**Heart Failure Prediction in Big Data**

Heart diseases are one of the most challenging problems faced by the Health Care sectors all over the world. These diseases are very basic now a days. With the expanding count of deaths because of heart illnesses, the necessity to build up a system to foresee heart ailments precisely. The work in this paper focuses on finding the best way for identification of heart diseases. WE used Random Forest for the prediction of heart disease by making the use of dataset provided by Kaggle. We utilized various characteristics which relate with this heart diseases well, to find the better algorithm for prediction. The result of this study indicates that the Random Forest algorithm is one of the efficient algorithms for prediction of heart disease.

*Keywords: Random Forest, Heart Disease Prediction, Kaggle.*

**INTRODUCTION**

Heart disease or cardiovascular disease is a condition which involves the narrowing or the blockage of blood vessels in the heart which cause problems or failures in the human cardiovascular system. It causes many abnormal medical conditions like Hypertension, Cardiac Arrest, Arrhythmia, Stroke and Heart failure to name a few. It is one of the most challenging problems for the health care sector because of the complexity involved in the detection, diagnosis, and treatment of this condition. CVD's cause 17.9 million deaths worldwide every year which is approximately equal to 31% of all deaths in the world according to WHO. The occurrence heart disease in a person depends on a wide variety of factors like Age, Sex, Chest Pain Type, Resting Blood Pressure etc. Because of this, it may be a bit difficult to find the correct cause of the disease and to predict whether the person is prone to the disease or not, based on the conditions mentioned above. This can be of great help in the prediction of heart disease when certain data about a person is given. The work proposed in this paper focus mainly on various data mining practices that are employed in heart disease prediction. We used Kaggle and PySpark for obtaining and analysing the data respectively, which is helpful in increasing the accuracy of this algorithms.

**SPARK FRAMEWORK:** Spark is an open-source cluster computing framework used to increase the speed of computing and data processing for huge datasets, which is commonly known as Big Data. It is a computational engine, fundamentally, that works with very large amount of data by processing them using bath and parallel system processing mechanisms. It is an extended version of Hadoop Map Reduce and is useful in a larger number of types of computations, which includes interactive queries and stream processing. SQL queries, Streaming data, Machine learning (ML), and Graph algorithms are some of it is application fields. APIs in Java, Scala, Python and R are also provided by it which helps users proficient in different coding languages. In-memory cluster computing, which is the primary feature of spark, increases the processing speed of an application.

**PYSPARK:** It is the Python API to support Apache Spark. It is widely used by data scientists to work on Resilient Distributed Datasets (RDD). As Spark is written in SCALA and some programmers find it difficult to use it by coding in SCALA. PySpark helps programmers to use python instead of SCALA, and as python contains many different libraries, it is easier to work on huge datasets. PySpark links Python API to the spark core and initializes the Spark context.

**Random Forest Algorithm:** It is a type of Classification Algorithm comprising of numerous choices of trees. It utilizes the bagging and feature randomness when fabricating every individual tree to attempt to make an uncorrelated collection of trees whose forecast by advisory group is more precise than that of any single tree.

**METHODOLOGY**

**System Design:**

* Data Collection
* Data Understanding
* Data Assembly and Transformation
* Splitting of data
* Model Classification and Evaluation.

Data Collection: The data is collected from Kaggle. In this project the top five rows are shown below, and the target feature is used as our output, which is either 1 or 0. The result is calculated based on inputs we are taking from the patient.



Data Understanding: In this method we are going to understand the data type of each feature that we have collected from the patient.

Text

Description automatically generated

Data Assembly and Transformation: We are importing a library to represent all the data types into vector representation.

from pyspark.ml.feature import VectorAssembler

All columns except the target need to be converted to a vector, which we call it features.

We can see below that the features column is a vector based on all thirteen columns

Table

Description automatically generated

Splitting of Data: To train our model, we combine “features” and “target” as input/output.

final\_data = output.select(“features”, “target”)

Then, we are splitting final\_data to train and test as follows:

train, test = final\_data.randomSplit([0.7, 0.3])

Model Classification and Evaluation:

Random Forest algorithm is used as our classification method, and labelCol needs to be referred to as the target, and featuresCol is defined features, which already represented as vectors.

rf = RandomForestClassifier(featuresCol = ‘features’, labelCol = ‘label’)

In the next step, we train our algorithm, which is very similar to scikit, and see what is predicted output for our test case.

rfModel = rf.fit(trainData)  
 predictions = rfModel.transform(testData)  
 predictions.select('rawPrediction', 'prediction', 'probability').show(10)  
 predictions.select("label", "prediction").show(10)

Below is the sample output:

Accuracy = 0.8216525924047651  
Test Error = 0.17834740759523493  
[[27. 6.]  
[11. 50.]]  
0.8191489361702128  
0.1812949868408536  
0.8288473844184929

0.8216525924047651

rawPrediction|prediction| probability|  
+--------------------+----------+--------------------+  
|[2.57852702089555...| 1.0|[0.12892635104477...|  
|[1.79541074493214...| 1.0|[0.08977053724660...|  
|[5.60791601590022...| 1.0|[0.28039580079501...|  
|[6.59808423640259...| 1.0|[0.32990421182012...|  
|[7.74017929619157...| 1.0|[0.38700896480957...|  
|[18.5930938801606...| 0.0|[0.92965469400803...|  
|[3.76776326803094...| 1.0|[0.18838816340154...|  
|[8.68977952941367...| 1.0|[0.43448897647068...|  
|[2.19852095453323...| 1.0|[0.10992604772666...|  
|[1.80675307218490...| 1.0|[0.09033765360924...|  
+--------------------+----------+--------------------+  
only showing top 10 row

**CONCLUSION**

We have been identified that the Random Forest algorithm is one of the efficient algorithms in the prediction of heart diseases. Its accuracy was found to be 76.24%. In the future, this work can be upgraded by building up a web application based on the Random Forest algorithm and using a bigger dataset when contrasted with the one utilized in this examination. This will help in giving bigger dataset when contrasted with the one utilized in this examination. This will help in giving better outcomes and help healthcare experts in the prediction of coronary illnesses adequately and productively.

REFERENCES

1. Hosseini, S. (2020, September 4). *UCI Heart Disease Classification with pyspark*. Medium.

2. Alarsan, F. I., & Younes, M. (2019, August 31). *Analysis and classification of heart diseases using heartbeat features and Machine Learning Algorithms - Journal of Big Data*. SpringerOpen.