INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT

AKURDI, PUNE

Documentation On:

“Disease Prediction Using Deep Learning”

PG-DBDA FEB 2021

Submitted By:

Group No: 4

Sourabh Patil 1545

Ashish Singh Rao 1505

Mr. Prashant Karhale
Centre Coordinator

Mr. Akshay Tilekar
Project Guide

Abstract

Artificial neural network models were developed to perform disease detection and diagnosis using simple non metric independent and dependant data, through deep learning methodologies. Training of the models was performed with the use of an open database of 5000 records, containing different diseases in a set of 41 distinct classes of diseases. The significantly high success rate makes the model a very useful advisory or early warning tool, and an approach that could be further expanded to support an integrated disease identification system to operate in real cultivation conditions.

Keywords: Predictive Modeling, Artificial Neural Network

Chapter 1

1.1 Introduction

Machine Learning is an emerging approach that helps in prediction, diagnosis of a disease. Prediction of disease based on symptoms using machine learning. Previously used Machine Learning algorithms such as Naive Bayes, Decision Tree and Random Forest are employed on the provided dataset and predict the disease.

With big data growth in biomedical and healthcare communities, accurate analysis of medical data benefits early disease detection, patient care and community services. However, the analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease outbreaks. In this System, we streamline Deep learning algorithms for effective prediction of diseases.

To prevent this situation need better and perfect guidance to make the correct identification of diseases, and the ability to distinguish between two or more similar types of diseases in visuals.

This is where Artificial Neural Networks comes which is used for modelling non-linear problems and predict the output values for given input parameters from their training values.

1.2 Purpose

Detection of disease through some technique is beneficial as it reduces a large work of monitoring every symptom for disease, and at very early stage itself it detects the disease that is appear on person’s body and their internal conditions.

Project deals with the real time detection of diseases that affect the plant and the area affected using Artificial Neural Network (ANN) Model.

Artificial neural network models were developed to perform disease detection and diagnosis using different symptoms of person’s through deep learning methodologies. So that appropriate medicine can be used to prevent further damage to health from different viruses.

1.3 Scope of the project

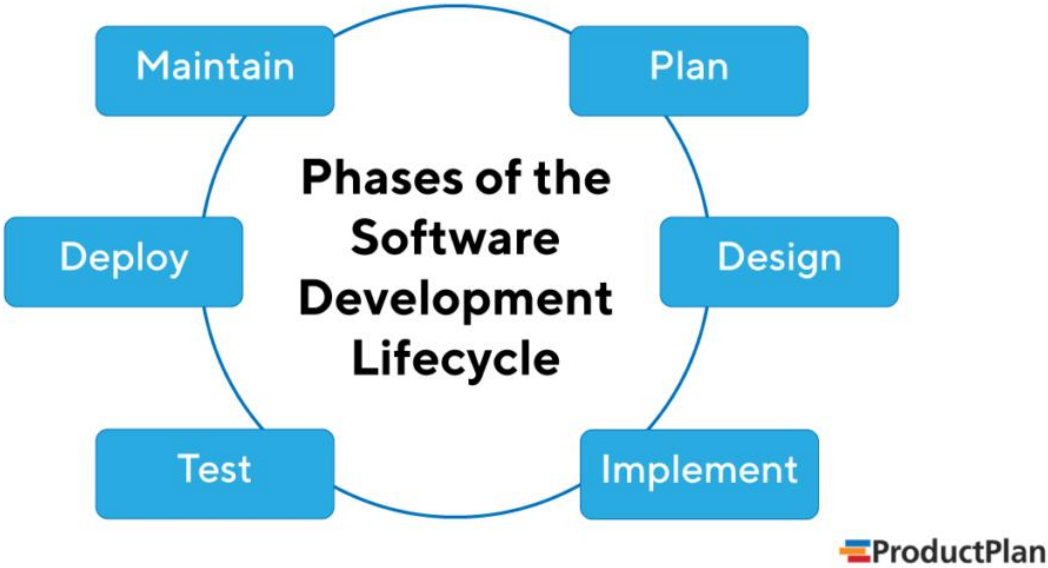
This project aims to provide a web platform to predict the occurrences of disease on the basis of various symptoms. The user can enter various symptoms and can find disease with their probabilistic figures.

Automatic detection of diseases can be detected with the help of Artificial Neural Network. ANN plays important role in detection of diseases, provide best result and reduces the human efforts.

Chapter 2

2.1Software Life Cycle Model

In order to make this Project we are going to use Classic LIFE CYCLE MODEL. Classic life cycle model is also known as WATER FALL MODEL. The life cycle model demands a Systematic sequential approach to software development that begins at the system level and progress through analysis design coding, testing and maintenance.



The stages of SDLC are as follows:

Stage1: Planning and requirement analysis

Requirement Analysis is the most important and necessary stage in SDLC.

The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry.

Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.

Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end

user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary.

For Example, A client wants to have an application which concerns money transactions. In this method, the requirement has to be precise like what kind of operations will be done, how it will be done, in which currency it will be done, etc.

Once the required function is done, an analysis is complete with auditing the feasibility of the growth of a product. In case of any ambiguity, a signal is set up for further discussion.

Once the requirement is understood, the SRS (Software Requirement Specification) document is created. The developers should thoroughly follow this document and also should be reviewed by the customer for future reference.

Stage2: Defining Requirements

Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.

This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.

Stage3: Designing the Software

The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.

Stage4: Developing the project

In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

Stage5: Testing

After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage.

During this stage, unit testing, integration testing, system testing, acceptance testing are done.

Stage6: Deployment

Once the software is certified, and no bugs or errors are stated, then it is deployed.

Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.

After the software is deployed, then its maintenance begins.

Stage7: Maintenance

Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.

This procedure where the care is taken for the developed product is known as maintenance.

2.2 Overall Description

2.2.1 Data:

Independent Data	Target Data
'itching', 'skin_rash','nodal_skin_eruptions', 'continuous_sneezing', 'shivering', 'chills', 'joint_pain', 'stomach_pain', 'acidity', ……., 'red_sore_around_nose', 'yellow_crust_ooze'	'prognosis'
Data is in non-metric format i.e. 0 and 1.	There are 41 classes for target data

2.2.2 Imports:

- Matplotlib: Matplotlib is a collection of command style functions that make matplotlib work like MATLAB. Each pilots function makes some change to a figure. We have used it to show visualizations of analysis.
- Numpy: Numpy is used to for mathematical operations. This package provides easy use of mathematical function.
- TensorFlow: TensorFlow is open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.
- Keras: Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3 Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, R, Theano, and PlaidML.
- Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, and clustering and dimensionality reduction via a consistence interface in Python.

2.3 Requirement Specification

2.3.1 Initial non-functional requirement will be:

- Getting the large datasets which can provide developer enough data to train the model.
- Maintain the minimum variance and bias so the model is successfully work.
- Avoid the underfitting and overfitting.

2.3.2 Initial functional requirement will be:

- Selecting the appropriate algorithms.
- Determining the appropriate input format to algorithm.
- Train the model.
- Test the model.

2.3.3 Hardware Requirement:

- Processor: Intel Dual Core and above
- RAM: Minimum 4GB
- OS: Linux

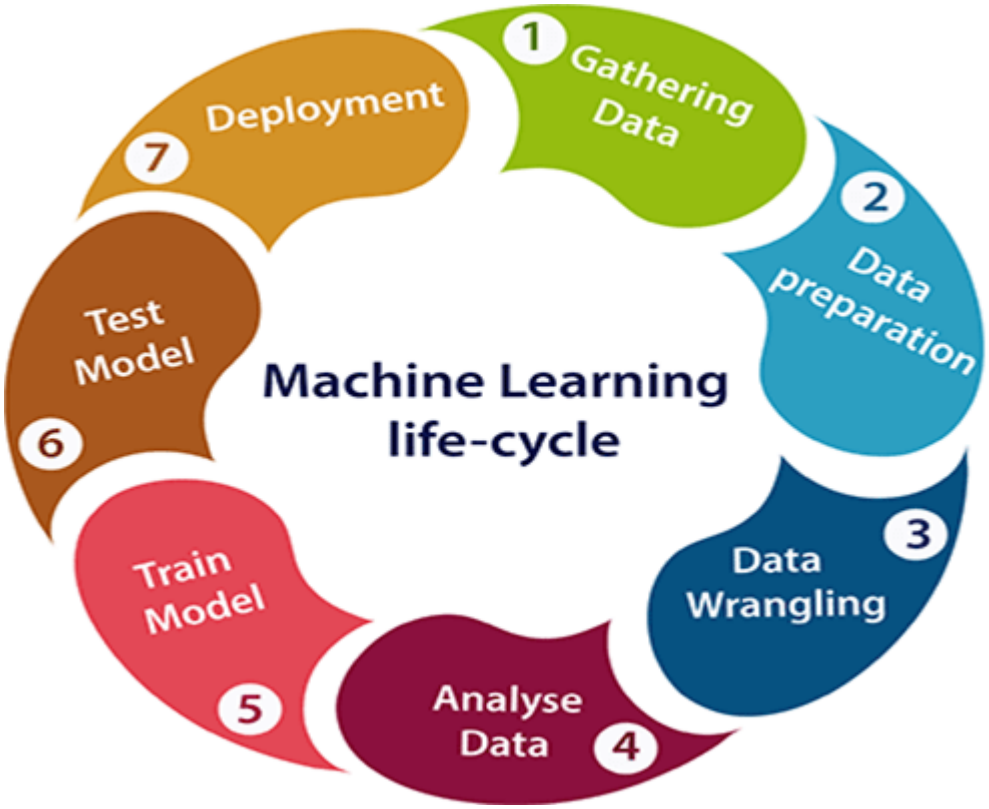
2.3.4 Software Requirement:

- Python3
- Google Colab
- Django Framework

Chapter 3

3.1 Data Analytics Lifecycle

The Data Analytics lifecycle is designed for Big Data problems and data science projects. The cycle is iterative to represent real project. To address the distinct requirements for performing analysis on Big Data, step – by – step methodology is needed to organize the activities and tasks involved with acquiring, processing, analyzing, and repurposing data.



1. **Gathering Data:**

   Data collection is defined as the procedure of collecting, measuring and analyzing accurate insights for research using standard validated techniques. A researcher can evaluate their hypothesis on the basis of collected data. In most cases, data collection is the primary and most important step for research, irrespective of the field of research. The

approach of data collection is different for different fields of study, depending on the required information.

2. **Data preparation:**

   Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data.

3. **Data Wrangling:**

   Data wrangling is the process of gathering, selecting, and transforming data to answer an analytical question. Also known as data cleaning or "munging", legend has it that this wrangling costs analytics professionals as much as 80% of their time, leaving only 20% for exploration and modeling.

4. **Data Analysis:**

   Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

5. **Train Model:**

   The sample of data used to fit the model. The actual dataset that we use to train the model (weights and biases in the case of a Neural Network). The model *sees* and *learns* from this data.

6. **Test Model:**

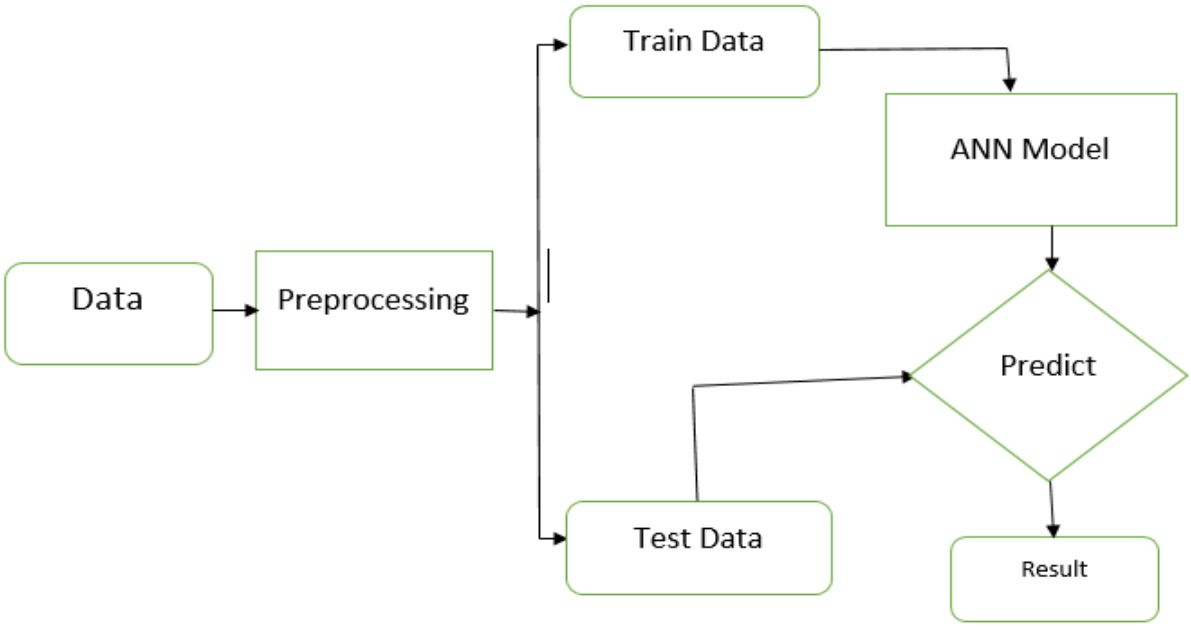
   The Test dataset provides the gold standard used to evaluate the model. It is only used once a model is completely trained (using the train and validation sets). The test set is generally what is used to evaluate competing models. Many a times the validation set is used as the test set, but it is not good practice. The test set is generally well curated. It contains carefully sampled data that spans the various classes that the model would face, when used in the real world.

7. **Deployment:**
   Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that is useful to the customer.

3.2 System Design

3.2.1 Flowchart of the System:

The flowchart of the algorithm is represented in Figure



The above flowchart describes the working flow of the project. First pre-processed the data and selected the model. Then algorithm was later implemented in python and deployed on web. The model was later tested against raw data.

Chapter 4

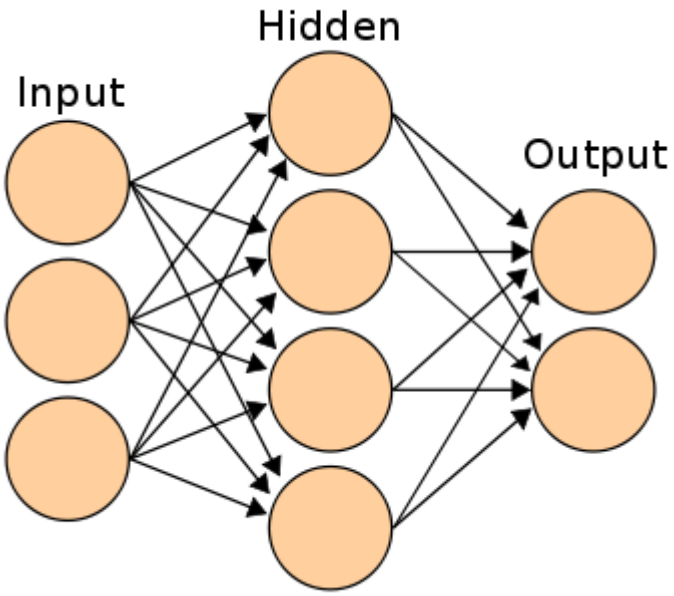
4.1 Model Building

4.1.1 Algorithm Research and Selection:

For this classification problem, following deep learning model was used:

Artificial Neural Network

We will start with understanding formulation of a simple hidden layer neural network. A simple neural network can be represented as shown in the figure below:



The linkages between nodes are the most crucial finding in an ANN. The only known values in the above diagram are the inputs. Let's call the inputs as I1, I2 and I3, Hidden states as H1, H2, H3 and H4, Outputs as O1 and O2. The weights of the linkages can be denoted with following notation:

W (I1H1) is the weight of linkage between I1 and H1 nodes.

Following is the framework in which artificial neural networks (ANN) work:

- Assign Random weights to all the linkages to start the algorithm
- Using the inputs and the (input->hidden node) linkages find the activation rate of Hidden Nodes.

- Using the activation rate of Hidden nodes and linkages to Output, find the activation rate of Output Nodes.
- Find the error rate at the output node and recalibrate all the linkages between hidden nodes and output nodes.
- Using the weight and error found at Output nodes, cascade down the error to Hidden Nodes.
- Recalibrate the weight between hidden nodes and the input nodes.
- Repeat the process till the convergence criterion is met
- Using the final linkages weights score the activation rate of the output

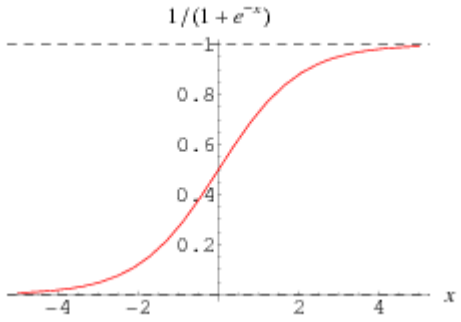
Few statistical details about the framework

Every linkage calculation in an Artificial Neural Network (ANN) is similar. In general, we assume a sigmoid relationship between the input variables and the activation rate of hidden nodes or between the hidden nodes and the activation rate of output nodes. Let's prepare the equation to find activation rate of H1.

Logit (H1) = W (I1H1) * I1 + W (I2H1) * I2 + W (I3H1) * I3 + Constant = f

= > P (H1) = 1/ (1+e^ (-f))

Following is how the sigmoid relationship looks like:



Main Steps to build a CNN (or) Conv. net:

1. Dense Layer
2. ReLU Layer (Rectified Linear Unit)
3. Dropout
4. Optimizer

1. Dense Layer

The dense layer is a neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer. The dense layer is found to be the most commonly used layer in the models.

In the background, the dense layer performs a matrix-vector multiplication. The values used in the matrix are actually parameters that can be trained and updated with the help of backpropagation.

The output generated by the dense layer is an 'm' dimensional vector. Thus, dense layer is basically used for changing the dimensions of the vector. Dense layers also applies operations like rotation, scaling, translation on the vector.

2. ReLU Layer

ReLU stands for the Rectified Linear Unit for a non-linear operation. The output is ƒ(x) = max (0, x). We use this because to introduce the non-linearity to CNN.

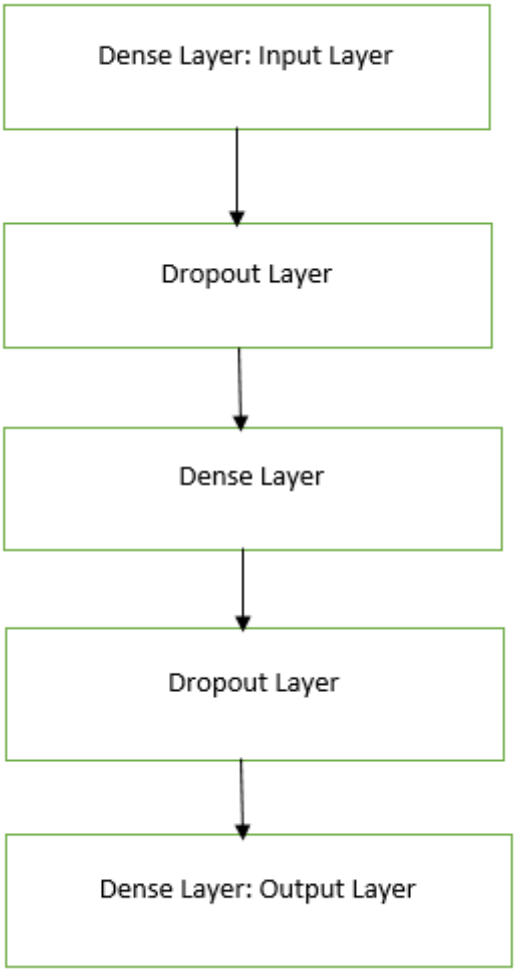
3. Dropout

Dropout is a regularization method that approximates training a large number of neural networks with different architectures in parallel.
During training, some number of layer outputs are randomly ignored or "*dropped out*." This has the effect of making the layer look-like and be treated-like a layer with a different number of nodes and connectivity to the prior layer. In effect, each update to a layer during training is performed with a different "*view*" of the configured layer.

4. Optimizers

Optimizers are algorithms or methods used to change the attributes of the neural network such as weights and learning rate to reduce the losses. Optimizers are used to solve optimization problems by minimizing the function.

4.2 Model Flowchart



This is the flow chart of ANN model as explained in section 4.2

4.3: Algorithm Implementation

To implement the idea of disease detection and to train the machine accordingly requires lot of steps which are mentioned below: -

1. Load Label Data for input like Training Data, Testing Data in folder.
2. Import all libraries like NumPy, Pandas, Matplotlib, Tensorflow, Keras, Dense, Sequential Model, DropOut, and LabelEncoder.
3. Perform pre-processing of data, check whether any null values in data, and check all data in numeric format.
4. Convert the non-numeric data into numeric form using LabelEncoder
5. Divide the data in training data and test data with 20% for testing data And 80% of data for training.
6. Use Artificial Neural Network (ANN) for classification of data into different classes.
7. Add first dense layer with 32 neurons and ReLU activation function for loss initialization, add input parameters for prediction.
8. Add dropout layer to avoid overfitting from model.
9. Add second dense layer with 64 neurons and ReLU activation function for loss initialization.
10. Add second dropout layer to avoid overfitting from model.
11. Add third dense layer with 128 neurons and ReLU activation function for loss initialization.
12. Add third dropout layer to avoid overfitting from model.
13. Add forth dense layer for predicting output of data with 41 neuron for 41 classes. Use activation function SoftMax to predict class more than two classes.
14. Optimize model using Adam optimizer and loss function is used Sparse_categoricalentropy for classification of 41 classes.
15. Model Fit Function is used to fit all variables like training set, steps per epoch=200, epochs=150.
16. Apply predict function to get prediction of test data.
17. Apply confusion matrix to get actual values and predicted values difference.
18. Last step is save the model.

4.4 Test Result

4.4.1 Training Accuracy and Testing Accuracy

```
accuracy: 0.9055 - val_loss: 0.5112 - val_accuracy: 0.8663

accuracy: 0.8944 - val_loss: 0.0600 - val_accuracy: 0.9737

accuracy: 0.9156 - val_loss: 0.0850 - val_accuracy: 0.9615

accuracy: 0.9102 - val_loss: 0.0870 - val_accuracy: 0.9792

accuracy: 0.9021 - val_loss: 0.0543 - val_accuracy: 0.9737

accuracy: 0.9190 - val_loss: 0.1025 - val_accuracy: 0.9652

accuracy: 0.9083 - val_loss: 0.0735 - val_accuracy: 0.9841

accuracy: 0.9206 - val_loss: 0.0591 - val_accuracy: 0.9841

accuracy: 0.8691 - val_loss: 0.0649 - val_accuracy: 0.9829

accuracy: 0.9035 - val_loss: 0.0614 - val_accuracy: 0.9737

accuracy: 0.9015 - val_loss: 0.0714 - val_accuracy: 0.9737

accuracy: 0.9091 - val_loss: 0.0549 - val_accuracy: 0.9737

accuracy: 0.8853 - val_loss: 0.1103 - val_accuracy: 0.9518
```

- We have applied Artificial Neural Network on training data and got accuracy of 88%.
- We got test accuracy of 95%

4.5: User Interface

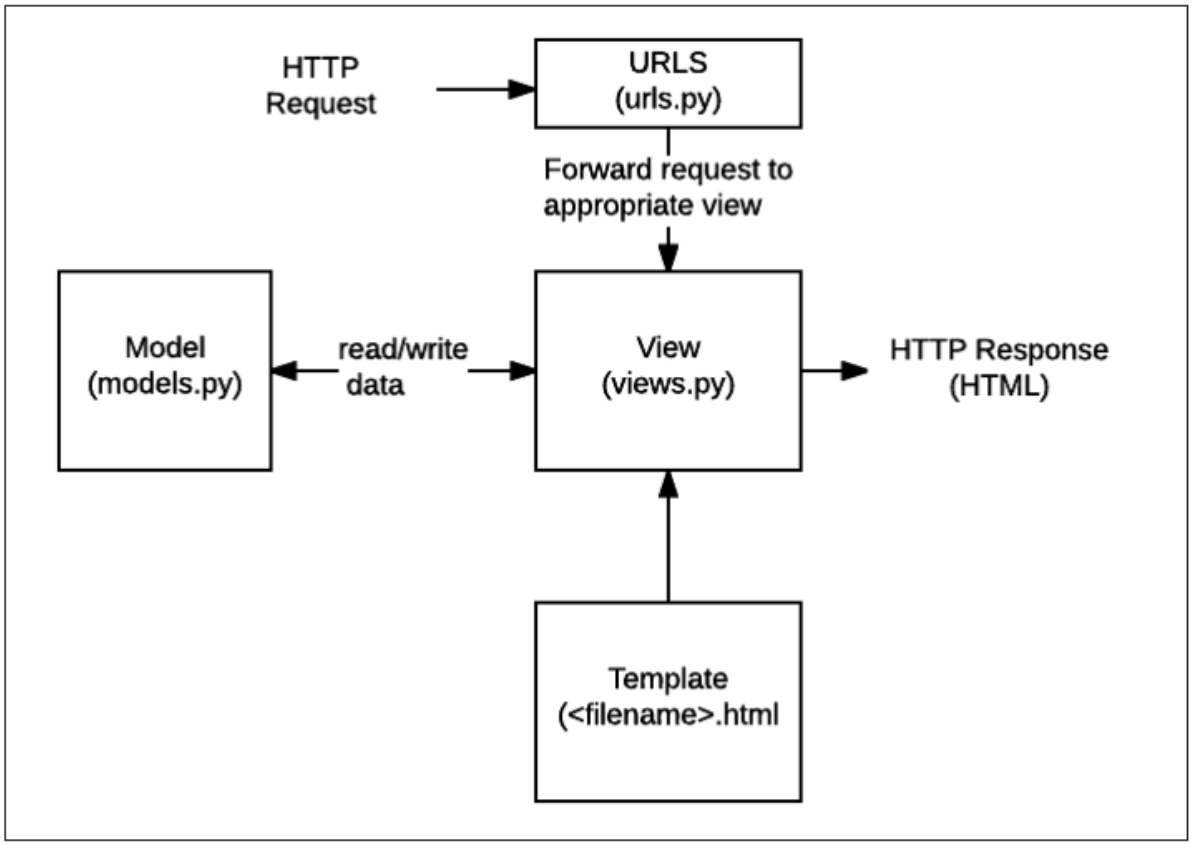
4.5.1: Django Framework:

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Ridiculously fast: Django was designed to help developers take applications from concept to completion as quickly as possible.
Reassuringly secure: Django takes security seriously and helps developers avoid many common security mistakes.
Exceedingly scalable: Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

Django web applications typically group the code that handles each of these steps into separate files:

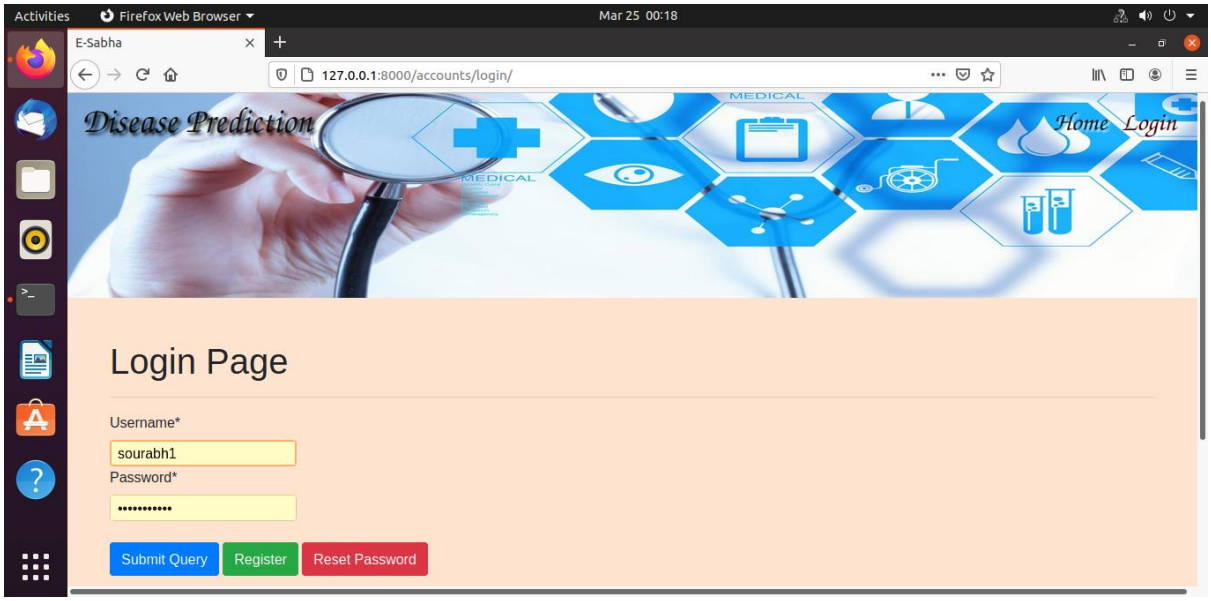
- URLs: While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.
- View: A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via *models*, and delegate the formatting of the response to *templates*.
- Models: Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
- Templates: A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A *view* can dynamically create an HTML page using an HTML template, populating it with data from a *model*. A template can be used to define the structure of any type of file; it doesn't have to be HTML!



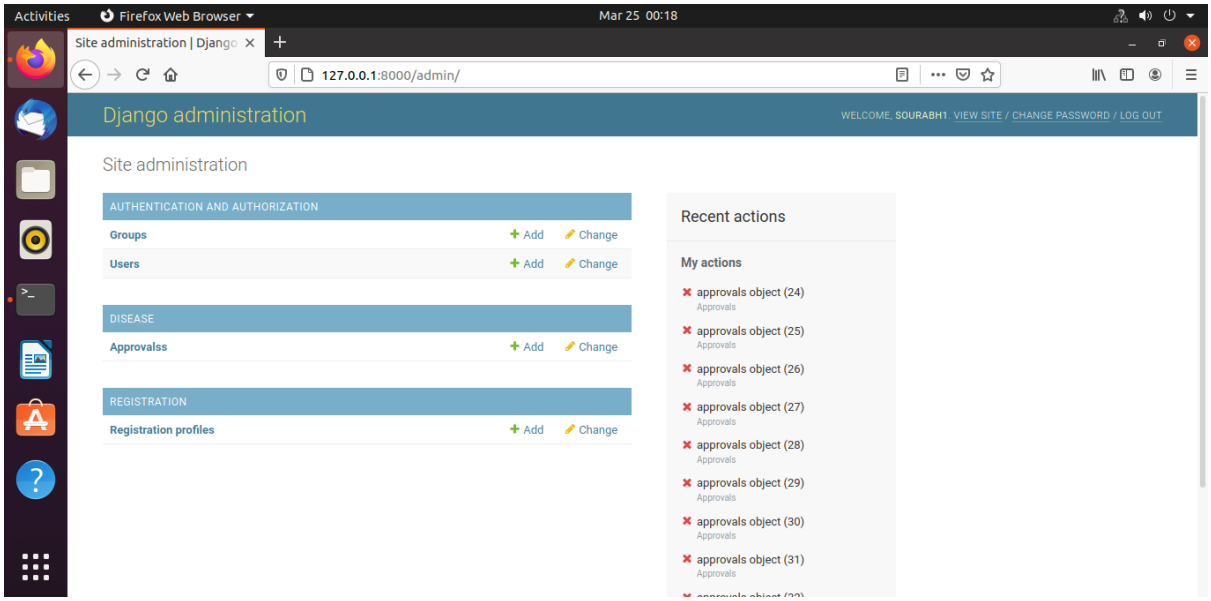
4.5.2 Disease Prediction User Interface:

Home page:

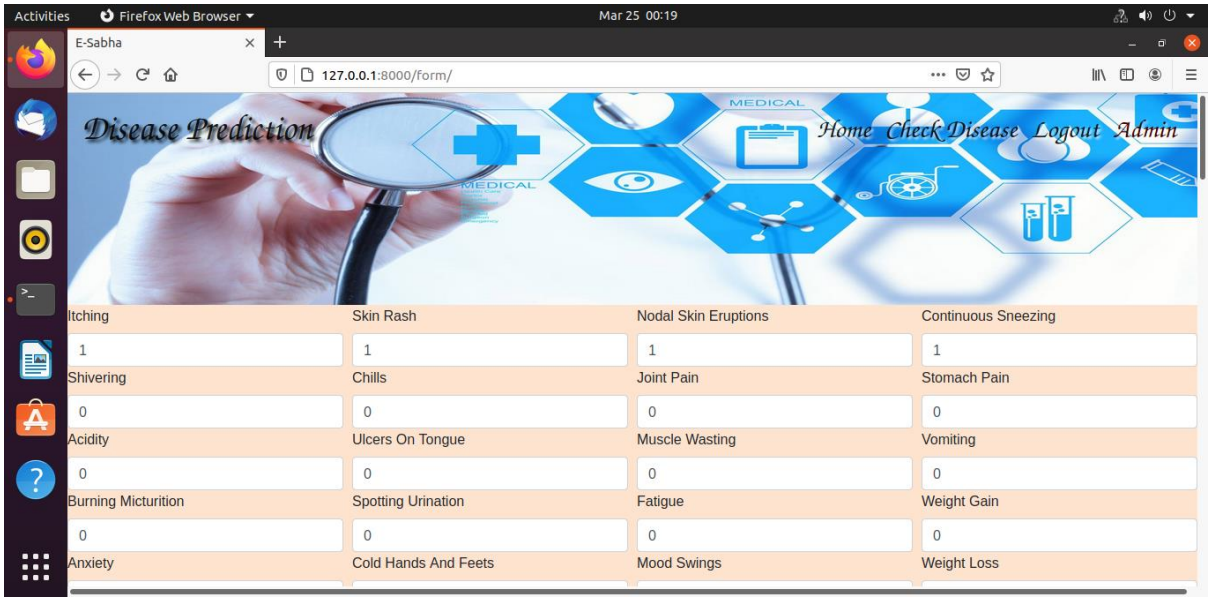
Login Page:



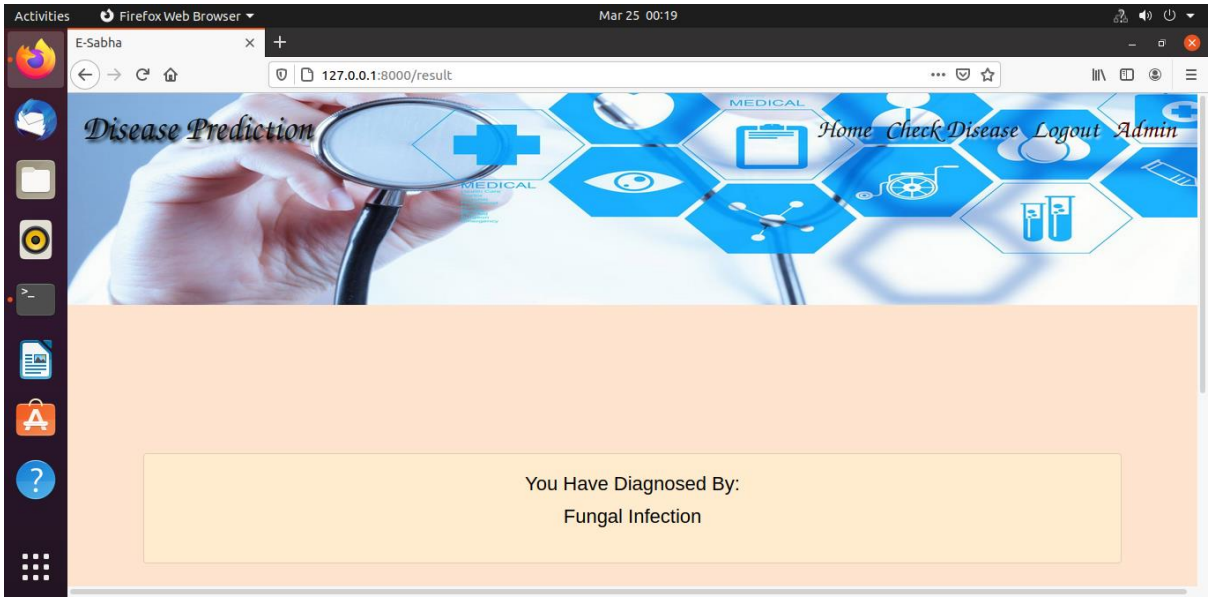
Admin Page:



Disease Prediction Form:



Result:



Chapter 5

Conclusion

This system has utilized deep learning capabilities to achieve human disease detection system. This system is based on a simple classification mechanism which exploits the feature extraction functionalities of ANN. For prediction finally, the model utilizes the fully connected layers. The system has achieved an overall 95% testing accuracy on publically accessible dataset.

It is concluded from accuracy that ANN is highly suitable for human disease detection. This system can be used in hospitals and medical centres for easily disease detection.

Chapter 6

Reference

- Disease Prediction using Machine Learning." (2019).Mr. Chala Beyene, Prof. Pooja Kamat, “Survey on Prediction and Analysis the Occurrence of Heart Disease Using Data Mining Techniques”, International Journal of Pure and Applied Mathematics, 2018.
- Balasubramanian, Satyabhama, and Balaji Subramanian. "Symptom based disease prediction in medical system by using K-means algorithm." International Journal of Advances in Computer Science and Technology 3.