AN IDEA DOCUMENTAION FOR ACM Code2Create 2018

PREDICT DISEASE AND THEIR SEVERITY BASED ON 'INPUT SYMPTOMS' USING MACHINE LEARNING

Submitted by,

TEAM - _pygeeks1991

Team Members,

1.**Nabarun Chatterjee**.

2.**Palash Sarkar**.

____ *INSTITUTE* ____

UNIVERSITY INSTITUTE OF TECHNOLOGY
THE UNIVERSITY OF BURDWAN
GOLAPBAG (NORTH), BURDWAN-713104.

TABLE OF CONTENTS

1.PROBLEM STATEMENT –

Predict disease and their severity based on ‘input symptoms’ using different machine learning models.

2.SOLUTION TO THE PROBLEM–

What we propose is an intelligent system that can predict the disease/problem based on the symptoms, a particular person is suffering from. Not only this, the system will also be able give the severity of the disease and alert the person to consult a real-time doctor or get professional help. The system was trained on a detailed and well-structured data-set specifically designed and maintained for this kind of work. We trained different machine learning models and took different approaches to increase the efficiency and accuracy of the system. The system is showing the most probable disease according to the symptoms inserted by the user. The interface is a website that will be cloud controlled so that we can access all information whenever and wherever we require.

3.UNIQUENESS OF OUR SOLUTION–

What we have done is very simple yet not explored by many. This is not a linear model that will take in the symptoms, match them with the diseases and give us the result, no. This Is a much more dynamic and non-linear model that can adapt to it's needs, that's why we have used machine learning and trained many models on the problem specifically, due to this the predictions are much faster and accurate.

4.CHALLENGES FACED–

In the beginning, we decided to use a NoSQL database(pref. MongoDB), for the storage and retrieval of various user-related data and our huge data-set for training our machine learning. In this particular case a 'document-clustering' architecture would have been the most efficient rather than tedious SQL processing, but python's latest 'Django Web-Framework', has stability issues with the 'MongoEngine API' and hence, we could not commit our document models to the actual 'mongoDatabase'. Also, Django's widgets system has been deprecated and does not override input field types (esp. 'Password Fields') for Django generated model forms , even DTL could not help us overcome this

problem. Also, we had decided to use a secure email connectivity in python using 'XOAUTH2' authentication, but in-the-end we had to drop it resulted in a lot of boiler-plate code. There was a classic overfitting and under-fitting problem with the non-linear models that we were training because of the amount of data present. SVM's simply would not give the desired result. Same with Randomforest , so we had to work with the advanced version of Gaussian Naïve Bayes which was a bit of a problem in the beginning.

5.TECHNOLOGY STACK AND SYSTEM MODULES–

1. Model View Pages/Front-End Web Interface: *HTML, Bootstrap.CSS and DTL (Django template language) for dynamic content generation and responsive web design, for our website.*

2.Middleware for connectivity between server and web pages and also for DOM programming: *JavaScript, JQuery for async server connectivity and page content manipulation.*

3.Mainstream programming language and Web framework: *Python3 and Django web-framework for restful web services and request handlers and developing various computational modules.*

4.Machine-Learning module: *Python3 and ScikitLearn for the ai model ("**decision trees/random forests**/* Gaussian Naïve Bayes*") for symptom analysis and disease prediction.*

5.SqlLite3 Database for fast JSON formatting: *SQL for storing data-sets, results and other user related information.*

6.Openshift Cloud Platform: *For hosting our local Django app on the openshift-c service platform.*

6.FUTURE WORK–

1. Integrate it with some natural processing toolkit to get the best out of this system and make it more dynamic. Like DialogFlow.

2. Real Time Language Translation targeted for the rural population.

3. Better security of the system when taken online.

4. Better UI and functionalities.

7.PROJECT LINKS–

GitHubLinks :

https://github.com/Tejas07PSK/ACM_c2c_DP/tree/master/c2cHack