ORIGINAL RESEARCH PAPER

Computer Science

VITAMIN D DEFICIENCY PREDICTION

KEY WORDS:

B Deekshitha B Tech IV[th] Year Project Guide Anurag University, Hyderabad.

Shree Varsha* B Tech IV[th] Year Project Guide Anurag University, Hyderabad. *Corresponding Author

Sakilam Varsha B Tech IV[th] Year Project Guide Anurag University, Hyderabad.

YV Reddy Project Guide Anurag University, Hyderabad.

ABSTRACT

Vitamin D Deficiency (VDD) is one of the most significant global health problem and there is a strong demand for the prediction of its severity. The independent parameters like age, sex, weight, height, body mass index (BMI), waist circumference, body fat, bone mass, exercise, sunlight exposure, and milk consumption were used for prediction of VDD. Factors such as lack of sunlight exposure, low physical activity, poor dietary habits, lack of sleep and stress increase the risk of the impacts due to Vitamin D Deficiency. There are certain bad habits that increase the risk of VDD . This project aims at predicting the occurrence of VDD using Gaussian Naive Bayes classifier and Random Forest Prediction classifier.

INTRODUCTION

The project comprises of four modules. First module deals with the storing the patient symptoms and habits data in the DB, the second module deals with the creation of comma seperated value files of the data stored in the DB. The third module deals with the development of python routines to execute the Gaussian Naive Bayes classifier on the data set and the final module deals with the Random Forest Classification.

The project uses the 'sklearn' module of Python to perform the gaussian Naive Bayes and Random Forest Classifications. A random Forest is a collection of Random Forests. In a Random Forest, at each level a decision is made on the criteria to be followed for making a node as root note at that level. In random forest, gini index formula is used to arrive at the criteria for the root node. The Gini coefficient sometimes called the Gini index or Gini ratio, is a statistical measure calculated by using a formula, involving the number of occurrences of different values of the attributes in the gievn data set.**Naive Bayes** is a classification algorithm of Machine Learning based on Bayes theorem which gives the likelihood of occurrence of the event. Naive Bayes classifier is a probabilistic classifier which means that given an input, it predicts the probability of the input being classified for all the classes. It is also called conditional probability. Gaussian Naive Bayes is Used when we are dealing with continuous data and uses Gaussian distribution, which is also known as normal distribution of a continuous variable, which is usually in the form of a bell shape.

Tools Used: slkearn's GNB classifier & RF classifier modules.

Algorithms Used: Gaussian Naive Bayes Classification and Random Forest classification

ADVANTAGES

The project is useful in understanding about the habits leading to Vitamin D Deficiency. The project is useful to the data analysts to understand more about the Gaussian Naive Bayes and Random Forest classifications. This project finally leads to the improvement of quality of the people lives.

Technical advantages:

Using latest Python's sklearn tool to implement the Gaussian Naive Bayes and Random Forest classifications.Using Python, which is chosen as the best programming language, by the Programming Community.More Functionality can be implemented with less no.of lines of code in Python. PyQt tool is used to create the Graphical User interfaces. All the Front end code is generated automatically by PyUIC.

Feasibility Analysis

The project being developed is economic with respect to peoples point of view. It is cost effective in the sense that has eliminated the paper work such as entry of the patient details completely. The project is also time effective because doctors can be able to identify the nerve easily. The result obtained contains minimum errors and most of the noise is removed. The technical requirement for the project is economic and it does not use any other additional Hardware and software.

Functional Requirements

A Functional Requirement (FR) is a description of the service that the software must offer. It describes the software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be calculation, data manipulation, business process, user interaction or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional specification.

User can perform below operations:

1. Get report
2. Type Symptoms
3. Submit
4. Know the Prediction
5. Close

Non Functional Requirements

Reliability: Software Reliability means operational reliability. It is described as the ability of a system or component to perform its required functions under static conditions for a specific period. Software reliability is also defined as the probability that a software system fulfills its assigned task in a given environment for a predefined number of input cases, assuming that the hardware and the input are free 5 of error. Software reliability is an essential connect of software quality, composed with functionality, usability, performance, serviceability, capability, maintainability and documentation.

Scalability: Scalability is often a sign of stability and competitiveness, as it means the network, system, software or organization is ready to handle the influx of demand, increased productivity, trends, changing needs and even presence or introduction of new competitors. It is described as scalable as it has an advantage because it is more adaptable to the changing needs or demands of its users or clients.

Security: Software security is an idea implemented to protect software against malicious attack and other hacker risks so that the software continues to function correctly under such

potential risks. Security is necessary to provide integrity, authentication and availability.

Usability: Usability includes methods of measuring usability, such as needs analysis and the study of the principles behind an object's perceived efficiency or elegance. In human-application interaction, usability studies the elegance and clarity with which the interaction with a web site (web usability) is designed. Usability considers user satisfaction and utility as quality components, and aims to improve user experience through iterative design.

ALGORITHMS

In statistics, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features (see Bayes classifier). They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve higher accuracy levels.

Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the statistics and computer science literature, naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naïve Bayes is not (necessarily) a Bayesian method.

GUSSIAN NAÏVE ALGORITHM: When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. Another common technique for handling continuous values is to use binning to discretize the feature values, to obtain a new set of Bernoulli-distributed features; some literature in fact suggests that this is necessary to apply naive Bayes, but it is not, and the discretization may throw away discriminative information. Sometimes the distribution of class-conditional marginal densities is far from normal. In these cases, kernel density estimation can be used for a more realistic estimate of the marginal densities of each class. This method, which was introduced by John and Langley, can boost the accuracy of the classifier considerably.

RANDOM FOREST: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered"Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.).[12] The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho and later independently by Amit and Geman in order to construct a collection of decision trees with controlled variance. Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

The above procedure describes the original bagging algorithm for trees. Random forests differ in only one way from this general scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called "feature bagging".The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the B trees, causing them to become correlated. An analysis of how bagging and random subspace projection contribute to accuracy gains under different conditions is given by Ho.

Typically, for a classification problem with p features, $\sqrt{p}$ (rounded down) features are used in each split. For regression problems the inventors recommend p/3 (rounded down) with a minimum node size of 5 as the default.[3]:592 In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameters.

ExtraTrees: Adding one further step of randomization yields extremely randomized trees, or ExtraTrees. While similar to ordinary random forests in that they are an ensemble of individual trees, there are two main differences: first, each tree is trained using the whole learning sample (rather than a bootstrap sample), and second, the top-down splitting in the tree learner is randomized. Instead of computing the locally optimal cut-point for each feature under consideration (based on, e.g., information gain or the Gini impurity), a random cut-point is selected. This value is selected from a uniform distribution within the feature's empirical range (in the tree's training set). Then, of all the randomly generated splits, the split that yields the highest score is chosen to split the node. Similar to ordinary random forests, the number of randomly selected features to be considered at each node can be

IMPLEMENTATION

Sypder: Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software.[3][4] It is released under the MIT license.[5]

Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.

Tkinker: Tk is a platform-independent GUI framework developed for Tcl. From a Tcl shell (tclsh), Tk may be invoked using the command package require Tk. The program wish (WIndowing SHell) provides a way to bring up a tclsh shell in a graphical window as well as providing Tk.