International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 02 | February - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

# Predicting Heart Disease and Diabetes

S. Arockia Nisha, P.G, Student,

*Department of Computer Science and Engineering, Thoothukudi*

S. Abarna, Assistant Professor,

*Department of Computer Science and Engineering, Thoothukudi*

*Address*

mailto:arockianiaha18@gmail.com

mailto:abarna.kichi@gmail.com

***Abstract*— The healthcare industry is no exception. Machine Learning can play an essential role in predicting presence/absence of Locomotors disorders, Heart diseases and diabetese and more. Such information, if predicted well in advance, can provide important insights to doctors who can then adapt their diagnosis and treatment per patient basis. And in addition, we also predicting the diabetes diseases the machine learning. Algorithms likeLogistic Regression, Random Forest, Support Vector Machine, Naïve Bayes and etc., are used to build a machine learning model. Here discusses and compares the various data analytics techniques available for the heart diseases and diabetese prediction. However, the selection of the appropriate algorithm from the pool of available algorithms imposes challenge to the researchers with respect to the chosen heart diseases and diabetese. The accuracy of training model should be higher and error rate should be minimum.**

***Keywords*— Heart Disease, Diabetes, Machine Learning.**

## I. INTRODUCTION

In this study, a tentative design of a heart disease and diabetese prediction system had been proposed to detect impending heart disease and diabetes using Machine Learning Techniques. For the accurate detection of the heart disease and diabetese, an efficient machine learning technique should be used which had been derived from a distinctive analysis among several machine learning algorithms. Diagnosis of heart is done by prediction of it.

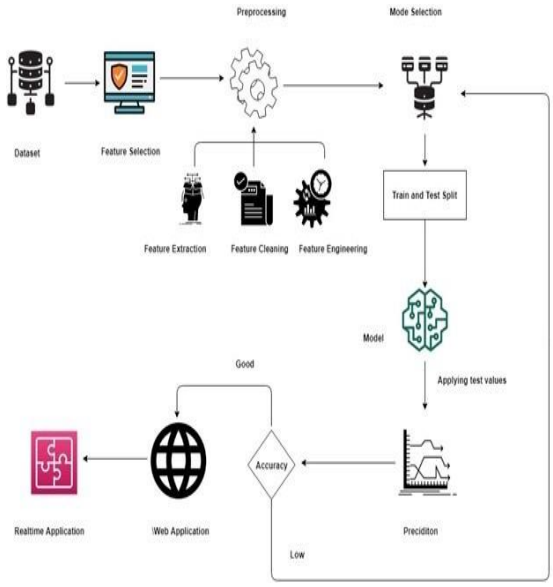
## II. DATA COLLECTION AND PROCESSING

Data collection is the process of collecting every detail about the heart diseases and diabetes like symptoms and the rate of possible counts. After collecting the every good data set we have to prepare the data set separately for everything. It will increase the accuracy rate as well as the perfect model. Preprocessing is the process of data we have to prepare our data set according to the symptoms.

We have to prepare two datasets for the model. Because preparing the data with possibilities at the same time is not a preparing the data's that can be understandable by machine. Preprocessing of selecting the features that will affect our model and which won't change in output. After the selection of the Features we will use that data to train our model.

## 2. DESIGN IMPLEMENTATION AND TESTING

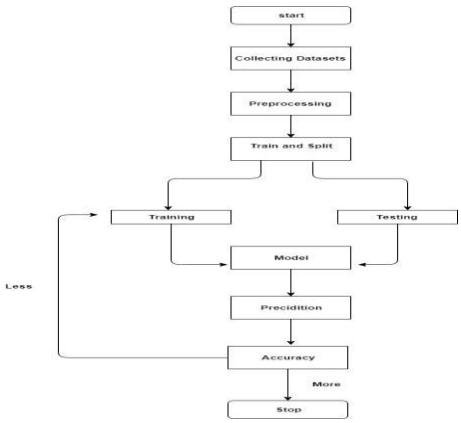
### *A. Architecture Design*




.

After the creation of model for heart and diabetes. In this we will split the data set into train and test. we will apply train set for training and test for predicting. After the prediction find the accuracy for every algorithm and finalize the perfect model. We need the web application to see the prediction results. But the machine learning and Web development is different domain. We are going to create the pipeline for interacting machine learning and machine learning model using the pickle package. The pickle package will store the machine learning model in the stage of prediction. After that user can give input and can get the output result.

*B. Flow Diagram*




*C. Module Building*

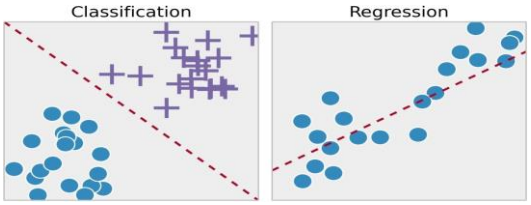
There is a very wide range of machine learning algorithms to choose from, most of which are available in the python library Scikit-learn. However, most of the implementations of these algorithms do not accept sparse matrices as inputs, and since we have a large number of nominal features coming from our n-grams features it is imperative that we encode our features in a sparse matrix. Out of the algorithms that do support sparse matrices in Scikit-learn, I ended up trying naive Bayes, logistic regression and support vector machine (SVM) with a linear kernel. I got the best results in cross validation using SVM with aneuclidean regularization coefficient of 0.1.

- **Supervised learning**
- **Unsupervised learning**
- **Semi Supervised learning**
- **Reinforcement**

.

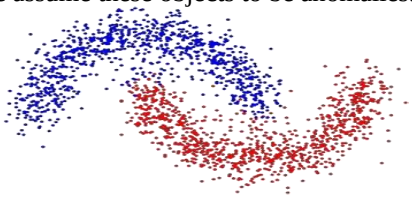
*1)Supervised learning*

Supervised learning is the task of inferring a function from labeled training data. By fitting to the labeled training set, we want to find the most optimal model parameters to predict unknown labels on other objects (test set). If the label is a real number, we call the task regression. If the label is from the limited number of values, where these values are unordered, then it's classification.



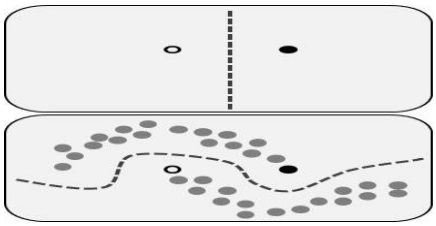

*2)UnSupervised learning*

In unsupervised learning we have less information about objects, in particular, the train set is unlabeled. What is our goal now? It's possible to observe some similarities between groups of objects and include them in appropriate clusters. Some objects can differ hugely from all clusters, in this way we assume these objects to be anomalies.



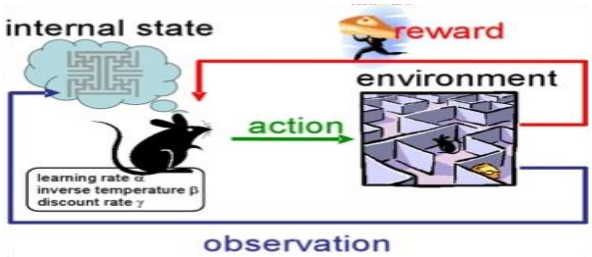
*3)Semi-Supervised learning*

Semi-supervised learning tasks include both problems we described earlier: they use labeled and unlabeled data. That is a great opportunity for those who can't afford labeling their data. The method allows us to significantly improve accuracy, because we can use unlabeled data in the train set with a small amount of labeled data.



*4)Reinforcement learning*

Reinforcement learning is not like any of our previous tasks because we don't have labeled or unlabeled datasets here. RL is an area of machine learning concerned with how software agents ought to take actions in some environment to maximize some notion of cumulative reward.




Imagine, you're a robot in some strange place, you can perform the activities and get rewards from the environment

for them. After each action your behavior is getting more complex and cleverer, so you are training to behave the most effective way on each step. In biology, this is called adaptation to natural environment.

*D. Naïve Bayes*

Naive Bayes is based on two assumptions. Firstly, all features in an entrance that needs to be classify are causative evenly in the decision (equally important). Secondly, all attributes are statistically self-determining, meaning that, knowing an attribute's value does not indicate whatever thing about other attributes' values which is not always true in practice. The process of classifying an instance is done by applying the Bayes rule for each class given the occurrence. In the fraud detection task, the following formula is calculated for each of the two classes (fraudulent and legitimate) and the class associated with the higher prospect is the predicted class for the instance.

*E. Support-vector Machine*

Figures In machine learning, support-vector machines (SVM's, also support-vector networks are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data is unlabeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups.The support-vector clustering algorithm, created by HavaSiegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

*F. 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 and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. The first algorithm for random decision forests was created 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 (as of 2019, owned by Minitab, Inc.). 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.

*G. System Testing*

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

*1)Unit Testing*

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

*2)Integration Testing*

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

*3)White Box Testing*

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

*4)Black Box Testing*

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box that can not be seen. The test provides inputs and responds to outputs without considering how the software works.

*5)Functional Testing*

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : Identified classes of valid input must be accepted. Invalid Input : Identified classes of invalid input must be rejected. Functions : Identified functions must be exercised.

Output :Identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

## III. CONCLUSIONS

The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. This project resolved the feature successfully and predicts the heart disease, with 85% accuracy. The model used was Logistic Regression. Further for its enhancement, we can train on models and predict the types of cardiovascular diseases providing recommendations to the users, and also use more enhanced models.

## REFERENCES

[1] Katarya, Rahul, and Polipireddy Srinivas. "Predicting Heart Disease at Early Stages using Machine Learning: A Survey." 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE, 2020.

[2] Gavhane, Aditi, et al. "Prediction of heart disease using machine learning." 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2018.

[3] Kohli, Pahulpreet Singh, and Shriya Arora. "Application of machine learning in disease prediction." 2018 4th International conference on computing communication and automation (ICCCA). IEEE, 2018.

[4] Krishnan, Santhana, and S. Geetha. "Prediction of Heart Disease Using Machine Learning Algorithms." 2019 1st international conference on innovations in information and communication technology (ICIICT). IEEE, 2019.

[5] Atallah, Rahma, and Amjed Al-Mousa. "Heart Disease Detection Using Machine Learning Majority Voting Ensemble Method." 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS). IEEE, 2019.