**FUZZY DESIGN FOR HALTH CARE ANALYTICS**

**MAKAUT INTERNSHIP PROJECT REPORT**

**Submitted by**

**NAME ROLL NO**

**ABHISHEK ANAND 33200117052**

**ASHUTOSH KUMAR 33200117039**

**MD. KALIMUDDIN 33200117028**

**SOURAV KUMAR DEO 33200117009**

**SURAJ KUMAR 33200117003**

*Under the guidance of*

**Mr. Soumen Ghosh**

**Assistant Professor, Department of Computer Science and Engineering**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**TECHNO INTERNATIONAL BATANAGAR**

**Maheshtala, Kolkata – 700141, West Bengal, India**

**ACKNOWLEDGEMENT**

I am greatly indebted to our project Guide **Mr. SOUMEN GHOSH, Assistant Professor of Computer Science and Engineering Department** for providing us all possible help and support while doing this project. Without her guidance the project would not get such a progress.

I also express my sincere thanks to **Mr. INDRANIL SENGUPTA, Head of the Department, Computer Science and Engineering,** whose ceaseless cooperation made it possible in completion of this project for extending all possible help.

I would like to thank **PROF.(Dr.) ANKUR GANGULY, Principal, Techno International Batanagar** for his colossal support and encouragement.

Finally I would like to thank our Project Coordinator **Dr. SOMDATTA CHAKRABORTTY, MAKAUT** for giving us opportunity to utilize our time efficiently during lockdown.

Lastly, I would like to extend our thanks to all respected Faculty Members and friends directly or indirectly associated with our project, who contributed their personal level best which enabled us for a successful completion of the project.

**ABSTRACT**

The data collected about the patients in remote healthcare applications constitutes to big data because it varies with respect to volume, velocity, variety, veracity, and value. The purpose of this project is to process such a large collection of heterogeneous data. The project is based upon the initial cluster formation, retrieval, and processing of the big data in the cloud environment. We use the apriori algorithm and the association rule for data mining. This system is based upon Fuzzy Math Model Approach with Python 3.7.3 (latest version). This system is designed in Anaconda(Jupiter).

**TABLE OF CONTENTS**

**TOPICS PAGES**

**Chapter – 1 --- Introduction**  1

**Chapter – 2 --- Detail to be inserted**  2

**Chapter -- 2 --- Classification**  4

**Chapter -- 3 --- Clustering**  5

**Chapter – 4 --- Usecase diagram & Dataflow diagram** 6

**Chapter – 5 --- Apriori Algorithm**  9

**Chapter – 6 --- Asociation Rule** 11

**Chapter – 7 --- Result & Discussion (Graph)**  12

**Chapter – 8 --- Merit & Demerits** 13

**Chapter – 9 --- Future Scope** 14

**Chapter – 10 --- Conclusion** 15

**References**

**LIST OF FIGURES**

FIGURE NO. FIGURE NAME PAGE NO

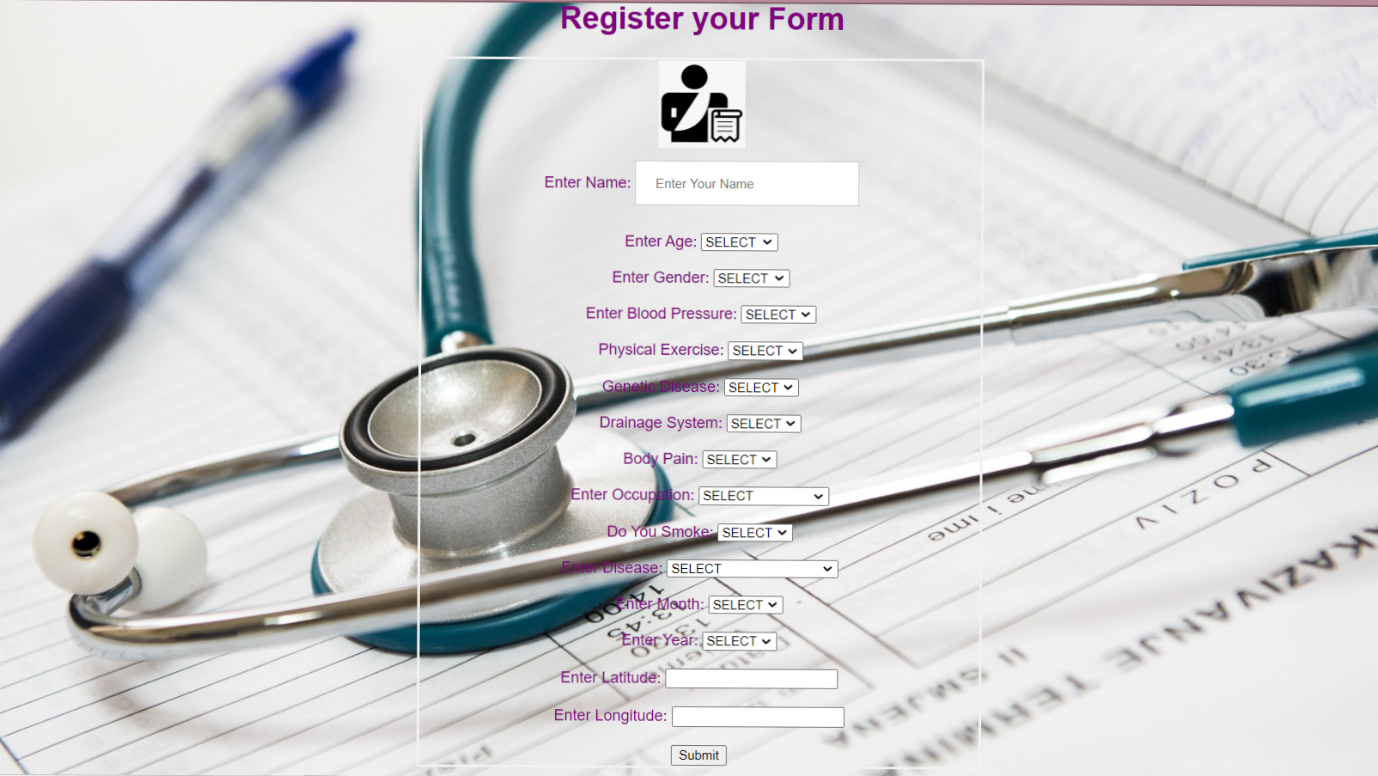
1. Use case Diagram 6
2. Data flow diagram (level 0) context level 7
3. DFD level 1 7
4. Level 2 data flow diagram 8
5. **INTRODUCTION**

**This project is aimed to provide people a hands-on check-up of a disease based on the symptoms they provide. For this project we have collected data from remote villages of West Bengal. The data ranges from the quality of water and food they consume to the drainage system of their area. We have focused on all the symptoms that a doctor asks from a patient.**

**At first a user have to register to our website. After a successful registration and a validated login the user fills a form which asks general information related to his age, gender, and other health related questions. This data is then stored in a CSV file and then processed through algorithms like Fuzzy logic, Apriori algorithm and Association Rule. After the complete processing a health report is generated and given to the user. This health report shows the probability and percentage of a disease which he/she may be suffering from.**

**From all the collected data we also calculated that what the common symptoms for a set of diseases are. And this result is not shared with the user and is only for the analysis by admin.**

1. **DETAILS TO BE INSERTED:**

****

* **Age:** There are four age group assigned user have to enter the age group from which they belong.
* **Gender:** The user will enter their gender here.
* **Blood Pressure:** The users have to enter the state of their blood pressure that whether it is high or low.
* **Physical Exercise:** The user has to enter the detail that whether they do physical exercise or not.
* **Genetic Disease:** Here the user has to update the detail that whether they have any genetic disease or not.
* **Drainage System:** Here the user will fill the detail that whether they have a drainage system or not.
* **Body Pain:** The user has to update that whether they are having body pain or not.
* **Occupation:** Here the user will have to update about their occupation.
* **Smoking Habit:** The user will update this block considering whether the smoke or not.
* **Disease:** Here a list of diseases are given in the drop down the user will have to enter the disease they are dealing with.
* **Month:** The user will enter the month here.
* **Year:** The user will enter the year here.
* **Latitude:** The user will enter the latitude here.
* **Longitude:** The user will enter the longitude here.

1. **CLASSIFICATION**

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known).

“How is the derived model presented?” The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks. A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily be converted to classification rules. A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. There are many other methods for constructing classification models, such as naïve

Bayesian classification, support vector machines, and k-nearest neighbour classification.

1. **CLUSTERING**

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression. Although classification is an effective means for distinguishing groups or classes of objects, it requires the often costly collection and labelling of a large set of training tuples or patterns, which the classifier uses to model each group. It is often more desirable to proceed in the reverse direction: First partition the set of data into groups based on data similarity (e.g., using clustering), and then assign labels to the relatively small number of groups. Additional advantages of such a clustering-based process are that it is adaptable to changes and helps single out useful features that distinguish different groups.

1. **USERCASE DIAGRAM & FLOW DIAGRAM**

LOG IN/REGISTER

ADMIN

Display Result (disease possibility)

USER

VERIFY PASSWORDD

<<include>>

<<extend>>

<<include>>

<<Include>>

APP

DATA FILLED

DISPLAY LOGIN ERROR

**USE CASE DIAGRAM**





1. **APRIORI ALGORITHM**

Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules. The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties. Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k+1)-itemsets. First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L1. Next, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of the database. To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, is used to reduce the search space.

**Apriori property:** All nonempty subsets of a frequent itemset must also be frequent.

“How is the Apriori property used in the algorithm?” To understand this, let us look at how Lk−1 is used to find Lk for k ≥ 2. A two-step process is followed, consisting of join and prune actions.

**1. The join step:** To find Lk, a set of candidate k-itemsets is generated by joining Lk−1 with itself. This set of candidates is denoted Ck. Let l1 and l2 be itemsets in Lk−1. The notation li[ j] refers to the jth item in li (e.g., l1[k−2] refers to the second to the last item in l1). By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. For the (k −1)-itemset, li, this means that the items are sorted such that li [1] < li[2] < ... < li [k − 1]. The join, Lk−1 on Lk−1, is performed, where members of Lk−1 are joinable if their first (k − 2) items are in common. That is, members l1 and l2 of Lk−1 are joined if (l1[1] = l2[1]) ∧ (l1[2] = l2[2]) ∧...∧(l1[k−2] = l2[k−2]) ∧(l1[k−1] < l2[k−1]). The condition l1[k−1] < l2[k−1]simply ensures that no duplicates are generated. The resulting itemset formed by joining l1 and l2 is l1[1], l1[2],..., l1[k −2], l1[k −1], l2[k −1].

**2. The prune step:** Ck is a superset of Lk, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included inCk. A scan of the database to determine the count of each candidate in Ck would result in the determination of Lk (i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to Lk). Ck, however, can be huge, and so this could involve heavy computation. To reduce the size of Ck, the Apriori property is used as follows. Any (k − 1)-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any (k − 1)-subset of a candidate k-itemset is not in Lk−1, then the candidate cannot be frequent either and so can be removed from Ck. This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets.



1. **ASSOCIATION RULE**

Once the frequent itemsets from transactions in a database D have been found,

it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence). This can be done using Equation :

confidence(A ⇒ B) = P(B|A) =

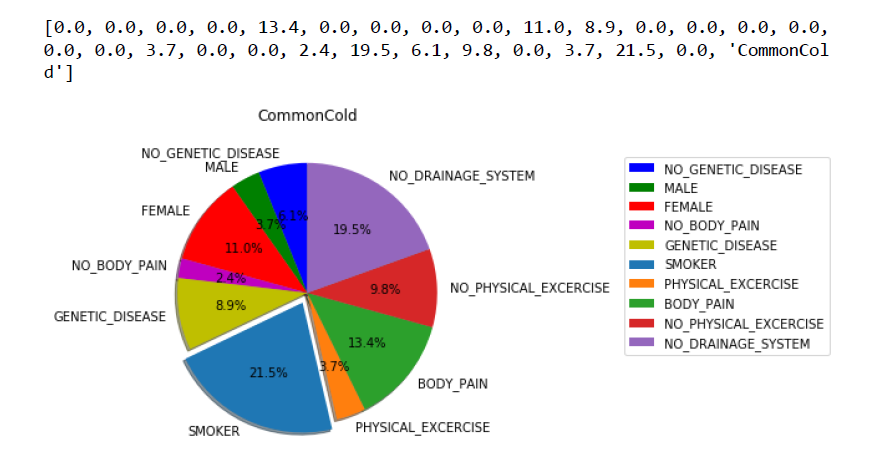
The conditional probability is expressed in terms of itemset support count, where support count(A∪B) is the number of transactions containing the item sets A∪B, and support count(A) is the number of transactions containing the itemset A. Based on this equation, association rules can be generated as follows:

* For each frequent itemset l, generate all nonempty subsets of l.
* For every nonempty subset s of l, output the rule “s ⇒ (l − s)” if ≥ min conf, where min conf is the minimum confidence threshold.

Because the rules are generated from frequent item sets, each one automatically satisfies minimum support. Frequent item sets can be stored ahead of time in hash tables along with their counts so that they can be accessed quickly.



1. **RESULT & DISCUSSION**

****

After all the operations has been done, we will get a pie-chart of the respective diseases that the person is diagnosed with. The pie-chart will so the percentage of symptoms, activities & circumstances etc. that are responsible for that particular disease.

1. **MERITS & DEMERITS**

**Merits-**

1. A user can check the possible disease he may be suffering from and he don’t need to visit a doctor.

**Demerits-**

1. This System gives only a probability of the disease and is not the final approved report. So the user must consult a doctor.

1. **FUTURE SCOPE**

We have planned to upgrade this website into an app.

1. **CONCLUSION**

This fuzzy system captures all the variations of the symptoms knowledgeable and so, it will be useful to infer the exact stage of patient risk as per the expert’s knowledge. This is essentially useful to take proper decision at the right time for giving treatment to the patients immediately. Such risk found to change the life style, food habits, have monitoring from the treatment in future.