Stroke Prediction Analysis Using Machine Learning Approach

Goutham Jagadeesha, MSc Big Data Analytics and Artificial Intelligence, L00157174

***Abstract*— Stroke is a brain attack which causes your body to become paralysed. It is a very common problem throughout the world. Every one in six men and every one in five women is at risk for developing stroke. It is the second highest cause of death and highest cause of disability in the world. What happens in a stroke? In about 80 percent of cases stroke is due to blockage of blood vessel and brain due to blood clot. In 20 percent of cases, stroke is due to rupture of blood vessel and brain. In both cases, parts of affected brain suddenly stopped working. What causes stroke? There are many causes, most important causes include high blood pressure, especially long standing and uncontrolled high blood pressure, diabetes, high cholesterol and heart diseases are also an important risk factor. Another important risk factor is tobacco in the form of cigarette, shisha, or even chewing tobacco. These factors over a period of time affect blood vessels of brain, ending up another blockage or rupture of them. We have used different machine learning models to predict stroke by using a open source data set available in Kaggle. We have used Logistic Regression, Decision Tree, Random forest and Gradient Boosted Tree algorithm to train our model out of which Gradient Boosted Tree was the best performing model as we have achieved about 80 percent accuracy which is close to 3 percent higher than our second best performing model which is Logistic Regression. Data visualization and model comparison have been made in this project.**

Git Hub Link: BDA Technical Project L00157174

***Index Terms*— Stroke, Machine Learning, models, stroke prediction**

I. Introduction

When blood flow to certain sections of the brain is interrupted or reduced, the cells in those areas of the brain do not receive the nutrients and oxygen they require, and they die. A stroke is a medical emergency that need immediate medical care. To avoid additional damage to the afflicted area of the brain and potential consequences in other sections of the body, early identification and effective therapy are essential. Strokes impact fifteen million people globally each year, according to the World Health Organization (WHO), with one person dying every four to five minutes in the afflicted population. According to the Centers for Disease Control and Prevention (CDC), stroke is the sixth greatest cause of death in the United States. Stroke is a noncommunicable illness that claims the lives of about 11 percent of the population. Approximately 795,000 persons in the United States suffer from the devastating consequences of strokes on a regular basis. It is the fourth largest cause of mortality in India. Ischemic and hemorrhagic strokes are the two types of strokes. Clots hinder drainage in a chemical stroke, whereas a weak blood artery breaks and bleeds into the brain in a hemorrhagic stroke. Stroke can be avoided by living a healthy and balanced lifestyle that includes quitting smoking and drinking, maintaining a reasonable BMI and glucose level, and having great heart and kidney function. Predicting a stroke is critical, because it must be treated quickly to avoid permanent damage or death. With advancements in medical technology, it is now feasible to predict the start of a stroke using machine learning techniques. The algorithms used in machine learning are advantageous since they allow for accurate prediction and analysis. The majority of prior stroke research has concentrated on predicting heart attacks, among other things. Only a few research have been conducted on brain stroke. The major goal of this research is to show how machine learning may be used to predict the start of a brain stroke. The most essential component of the methods used and the results obtained is that Random Forest performed the best of the four classification algorithms examined, obtaining a higher accuracy metric than the others. The model has one drawback: it is trained on textual data rather than real-time brain pictures. This article shows the implementation of four machine learning classification techniques.

Machine learning has already been used by a number of researchers to predict strokes. Text mining and a machine learning classifier were employed by Govindarajan et al. to classify stroke conditions in 507 people. They looked at a number of machine learning approaches for training, including the Artificial Neural Network (ANN), and discovered that the SGD algorithm offered the best results, with a 95 percent success rate. Amini et al. conducted studies to predict the occurrence of a stroke. In 807 healthy and unwell people, they categorized 50 risk factors for stroke, diabetes, cardiovascular disease, smoking, hyperlipidemia, and alcohol consumption. They employed the c4.5 decision tree algorithm (which has a 95 percent accuracy rate) and the K-nearest neighbor technique (which has a 95 percent accuracy rate) (94 percent accuracy). Cheng et al. published a research paper on predicting the prognosis of an ischemic stroke. They employed 82 ischemic stroke patient data sets, two ANN models, and accuracy scores

of 79 and 95 percent in their research. Cheon et al. undertook study to see if a stroke patient's mortality might be predicted. They used 15,099 people in their study to determine the stroke incidence. They used a deep neural network approach to identify strokes. PCA was used by the authors to extract data from medical records and predict strokes. They have an area under the curve of 83 percent (AUC). Singh et al. used artificial intelligence to predict strokes in their study. In their investigation, they used the cardiovascular health study (CHS) dataset to test a new approach for predicting stroke. They also performed a feature extraction followed by a principal component analysis using the decision tree approach. The model was created using a neural network classification approach in this example, and it was 97 percent accurate.

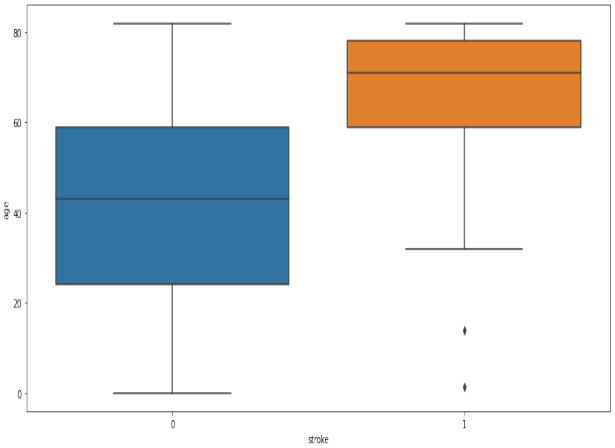
Chin et al. investigated the precision of an automated early ischemic stroke detection system. Their main goal was to develop a Convolutional Neural Network-based technique for automating primary ischemic stroke (CNN). For the goal of training and testing the CNN model, they gathered 256 images. In the image processing of their system, they used the data lengthening approach to enlarge the acquired picture. Their CNN approach was able to attain a 90 percent accuracy rate. Sung et al. investigated the development of a stroke severity measure. They studied 3577 individuals who had suffered an acute ischemic stroke. To develop their prediction models, they used a range of data mining techniques, including linear regression. They outperformed the k-nearest neighbor technique in terms of prediction (95 percent confidence interval). Machine learning was utilized by Monteiro et al. to predict the functional outcome of an ischemic stroke. They put their strategy to the test on a patient who died three months after being admitted. They were able to get an AUC of more than 90. In order to evaluate the risk of stroke, Kansadub et al. undertook study. The researchers used Naive Bayes, decision trees, and neural networks to evaluate the data and predict strokes. In their investigation, they evaluated the accuracy and AUC of their pointer. All of these algorithms were classified as decision trees, with naive Bayes delivering the most accurate results. The categorization of an ischemic stroke was investigated by Adam et al. They used two models to classify ischemic strokes: the k-nearest neighbor approach and the decision tree strategy. Medical professionals found the decision tree technique to be more effective for categorizing strokes in their study.

As previously said, the most significant contribution of our study is that we applied several machine learning models on a publically available dataset. The majority of the researchers in earlier studies employed a substantial model to predict stroke disease. We did, however, utilize four alternative models and compare the findings to earlier research.

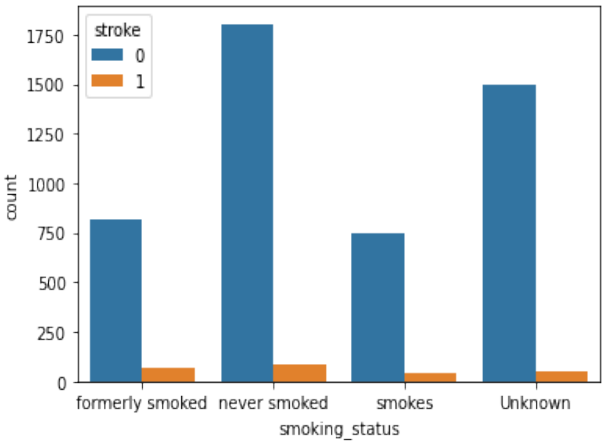
II. Data Set Description

The stroke prediction dataset was used to perform the study which was available in kaggle. There were 5110 rows and 12 columns in this dataset.The data dictionary, I pulled from Kaggle has the following columns, a unique identifier, so it was an integer identifying which patient the different variable options belong to. There was also the option of gender which were male and female. Age of the patient, it was an integer rather than a float. Hypertension which was listed as zero if the patient didn't have hypertension, or one if the patient had hypertension. Heart disease, so that was zero if the patient didn't have a history of heart disease, or one if the patient has heart disease. Ever married had different options for it which were no or yes. Work type, and the levels for it were children, government job, never worked, private or self-employed. Resident's type and its levels were rural, or urban. Average glucose level, so this was average glucose level in the blood, and it was measured in milligrams per decilitre. BMI body mass index, which is a ratio of mass and height and it's typically used to measure somebody's obesity or malnourishment. Smoking status, and the options for it were formerly smoked, never smoked, smokes or unknown and finally, stroke if the patient had a stroke or if they did not and the levels for that were one or zero respectively. So, the objectives of my project were to attempt to answer the central question, so can we predict stroke? I also wanted to determine what different machine learning algorithm performs best out of these four, Logistic Regression, Decision Tree, Random Forest, and Gradient Boosted Tree. I was also interested in seeing what kind of insight we can pull from this data. So prior evidence, it's a well-documented fact that a healthy BMI range is anywhere between 20 to 25. About 25 is a little bit on the obese side and under 20 is starting to get into malnourishment. And so, we have a slew of problems associated with being under over this interval. A healthy average glucose level range of anywhere between 83 to 120. And again, we have a slew of medical problems associated with being underneath or over the over this. This interval we also see a high rate of stroke and samples where individuals have a history of hypertension, heart disease, smoking and gender listed as female. So, one of the things I found was that if you were a female who was older and has hypertension, you're more likely to develop stroke. Furthermore, I also see there is a lot missing values in terms of the colum 'bmi' which we will be filling in order to train the model effectively. Also, there are many outliers in the average glucose values. We have stroke versus age, stroke versus body mass index, and stroke versus average glucose level. So here we have some bar plots for the different categorical features, and I wanted to take a look at the occurrence of stroke within with people who responded to different levels of this of these features. So surprisingly, there were there wasn't a big difference between male and female as you can see here in the scatterplot. Unsurprisingly, we do have a big difference in the amount of stroke and people with hypertension versus not. We have a big difference again in individuals who have heart disease versus the ones that do not. Surprisingly, we have a difference between people who were married or not in the amount of stroke in the dataset. With respect to work type versus estimated stroke average and you can see private and government job are pretty much almost equal and self-employed is a little bit higher and I kind of associate this with a little bit more stress of owning your own business or you are being self-employed in general and here we have residence type. There's not really a big difference here as well.

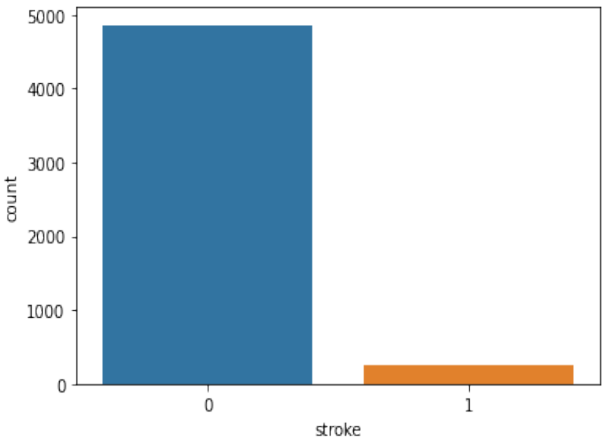
Furthermore, I noticed that most of the of the individuals who had stroke were above the healthy BMI range which is 40. Furthermore, we would expect more obese individuals to have greater occurrence of stroke.



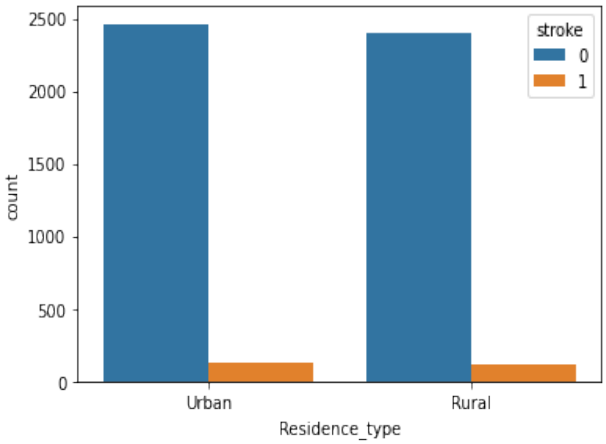
$Fig1 : Age$ (1)



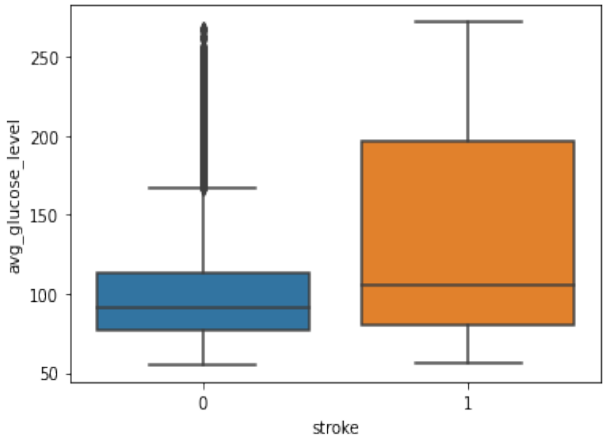
$Fig2 : Smoking - Status$ (2)



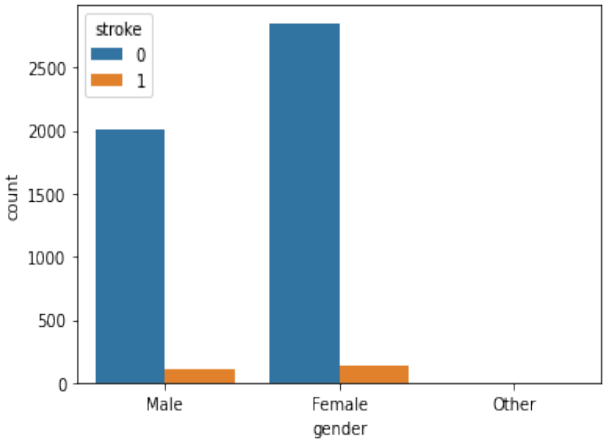
$Fig3 : Stroke - Count$ (3)



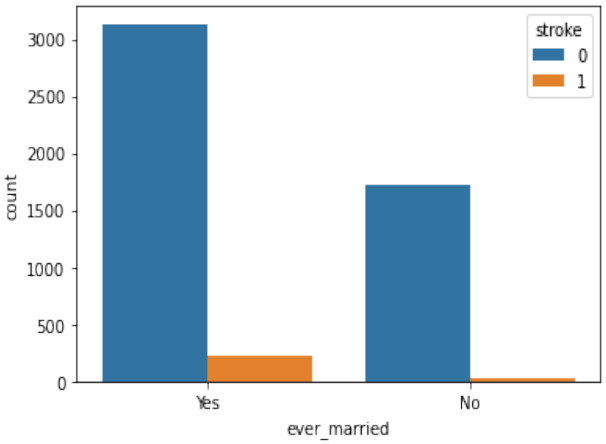
$Fig4 : Residence - Type$ (4)



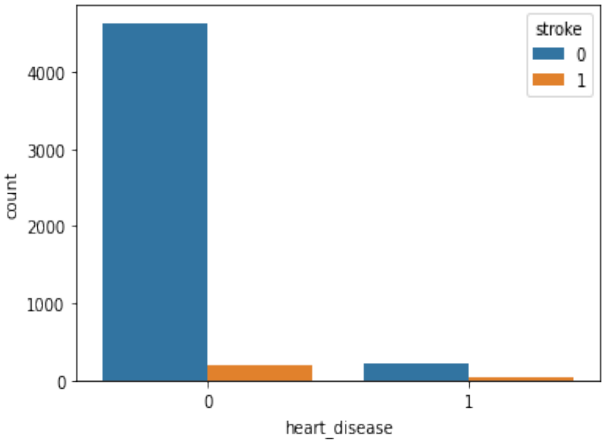
$Fig5 : Average - Glucose - Level$ (5)



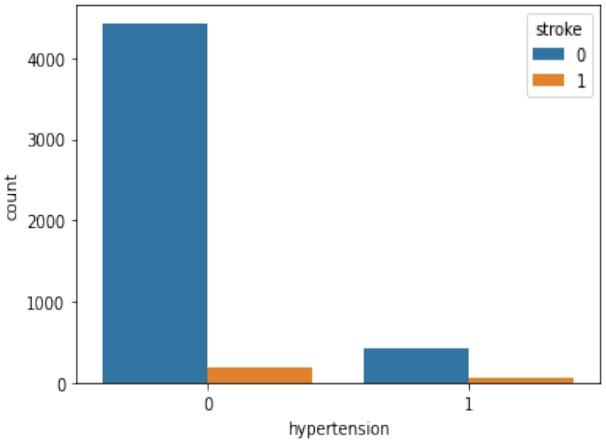
$Fig6 : Gender$ (6)



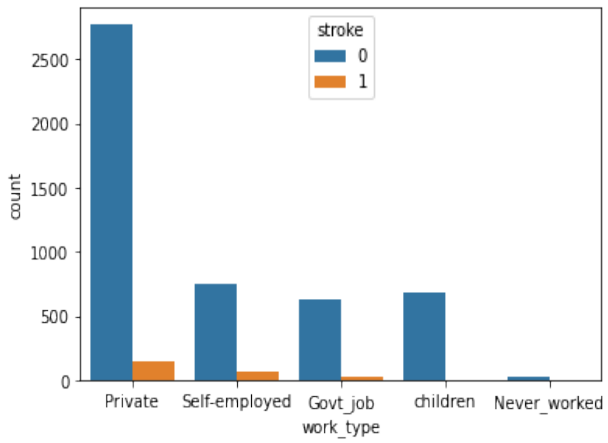
$Fig7 : Married - status$ (7)



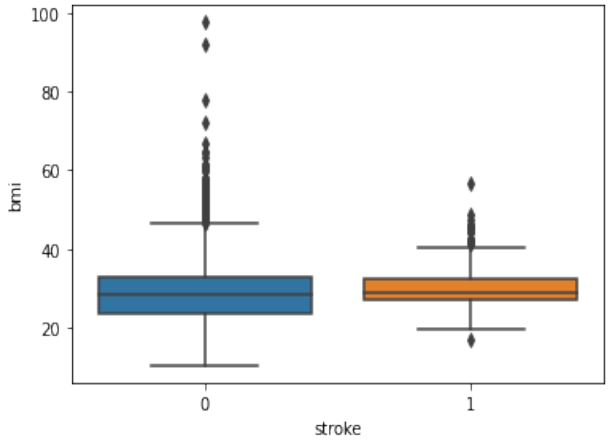
$Fig8 : Heart - Disease$ (8)



$Fig9 : Hypertension$ (9)



$Fig10 : Work - type$ (10)



$Fig11 : BMI$ (11)

III. Methods

We have carried out the entire project in Google collab platform. At first, we have installed the required spark libraries and have imported all the required libraries such as NumPy, pandas, matplotlib, etc. Once all the required libraries are imported, we have created a Spark Session with the app name as stroke prediction. Post initializing a Spark session, we have imported the csv data of patients which is uploaded to GitHub repository via the URL. And we have passed this URL to the Spark Session. In order for the spark data frame to correctly read the CSV data file, we have made sure that we have provided the header as true, and we have set the infer schema to true so that the spark builder can automatically infer the schema of the data frame. If we further go down to visualising the data and looking at the schema, we see that there are 5110 rows with 11 columns, out of which the gender column is a string, the age is a double, hypertension is integer, heart disease is an integer, ever married is a string, work type is a string, resident type is a string, average glucose level is double, BMI is string, smoking status is a string and the last which is our target variable stroke is an integer. Before we proceed with visualising the data or analysing the data, we have separated the numerical columns which would be age, hypertension, heart disease, average glucose level

and stroke and we have separated the categorical columns which would be gender, ever married, work type, residents' type, BMI and smoking status. When we further investigate the data, we see that the BMI has only 4909 values which means that there are 201 values missing. To train the model effectively, we have filled the null values of BMI with the mean value of the BMI, we see that the mean value of the BMI would be 28.44. So all the columns which has a blank or no value on BMI is filled with the value 28.44. Furthermore, to provide this data to our machine learning models, we have converted all the strings to integers by using string indexer and, by using vector assembler, we have converted all the data types to a similar range. Furthermore, we have also created a Pyspark pipeline wherein once the data is passed on to it, it is automatically string indexed encoded and the data is transformed, and it is fitted into the data frame. So once all the data is passed on to the pipeline, we have split the data into 70 and 30 percent, wherein the 70 is training data set and the 30 percent would be the test data set. We have used the PySpark ML classification library to import various decision-making algorithms. Machine Learning models such as logistic regression, as well as Decision tree, random forest classifier and gradient boosted tree classifier. Upon training the model and we have used binary classification evaluator to check the area under the curve of each classification model, which is the ROC. We see that our logistic regression model was able to provide a 79 percent accuracy, Decision tree model was able to provide 52 percent accuracy and the Random Forest classifier algorithm was able to provide 77 percent accuracy. Furthermore, the gradient boosted tree classifier was able to provide us 80 percent accuracy. When comparing all the models here which is logistic regression, decision tree, random forests, and gradient boosted tree. We see that gradient boosted tree was much more efficient in predicting the stroke which gave us accuracy of 80 percent. We furthermore checked the correlation of this data by plotting a heat map also we have compared the model and the accuracy which is indicated in the bar chart below.

IV. Proposed Algorithms

A. Logistic Regression

Figure 12 depicts the flowchart for the logistic regression model. LR is one of the most often used ML algorithms in the supervised learning method. It is a forecasting strategy that predicts a categorical dependent variable using a set of independent factors. The output of a categorical dependent variable is predicted using logistic regression. As a result, the output will have to be discrete or categorical. Yes or no, 0 or 1, true or false, etc. are all possibilities, but only probabilities between 0 and 1 are presented. The methods of logistic regression and linear regression are fairly similar. LR is used to solve classification difficulties, whereas linear regression is used to solve regression problems. We utilize an S-shaped logistic function to forecast the two maximum values instead of a regression line (0 or 1).

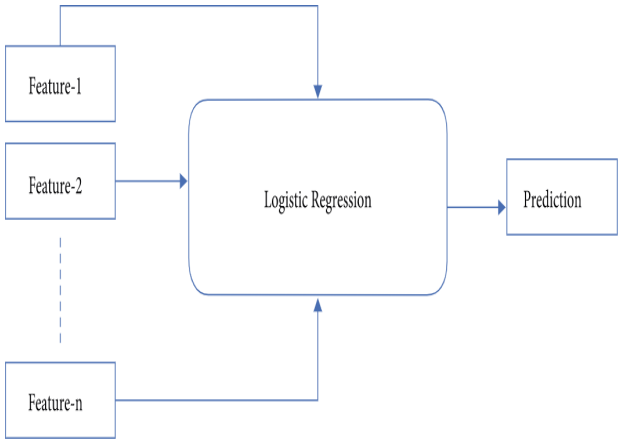


Fig 12: Logistic Regression Model

B. Decision Tree

Classification with DT addresses both regression and classification difficulties. This approach is also a supervised learning model since the input variables already have a linked output variable. It has the appearance of a tree. In this strategy, the data is continually split based on a certain parameter. A decision tree is made up of two parts: a decision node and a leaf node. The data is split at the first node, and the result is produced at the second. Figure 5 shows the fundamental construction of the DT classifier. The DT is simple to understand since it mimics the stages that a person goes through while making a real-life choice. It might be quite helpful in addressing decision-making challenges. Consider all of the possible solutions to a problem. Data cleaning is not as necessary as it is with other approaches.

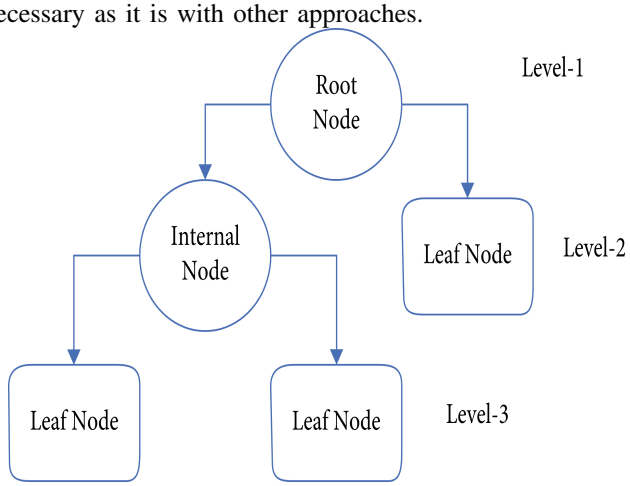


Fig 13: Decision Tree Model

C. Random Forest

RF classification was used as the classification algorithm. RFs are made up of a number of separate decision trees that were each trained on a random sample of data. These trees are produced throughout the training process, and the outputs of the decision trees are gathered. The ultimate forecast generated by this algorithm is determined by a vote method. This technique requires each DT to vote for one of the

two output classes (in this case, stroke or no stroke). The RF method, which picks the class with the greatest votes, determines the final forecast. One of the random forest's most appealing aspects is its versatility. It may be used for tasks like relapse detection and categorization, and the overall weighting assigned to information attributes is clear. Furthermore, it is a desirable technique since the default hyperparameters it applies frequently provide clear expectations. Because there are so few of them to begin with, understanding the hyperparameters is crucial. Overfitting is a well-known issue in machine learning, yet it only happens once in a while with the arbitrary random forest classifier. The classifier will not overfit the model if there are enough trees in the forest.

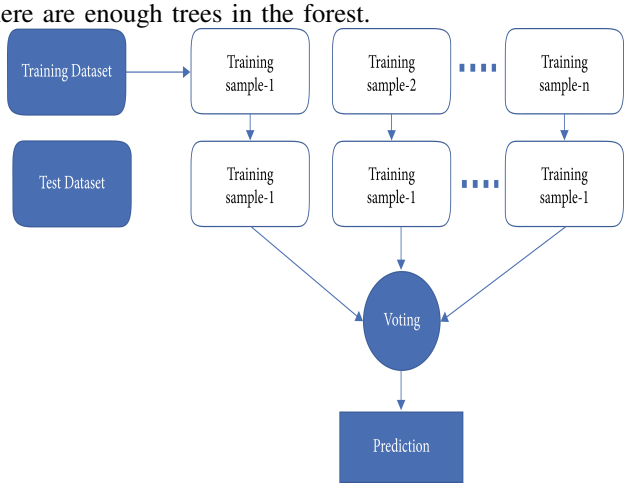


Fig 14: Random forest Model

D. Gradient Boosted tree

Gradient Boosted Trees and Random Forests are both ensembling algorithms that combine the outputs of separate trees to accomplish regression or classification. They both incorporate a large number of decision trees to lessen the danger of overfitting that each tree confronts alone. They differ, however, in how the separate trees are constructed and how the results are merged. Bagging is used in random forests to create independent decision trees that are then combined in parallel. Gradient boosted trees, on the other hand, employ a technique known as boosting. Boosting is a technique for successively combining weak learners (typically decision trees with just one split, sometimes known as decision stumps) so that each new tree corrects the faults of the prior one. Gradient boosting decision trees combine several weak learners to produce a single strong learner. Individual decision trees are the poor learners in this case. All of the trees are connected in a succession, with each tree attempting to reduce the mistake of the one before it. Boosting algorithms are often difficult to train but extremely precise due to this sequential relationship. Models that learn slowly do better in statistical learning. As the model improves, the weak learners are fitted in such a way that each new learner fits into the residuals of the preceding stage. The final model combines the results of each phase, resulting in a strong learner. The residuals are detected using a loss function. For example, in a regression work, mean squared error (MSE) can be utilized, and in a classification task, logarithmic loss (log loss) can be employed. It's worth noticing that when a new tree is added to the model, nothing changes. The residuals from the present model are well-fit by the decision tree that has been introduced.

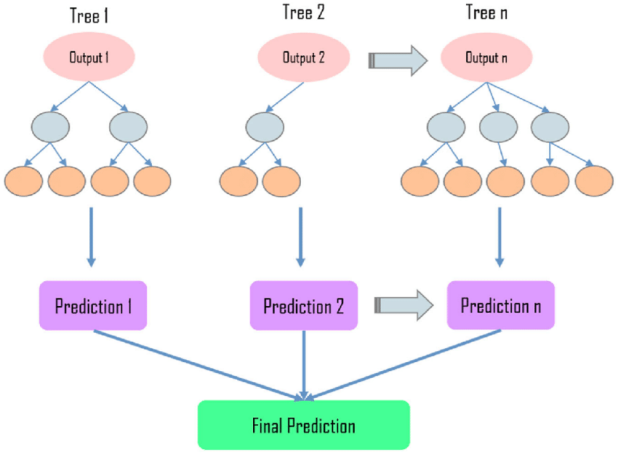


Fig 15: Gradient Boosted tree Model

V. Co-Relation

Measuring the inter-feature correlation might be beneficial at times. If a feature is substantially linked with the class (target), this might indicate that the characteristic is useful for learning about the class. Furthermore, if a feature is significantly associated with other variables, we might eliminate one of them, lowering the model's complexity while perhaps enhancing its learning. Upon analysing the below heat map we see that the following variables have an effective correlation: avg glucose level, age, hypertension, heart disease, ever married

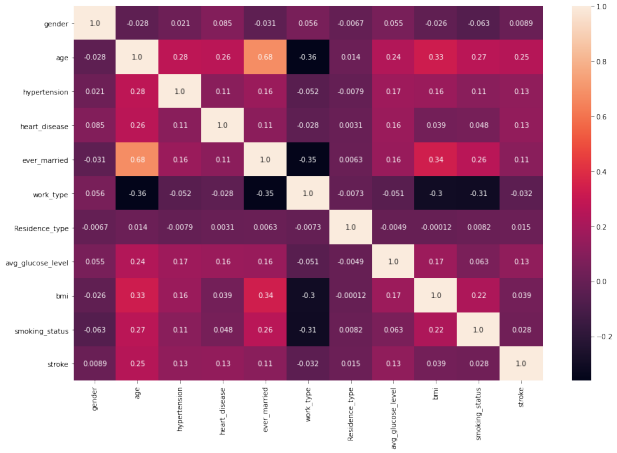


Fig 16: Co-relation matrix

VI. Model Training

We have used the area the curve method to determine the efficiency of the algorithms. The AUC is a measure of a classifier's ability to discriminate between classes and is used to describe the ROC curve. The higher the AUC, the better the model's ability to distinguish between positive and negative classifications. In a ROC curve, a higher X-axis value suggests that there are more False positives than True negatives. There are more True positives than False negatives

when the Y-axis value is greater. As a result, the threshold should be determined by the capacity to maintain a stable ratio of False positives to False negatives. Below are the results:

1. Logistic Regression: Test Area Under ROC - 0.7903047490867173
2. Decision Tree: Test Area Under ROC - 0.5224355893097481
3. Random Forest: Test Area Under ROC - 0.7714381849644297
4. Gradient-Boosted Tree: Test Area Under ROC - 0.809792187399859

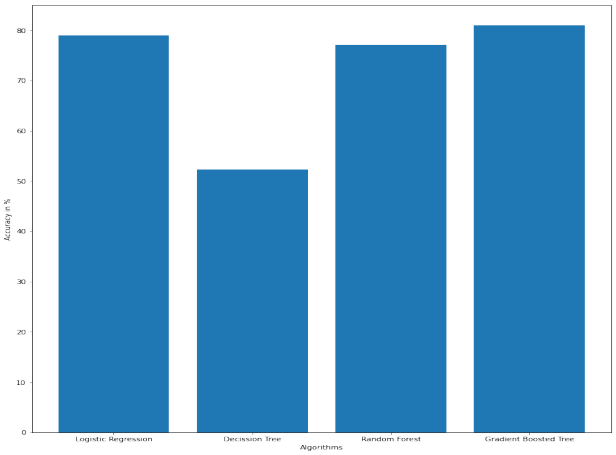


Fig 17: Model Accuracy

VII. Results

Upon visualizing, training and modelling the data I was able to achieve the best result using the Gradient boosted tree classifier algorithm which fulfills my first goal. Furthermore, I see that The target characteristic is strongly correlated with age, hypertension, avg glucose level, heart disease, ever married, and BMI. Gender, on the other hand, has a negative correlation with stroke.

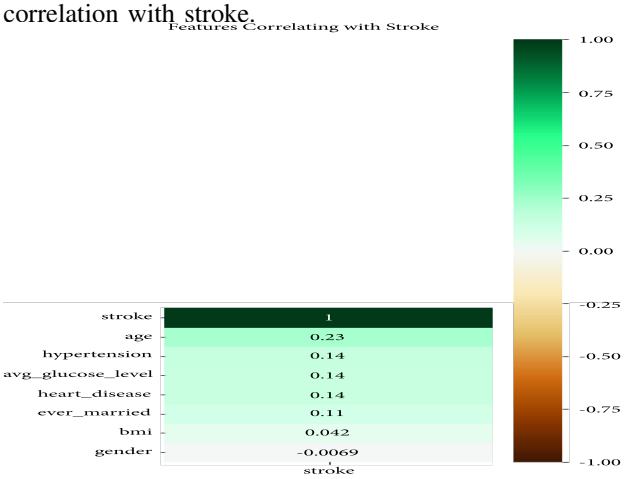


Fig 18: Important Co-related Features

VIII. Conclusion

Stroke is a life-threatening medical condition that requires prompt treatment to avert future problems. The creation of a machine learning model might assist in the early diagnosis of stroke and, as a result, the reduction of its severe repercussions. This study examines the effectiveness of multiple machine learning algorithms in correctly predicting stroke based on a variety of physiological factors. With a classification accuracy of 80 percent by using Gradient Boosted Tree Classifier algorithm.

IX. Future Work

The framework models may be improved in the future by employing a larger dataset and machine learning models such as AdaBoost, SVM, and Bagging. This will improve the framework's reliability as well as its appearance. The machine learning architecture may assist the general public in identifying the possibility of a stroke developing in an adult patient in return for only supplying some basic information.

X. References

[1] https://www.hindawi.com/journals/jhe/2021/7633381/ [2] https://www.kaggle.com/fedesoriano/stroke-prediction-dataset [3] https://spark.apache.org/docs/latest/ml-guide.html [4] https://www.analyticsvidhya.com/blog/2019/11/build-machine-learning-pipelines-pyspark/