International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Pregnancy Assistance using Ai-Based Application

Rijwan Khan, Vartika Singh, Sachin Singh

Abstract: ***With the onset of industrialization from the 18th century the family structure has drastically changed from the larger joint families to smaller nuclear families. The most affected people from this change are the ladies who are now left without any elderly guidance and support through pregnancy. We through our work aim to reduce this feeling and provide the basic help and care to these pregnant women. We are using a self-designed algorithm as a way to tackle this situation and help in providing proper guidance to all our users that are pregnant.***

Keywords: CART, Decision Tree, Set, pregnant

I. INTRODUCTION

We aim to develop a model that can assist a woman during their pregnancy. Our model will not only assist the woman but will also provide details about the important milestones during the pregnancy. We will be collecting various information from the woman through our app with the help of which we will be predicting the upcoming scenarios and providing assistance accordingly. The results would be calculated from a predictive model that would be built from the previous data that is available.

The assistance would be provided through an Application working as the frontend for all kinds of data collection and interaction with the user. This application will be supported with a backend data storage site that will store all the incoming data from the user. This user data would be used to predict important milestones of pregnancy and also give the answer to the user queries with the help of a prediction model stored over the cloud.

A. Decision Tree

Decision tree is a commonly used method for establishing classification systems based on multiple covariates or for prediction algorithms for a target variable. This method is used to classify a population into branch-like segments to construct an inverted tree with a root node, internal nodes, and leaf nodes. The algorithm does not require any parameters and can deal with large, complicated datasets without imposing a complicated parametric structure. When the sample size is large enough, study data can be divided into training and validation datasets. Using the training dataset to build a decision tree model and a validation dataset to decide on the appropriate tree size needed to achieve the optimal final model. This paper introduces frequently used algorithms used to develop decision trees (including CART) and describes the SPSS and SAS programs that can be used to visualize tree structure [12].

B. Random Forest

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction [8].

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speaks , the reason that the random forest model works so well because it is a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models[7, 8, 9].

The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for the random forest to perform well are [8] [2]

- There needs to be some actual signal in our features so that models built using those features do better than random guessing.
- The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

C. Dataset

Data obtained from the real world is not always in any certain order. Thus the real-world data is highly unstructured and without a proper relationship to define them all.

To handle the data we had many approaches that could have been used. We could have used the Decision tree, but we have chosen Set Theory and are using Sets to work with real-world data. These sets are known as Free-sets.

Revised Manuscript Received on March 30, 2020.
* Correspondence Author
Rijwan Khan*, Department of Computer Science and Engineering, ABES Institute of Technology, Ghaziabad, Affiliated to AKTU Lucknow, India. E-mail: rijwankhan786@gmail.com
Vartika Singh, Department of Computer Science and Engineering, ABES Institute of Technology, Ghaziabad, Affiliated to AKTU Lucknow, India. E-mail: vartika.singh2704@gmail.com
Sachin Singh, Department of Computer Science and Engineering, ABES Institute of Technology, Ghaziabad, Affiliated to AKTU Lucknow, India. E-mail: sachinsingh1056@gmail.com

They are used to extract the so-called frequent item sets (i.e., sets of items appearing in at least a given number of transactions) [3].

Hence the practical data set was not available anywhere that matched our need or is cleanable enough to be used hence we designed our own dataset depending on our needs.

Nomenclature

$\mathbf{P_i}$ The probability factor of occurrence of each set of symptoms
$\mathbf{O_i}$ The occurrence of D_i in the training dataset
n The number of a set of symptoms in the list
$\mathbf{\{D_1, D_2, ..., Dn\}}$ All the set of symptoms in the pregnant ladies
$\mathbf{D_k}$ New data entry in the model
$\mathbf{L_i}$ Unique element from all the sets of the list
V Selection parameter for each

II. WORKING IDEA

It will collect various information from the user at the time of first use that will include the height, weight, current medical condition and the medical history of the user. We will also take the monthly entry about the menstrual cycle to determine the menstrual health of the user. The data would be fed into the prediction model to help the user through our app.

III. PREDICTION MODEL

The nearest algorithm to be used in this case is the CART algorithm. There have been many attempts to improve the efficiency of this algorithm using ensemble learning methods like Boosting and Bagging. This helps in generating various classifiers and aggregating their results [1][4].

The CART algorithm uses a Decision Tree that builds the tree on the basis of calculated Information gain and Gini index. The attributes are represented in an order depending upon the values of Information gain. The Dataset in our case is unordered, i.e. There is no order in which symptoms occur in a woman's body [1][11].

Random Forest is also a good option for prediction with high accuracy but basically, it is also just a collection of multiple Regression trees hence again we are encountered with the same problem of unordered data as discussed with the CART algorithm[1][6][10].

Each woman has a different body and body structure and every pregnancy can be different and unique from the others. Also, the symptoms and diseases may not always occur in the same order in each woman. So, using the Decision tree algorithm to do our work would not be enough. We needed an algorithm to be able to represent all the possibilities of a pregnancy efficiently without any hierarchical relation in between the elements so we have chosen the "Set approach" over the decision tree method [3][6].

The prediction model here uses the Probability of the occurrence of a set of diseases in a pregnant lady. It will also use the set theory and the concept of bagging. The model will have two phases the learning phase in which the algorithm will learn from the test case data that is entered by the used during operation period and the prediction phase in which the app will provide the output depending upon the input of the user

Prevalence of self-reported pregnancy symptoms reported often or sometimes

Frequency	N = Valid responses	Often	Sometimes	Total prevalence
Urinary Frequency	209	52.20%	33%	85.20%
Tiredness	209	45.50%	41.50%	87%
Poor Sleep	211	27.50%	35.05%	62.55%
Back Pain	210	19.50%	40.50%	60.00%
Vaginal Discharge	205	17.60%	32.20%	49.80%
Forgetfulness	198	15.70%	39.90%	55.60%
Headache	200	14.50%	36%	50.50%
Vivid Dreams	201	13.90%	27.40%	41.30%
Taste Smell Changes	197	13.70%	20.30%	34.00%
Change in Nipples	196	13.30%	25%	38.80%
Nausea	207	12.60%	21.70%	34.30%
Change in Libido	197	11.20%	32%	43.20%
Hip Pelvic Pain	199	10.60%	23.10%	33.70%

Fig 1: Pregnancy symptoms and its occurrence [7]

A. Learning Phase

In this phase, the data would be used to train our model. During the Training Phase, the algorithm will use the training data to improve and refine the Probabilistic Factor of all the possible sets of symptoms. This phase will help and decide the real-life occurring probability of any set of symptoms.

B. Prediction Phase

In this phase, the model will predict the possible future complications that may occur during pregnancy using the real-time data provided by the user on our app. The data can be stored in a real time access cloud so that the data can be accessed anytime anywhere.

C. Probabilistic Approach

Here, the probability value is being calculated for all the possible sets of symptoms. This value is used to check the real-life occurrence of that set. Whenever new data is being encountered, the overall probability for all the sets is updated automatically. The update of the data is such that the following condition is always true-

$$\sum_{i=1}^{n} P_i = 1$$

Where,
n= Number of sets
P_i= i^{th} set of symptoms

Thus the rule of probability is held that the sum of probabilities for all possible outcomes is equal to one.

Explanation-

Consider the following sets of symptoms-

L= [{ D_1 , D_2 } , { D_2 , D_3 } , { D_1 , D_2 } , { D_2 , D_1 , D_3 } , { D_2 , D_3 } , {D_3}]

$P_1(D_1 , D_2)=\frac{2}{6}=\frac{1}{3}$; $P_2(D_2 , D_3)=\frac{2}{6}=\frac{1}{3}$

$P_3(D_1 , D_2 , D_3)=\frac{1}{6}$; $P_4(D_3)=\frac{1}{6}$

[P1+P2+P3+P4=1]

Now, a new data is encountered i.e.,{ D_2 , D_3 }, then the new probabilities calculated are-

$P_1(D_1 , D_2)=\frac{2}{7}$; $P_2(D_2 , D_3)=\frac{3}{7}$

$P_3(D_1 , D_2 , D_3)=\frac{1}{7}$; $P_4(D_3)=\frac{1}{7}$

[P1+P2+P3+P4=1]

So, this approach helps in proper assessment and analysis of our raw data.

D. Algorithm

This algorithm is using the set approach to bag the symptoms that occur together with each other. These symptoms are bagged together because they all occur together or one symptom causes the other with it. Once the sets are being formed, the algorithm will find the occurrence of each set of symptoms using the probabilistic occurrence of that symptom in the training phase.

- *For the probability factor:* Probability factor P_i will be calculated for the occurrence of each set of symptoms in the pregnant ladies. This factor will further help us in determining the probability of occurrence of those set in the real world and help in accessing the pregnant women accordingly.

1) While Training the model

Step 1:- Calculate the probability factor P_i for all the sets of symptoms *{D_1, D_2,....., Dn}* where Dn represents a set of symptoms. Save O_i as the occurrence of D_i in the training dataset.

Step 2:- Use the following formula for calculation of the probability factor:-

$$P_i = \frac{Total\ occurence\ of\ Di\ in\ the\ training\ data\ set}{n}$$

Where,

n= Total number of possible sets of symptoms, and

$$\sum_{i=1}^{n} P_i = 1$$

2) While providing assistance

Step 1:- For a set of symptoms D_k received, *Count=0*, check-

Step 2:- For *i=1* to *i<=n*

Step 3:-{

If the set of new symptoms matches with the set of diseases D_i i.e.

$$D_i \cap D_k = D_i \ and\ D_k$$

Then update the probability factor of *i* as-

$$P_i = \frac{(O_i + 1)}{n+1} \quad \textbf{-(1)}$$

$O_i = O_i + 1$

$i = i + 1$

$Count = Count + 1$

Step 4:- Else if the set of symptoms does not match with the set of symptoms D_i i.e.

$$D_i \cap D_k \neq D_i \ and\ D_k$$

Then update the probability factor of i as-

$$P_i = \frac{(O_i)}{n+1} \quad \textbf{-(2)}$$

$i = i + 1$

}

Step 5:- If *count = 0* , then add this set as a new set of diseases into the existing list of diseases. New list of dataset is *{D_1,D_2,.....,Dn+1}* and update $n=n+1$

- *For the prediction*

1) Analysis

Step 1:- For each new entry, D_k calculates the weight of the occurrence of the set.

Step 2:- For all *i=1* to *i<=n*

Step 3:- {

Calculate the following for D_i and D_k:-

$C_i = P_i * N_i$

Where,

P_i = Probability weight of the occurrence

$$N_i = count\ (D_i \cap L_i)$$

}

Step 4:- Set a selection value V

Step 5:- Select all the values of D_i where,

$$C_i \geq V$$

IV. RESULT

Case 1: *Only one set of symptoms is selected*

In this case, there is only one possibility of the occurrence of that set of symptoms. So, we will provide the solution and measures that should be taken to prevent that from occurring in the future.

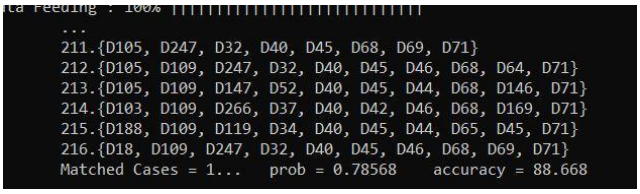

Fig. 2. Accuracy result of case 1

Case 2: *More than one sets of symptoms are selected*

In this case, the prediction will be made using the extra sets obtained after the completion of the algorithm. We will provide assistance using those sets obtained and predict the symptoms that are more likely to occur in the future.

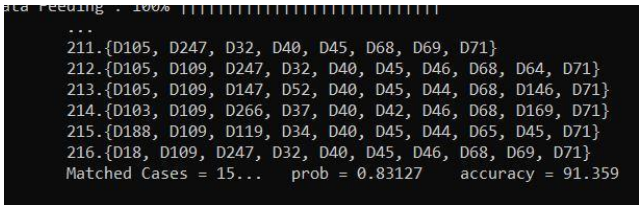

Fig. 3. Accuracy result of case 2

V. CONCLUSION

This algorithm can be used in any real life scenario that requires probabilistic prediction. It can be successfully used to reflect real life conditions such as disease predictions with the help of the user symptoms, predicting the natural disaster with the help of the readings from the various systems etc.

This algorithm is currently being used at a small level on our mobile application but in the future it can be used at a larger scale at hospitals and clinics for better understanding and tracking of health of pregnant ladies.

REFERENCES

1. Ali, Jehad, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. "Random forests and decision trees." International Journal of Computer Science Issues (IJCSI) 9, no. 5 (2012): 272.
2. Archer, Kellie J., and Ryan V. Kimes. "Empirical characterization of random forest variable importance measures." Computational Statistics & Data Analysis 52, no. 4 (2008): 2249-2260.
3. Boulicaut, Jean-François, Artur Bykowski, and Christophe Rigotti. "Free-sets: a condensed representation of boolean data for the approximation of frequency queries." Data Mining and Knowledge Discovery 7, no. 1 (2003): 5-22.
4. Breiman, L., 2017. Classification and regression trees. Routledge
5. Chen, Jianguo, Kenli Li, Zhuo Tang, Kashif Bilal, Shui Yu, Chuliang Weng, and Keqin Li. "A parallel random forest algorithm for big data in a spark cloud computing environment." IEEE Transactions on Parallel and Distributed Systems 28, no. 4 (2016): 919-933.
6. De'ath, Glenn, and Katharina E. Fabricius. "Classification and regression trees: a powerful yet simple technique for ecological data analysis." Ecology 81, no. 11 (2000): 3178-3192.
7. Foxcroft, K.F., Callaway, L.K., Byrne, N.M. et al. Development and validation of a pregnancy symptoms inventory. BMC Pregnancy Childbirth 13, 3 (2013) doi:10.1186/1471-2393-13-3.
8. Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2, no. 3 (2002): 18-22.
9. Loh, Wei-Yin. "Classification and regression trees." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1, no. 1 (2011): 14-23.
10. Moisen, G. G. "Classification and regression trees." In: Jørgensen, Sven Erik; Fath, Brian D.(Editor-in-Chief). Encyclopedia of Ecology, volume 1. Oxford, UK: Elsevier. p. 582-588. (2008): 582-588.
11. Patel, Bhaskar N., Satish G. Prajapati, and Kamaljit I. Lakhtaria. "Efficient classification of data using decision tree." Bonfring International Journal of Data Mining 2.1 (2012): 06-12.
12. Song, Yan-Yan, and L. U. Ying. "Decision tree methods: applications for classification and prediction." Shanghai archives of psychiatry 27.2 (2015): 130.

AUTHORS PROFILE

Dr. Rijwan Khan is B.Tech (CSE), M.Tech (CSE) and Ph.D in Computer Engineering. He has a total of 13 years of teaching experience. Now working as a Head of Department in ABES Institute of Technology. His areas of research are software testing, soft computing, and nature inspired algorithms. He published more than 25 research papers in different journals. He is a reviewer of more than 10 different Scopus and SCI indexed journals like journals of Inderscience, Springer, and Elsevier etc. He is the author of 3 books on C Programming, Data Structure using C, and Operating System and one book chapter in springer book.

Vartika Singh is an undergraduate student from the department of Computer science and Engineering. She is currently a student of ABES Institute of Technology, Ghaziabad. She is enthusiastic about Data Structure and has done multiple projects on Web Development. She has also done work in Machine learning and Application development.

Sachin Singh is an undergraduate Computer Science & Engineering student pursuing B.Tech at ABES Institute of Technology, Ghaziabad. His areas of interest are Machine Learning, Deep Learning and Data Science. He has done training in Machine Learning and Python. He is an core member of the Developer Students Club and is proficient in programming in python