical tasks.

## **USED FRAMEWORK**

**Frontend** – HTML, CSS, Bootstrap

**Backend** – MongoDB

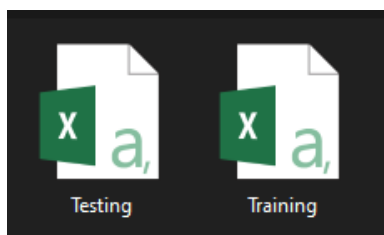
**Framework** - Django

## **APPLIED ML ALGORITHMS**

Logistic, SVM, Decision Tree, Random Forest, Navie Bayer's.

## **METHODOLOGY:**

### **Data Collection:**



### **About Dataset:**

This dataset will help you apply your existing knowledge to great use. Applying Knowledge to the field of Medical Science and making the task of physicians easy is the main purpose of this dataset. This dataset has **132** parameters on which **42** different types of diseases can be predicted.

### **Data Pre-processing:**

The pre-processing steps involve removing duplicates to ensure that there is no unnecessary redundancy in the data. Additionally, missing values and irrelevant data are also removed to ensure that the data is clean and accurate.

Once the pre-processing steps are complete, the data is split into training and testing datasets. The training dataset is used to train the machine learning model, while the testing dataset is used to evaluate its performance.

The splitting of data is crucial as it allows for the evaluation of the model's performance on unseen data, which is essential for determining its accuracy and reliability.

### **Model Training:**

The Decision Tree Classifier machine learning algorithm was used to train the model. The algorithm was trained on the pre-processed data to predict potential diseases based on symptoms input by the user.

### **Application Development:**

A web application was developed using Django. The application allows users to input their symptoms, which are processed by the machine learning model to predict potential diseases. The application was designed with a clean and user-friendly interface.

### **Database Integration:**

MongoDB was used as the database for storing the data collected in Step 1. MongoDB's flexibility in handling unstructured data made it ideal for this project.

## **INTEGRATION OF DJANGO WITH MONGODB**

```
DATABASES = {
    'default': {
        'ENGINE': 'django',
        'NAME': 'data',
    }
}
```

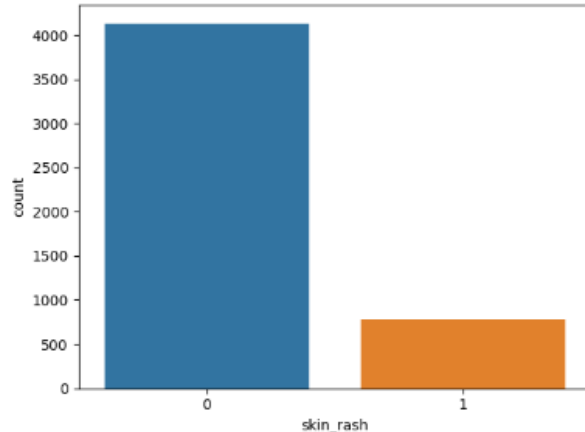
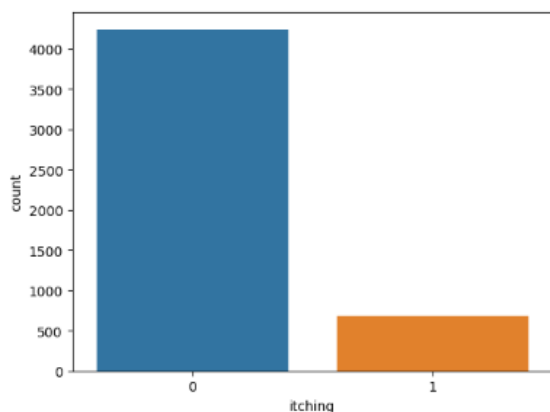
## Selecting the Target Variable:

TARGET VARIABLE = "Prognosis"

```
In [13]: train_set['prognosis']
```

```
Out[13]: 0          Fungal infection
1          Fungal infection
2          Fungal infection
3          Fungal infection
4          Fungal infection
...
4915      (vertigo) Paroymsal   Positional Vertigo
4916                                     Acne
4917          Urinary tract infection
4918                                     Psoriasis
4919                                     Impetigo
Name: prognosis, Length: 4920, dtype: object
```

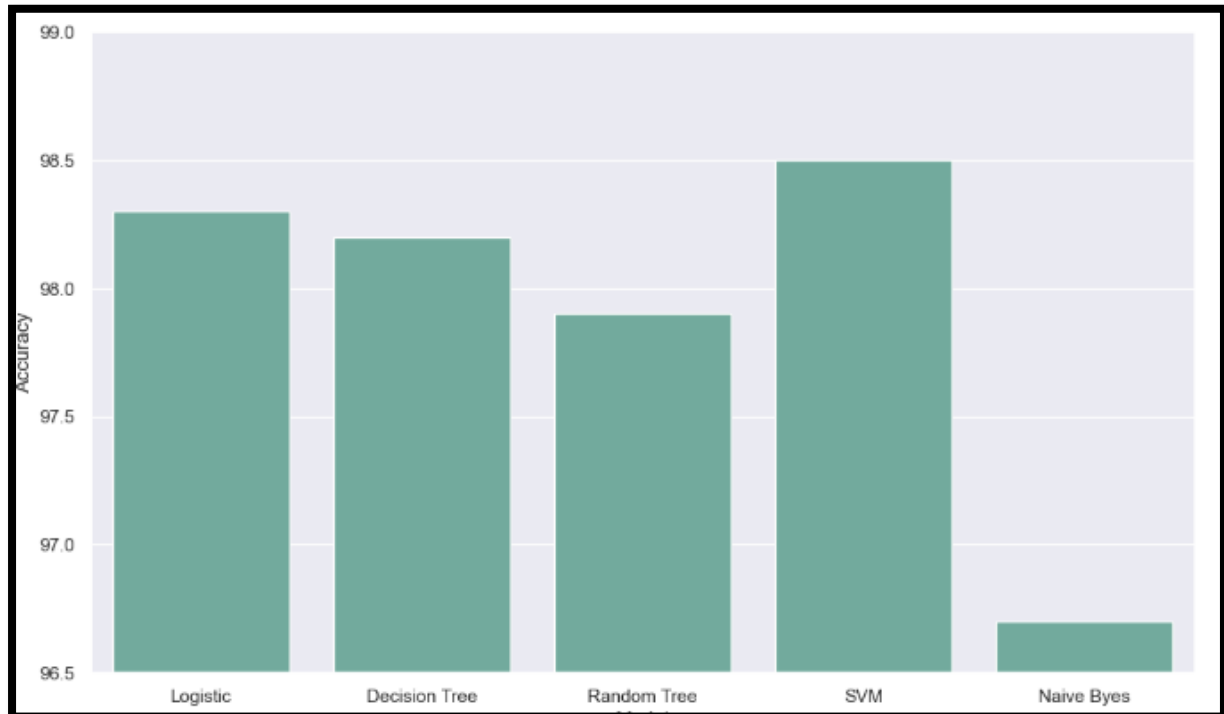
```
In [14]: for col in train_set.columns:
          if col == 'prognosis':
              continue
          sns.countplot(data = train_set , x = col)
          plt.show()
```



In the graph shown, the horizontal axis represents the count of individuals, while the vertical axis represents the symptoms given. The values on the vertical axis are binary, where 0 represents a person with no disease and 1 represents a person having the disease with the given symptom.

The graph provides a visual representation of the relationship between the count of individuals and the symptoms associated with the disease. It allows us to identify which symptoms are most commonly associated with the disease and how many individuals have the disease with the given symptom.

Overall, the graph provides valuable insights into the relationship between symptoms and the occurrence of a particular disease, which can be used to develop more effective disease prediction models.



In the above graph Compared to other classification algorithms such as Decision Tree, Random Forest, Naive Bayer's, and Logistic Regression, SVM tends to perform better when dealing with complex data that has a high degree of variability.

In the context of disease prediction, SVM can effectively handle cases where the relationship between symptoms and diseases is nonlinear or complex.

Furthermore, SVM has a regularization parameter that helps to reduce the risk of overfitting the model to the training data, which is a common problem in machine learning.

This regularization parameter helps to improve the generalization ability of the model and makes it more accurate in predicting unseen data.

Overall, the higher accuracy of SVM in disease prediction compared to other classification algorithms may be attributed to its ability to effectively handle complex and nonlinear data, and its regularization parameter that helps to prevent overfitting.

## DEMONSTRATION:

## HOME

### PREDICO

## " BE YOUR OWN DOCTOR "

DISEASE PREDICTION

Home About Contact Signup Signin

BE YOUR OWN DOCTOR ...

It might have happened so many times that you or someone yours need doctor's help immediately, but they are not available due to some reasons. The Disease Prediction system is an end user support. Here, we created a web application that allow users to get instant guidance on their disease through an intelligent system. Here we use some intelligent data mining techniques to guess the most accurate illness that could be associated with patient's details. Disease prediction has the potential to revolutionize healthcare by enabling early intervention and preventive measures. By identifying individuals who are at high risk for certain diseases, doctors and healthcare professionals can take proactive steps to prevent or mitigate the impact of those diseases. For example, patients who are identified as being at risk for diabetes can be given lifestyle advice and interventions to prevent or delay the onset of the disease.

Get started

## CONTACT PAGE:

### Contact Us

Name:

lokesh

Email:

lokesh@gmail.com

Message:

Submit

### Address

Thiruvallur District

Chennai, 630010

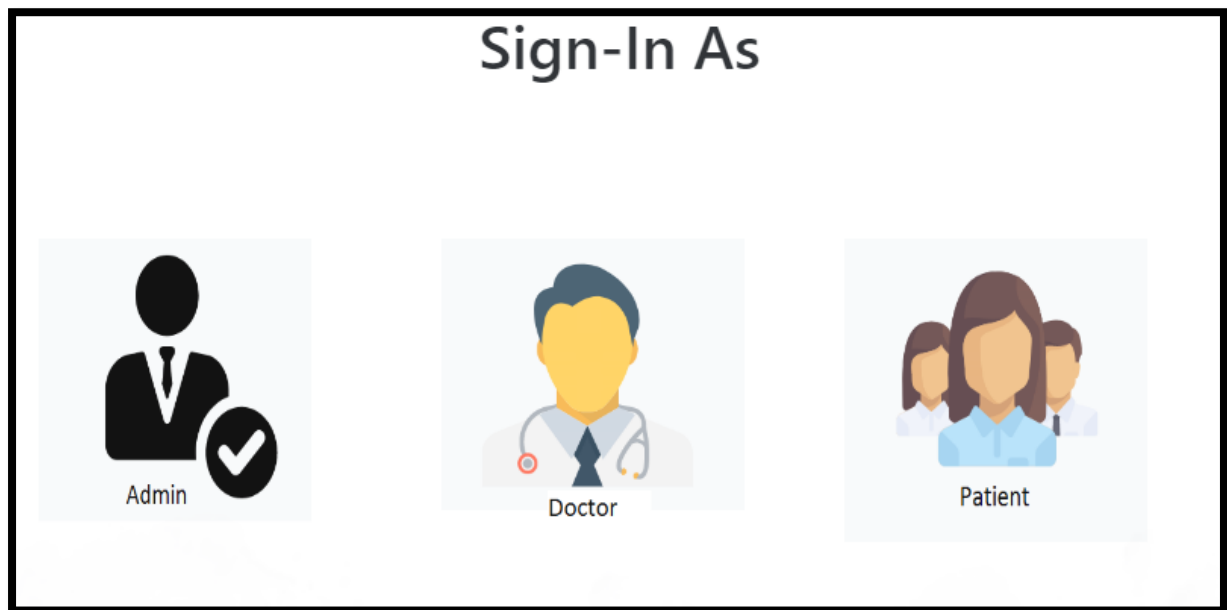
### Phone

(+91) 9078166555

### Email

honeshadevaloki@gmail.com

**Sign-In as page:**



**Login as Doctor:**

DISEASE PREDICTION [Home](#) [About](#) [Contact](#) [Signup](#) [Signin](#)

## Login as Doctor

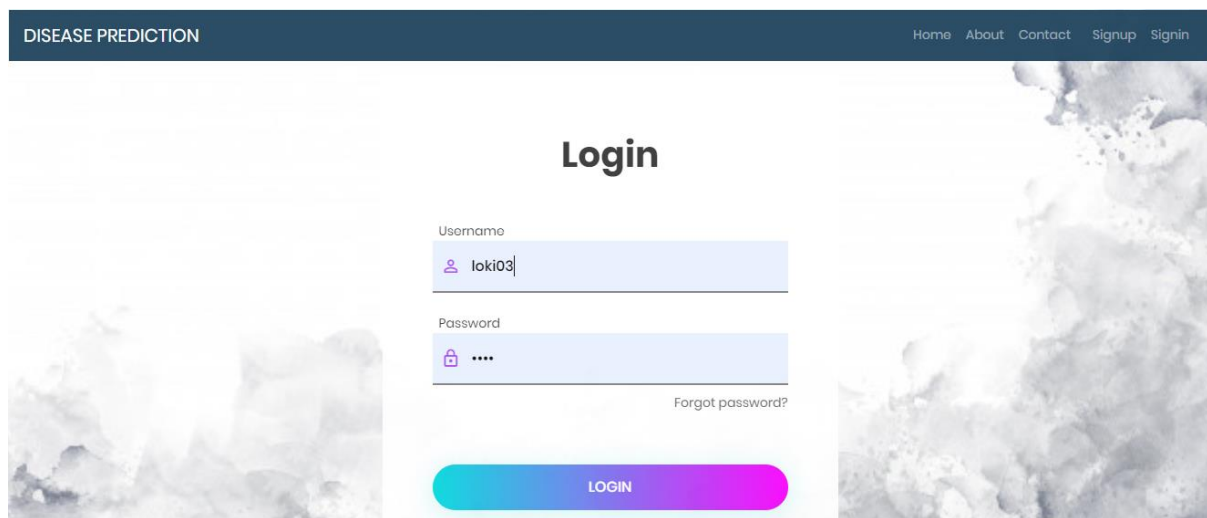
Username

Password

[Forgot password?](#)

[LOGIN](#)

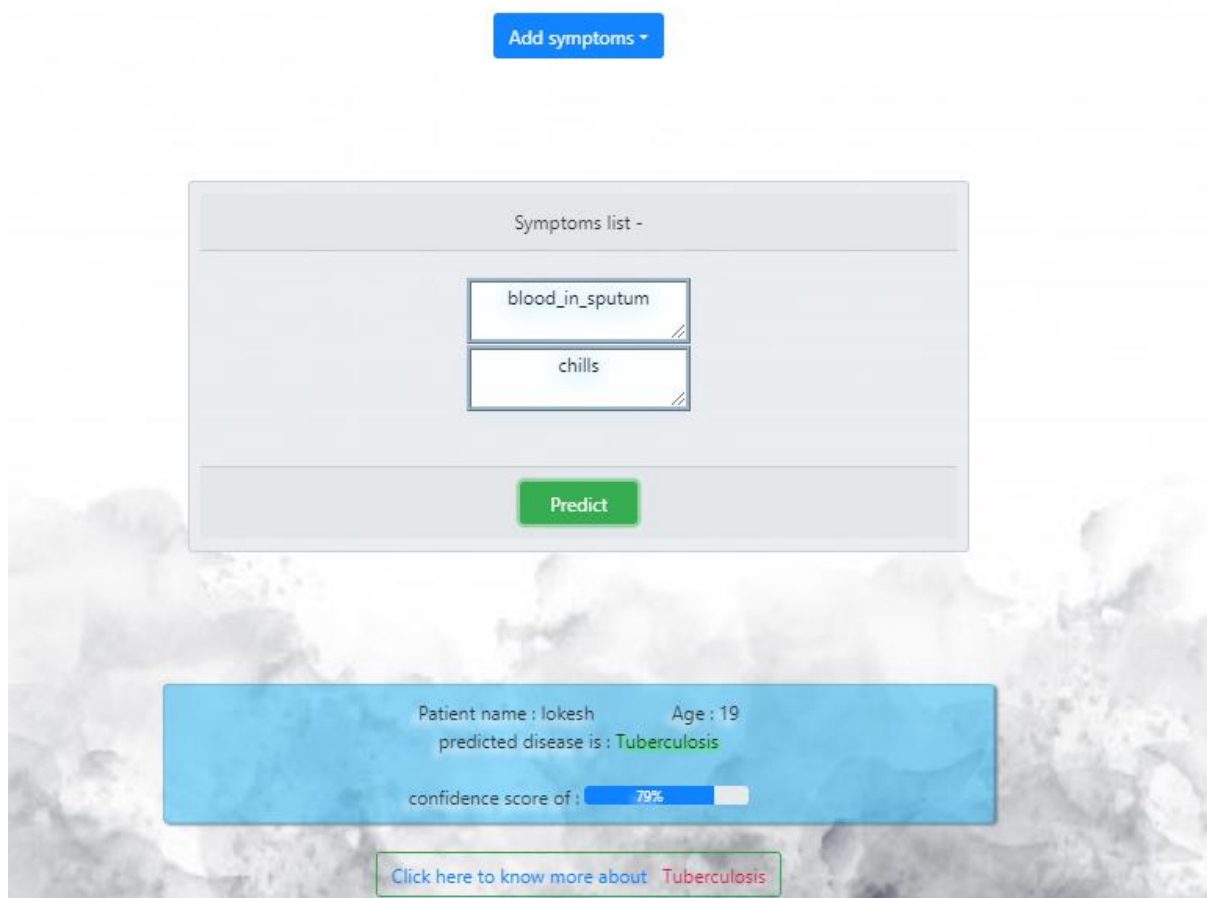
## Login as Patient:



The login interface features a dark blue header with the text "DISEASE PREDICTION" on the left and navigation links "Home", "About", "Contact", "Signup", and "Signin" on the right. The main content area has a light gray background with a large, stylized "Login" title in the center. Below the title, there are two input fields: "Username" with the value "loki03" and "Password" with masked characters "\*\*\*\*". A link "Forgot password?" is positioned to the right of the password field. A large, rounded "LOGIN" button with a blue-to-purple gradient is centered below the input fields.

With the given input symptoms, here it is predicting the symptoms:

Identify possible conditions and treatment related to your symptoms.



The symptom prediction interface includes a blue "Add symptoms" button at the top. Below it is a "Symptoms list" box containing two input fields with the text "blood\_in\_sputum" and "chills". A green "Predict" button is located at the bottom of this box. Below the symptoms list, a blue box displays the patient information: "Patient name : lokesh", "Age : 19", "predicted disease is : Tuberculosis", and "confidence score of : 79%". A link "Click here to know more about Tuberculosis" is provided at the bottom.

For the given input the predicted Disease is Tuberculosis with score of 79%



**Results:**

In the disease prediction project using Django and MongoDB, a web application was developed that allows users to input their symptoms, and the application accurately predicts potential diseases. The development of the application involved integrating the machine learning models trained on the pre-processed data into a Django web framework, which allows for the creation of dynamic and responsive web applications.

The user interface of the web application was designed to be intuitive and easy to use, with clear and concise instructions for users on how to input their symptoms and obtain the predicted diseases.

The user interface was also designed to be visually appealing, with clear and easy-to-read text and graphics.

Overall, the development of the disease prediction web application using Django and MongoDB represents a significant achievement in the field of healthcare, as it provides an efficient and accurate means of predicting potential diseases based on symptoms. The intuitive and easy-to-use user interface of the application further enhances its usability, making it accessible to a wide range of users.

**Conclusion:**

Disease prediction using Django and MongoDB is an important area of research that aims to improve healthcare outcomes through the use of machine learning algorithms. This project involved several key steps, including data collection, pre-processing, and splitting into training and testing datasets.

Various machine learning algorithms such as Decision Tree, Random Forest, Naive Bayes, Logistic Regression, and Support Vector Machine (SVM) were used for disease prediction, with SVM performing the best in terms of accuracy and reliability.

By accurately predicting the occurrence of diseases based on various symptoms and risk factors, disease prediction using Django and MongoDB has the potential to revolutionize healthcare and enable early detection and prevention of diseases, ultimately leading to better healthcare outcomes for patients.

**\*\*\*THE END\*\*\***