Acute Ischemic Stroke Prediction

Using Machine Learning to predict Hemorrhagic strokes in patients

Ritvik Chebolu (rc2388@rit.edu) (UID: 376005372)
Alekya Yakama (ay2423@rit.edu) (UID: 774009415)
Murali Krishna Muthukuru (mm6589@rit.edu) (UID:786000428)
SaiTulasi Kamma (sk5656@rit.edu) (UID:788006833)
Ganesh Sandeep Kanumilli (gk5951@rit.edu) (UID: 383006249)

1. Our Hypothesis

The aim of our project is to predict acute ischemic strokes in susceptible patients taking into consideration various medical factors that could potentially influence the effect of stroke to eventually build machine learning models that could train from a model dataset and make predictions for future patients.

In our case, we first assume all medical features in the dataset to be contributing factors to predict the stroke severity in a patient. We then move on to predict the stroke severity using only the factors that are more likely to influence the stroke. We find it reasonable to hypothesize that these influencing features will be responsible for strokes that may be observed in patients outside this dataset as well.

2. Data Extraction

Our dataset originated from Gandhi Medical Hospital, Hyderabad, where a post graduate medical student collected data from 200 patients in the year 2020 who got admitted after reporting symptoms of a brain stroke. Studies like this could possibly help researchers and doctors understand the underlying factors that could lead to a stroke and also estimate the stroke severity based on factors such as age, serum albumin, MRS score (Modified Rankin Scale score) and SSS.

The stroke severity index from this table can again be used to predict the disability of the patient. So if the first features were categorized as the ‘cause’ of the stroke, the stroke severity would be the ‘effect’. And if the stroke severity index is considered as the ‘cause’, then the disability experienced by the patient would be considered ‘effect’.

The raw dataset had 5 categorical feature columns (excluding the patient name) and 6 numeric feature columns. The ‘Sl. No.’, ‘Name’ and ‘Patient ID’ were dropped since they do not make any significant contribution while predicting the stroke severity. The ‘Sex’ column was label encoded since there were only 2 categories. The ‘Stroke Severity’ and ‘Disability’ were also label encoded since they were ordinal features. The ‘Addictions’ and ‘Comorbid Conditions’ features were segregated using one hot encoding since there were cases where a patient had multiple comorbid conditions and/or addictions.

The dataset contains just 200 data points and might seem to be quite small for making predictions. However, it costs a fortune and also a lot of effort for healthcare

institutions to collect such primary raw data (or observational data). The data may have been biased since it was collected from just one hospital. For example, the people in the region could have genetically developed a particularly high value of serum albumin due to which there could be more frequent strokes. Also, the dataset is slightly inclined towards the elder generation as this seems to be the case with many disorders and health conditions throughout the world.

Being a small dataset, there was a lot of possibility of noise, bias and variance. There are possibilities of generating synthetic data using python libraries like Pandas, Faker and Virtual Data Lab. However, they generate random data without trying to identify any underlying patterns, which ultimately reduces the accuracy of the machine learning model. Hence, we decided to maximize our precision by using methods like k-fold cross validation so that a part of the data is not lost while training the ML model. Though there could be numerous other factors other than those considered in the dataset which could have strongly influenced the stroke severity, we have restricted ourselves to using the data from our dataset. Future research and data collection could count on such features for making better predictions.

3. Literature Review

An ischemic stroke occurs when there is a sudden blockage or reduction in blood flow to the brain, which could prevent sufficient oxygen supply to the brain tissues required for biological processes. To predict the stroke severity, a fair amount of previous research was conducted using numerous methods.

[1] In this study of previous research, four diverse machine learning algorithms were implemented for acute ischemic stroke prediction namely Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), Extreme Gradient Boosting (XGB) of which Logistic Regression is implemented for Classification problems. This is rooted from the idea of conditional probability with the use of sigmoid function as its cost function. Compared to LR, SVM constructs a set of hyperplanes in higher-dimensional space that result in the greatest distance from the closest training data point for a given class. The larger the distance for such a margin, the lower the generalization error of the classifier. As an end-to-end tree boosting system proposed for sparse data and weighted quantile sketch for approximate tree learning, RF builds a multitude of decision trees for training and outputs a classification as derived from an aggregate prediction of individual trees. A minimal number of resources was used to solve a problem of real-world scale.

[2] In this retrospective study of previous research, 512 new patients were enrolled. The modified Rankin scale (mRS) which is one of the crucial factors was

predicted by using extreme gradient boosting (XGB) and gradient boosting machine (GBM) models using biomarkers from admission and 24 hours after admission. The use of decision-tree-based GBMs can improve the decision-making processes for stroke treatment. If stroke patients are separated into groups based on recanalization or nonrecanalization, it may help with treatment planning.

4. Analysis Strategy

Our first step was to import basic libraries which will be useful for the tasks to come in the project. Then the dataset was imported into a data frame using pandas. To understand the relationship between all the parameters present in the dataset, a correlation matrix has been evaluated for the said dataset and a heatmap is generated in order to visualize the correlation. The next logical step to be taken is to understand the properties of the dataset like the number of missing values, unique value in each column and the datatypes of each column. This result decides the type of pre-processing, feature engineering and encoding that has to be done on the dataset before splitting the data and feeding it to the machine learning models. Then we generated a data profile using pandas-data profiling library which is an in depth visualization of all individual features along with their inter dependencies. As a part of pre-processing we then dropped a few unnecessary columns which proved to be non-effective on stroke severity like serial number, name of patient etcetera. As the data we are using has a limited amount of 200 records we perform K-fold cross validation which is a resampling technique used to increase the limited amount of data while training and testing machine learning models. The reason behind not opting for the generation of synthetic data as it might side track the results of the model due to wrong patterns of data. Then we performed one-hot encoding in excel sheet as we faced multiple challenges when we tried to do the same in python. The column addictions and comorbid conditions had multiple string values separated by comma. Later label encoding was executed on few required columns like sex, stroke severity and disability.

Once all the necessary changes are made to the dataset the next step was to build models and train them. Before starting off with the models we split the dataset into train and test on 75:25 ratio. The models to be used are decided based on the research done in the literature review and the models which are said to be performing well in stroke prediction. In this project eight different models have been built and trained namely, Random Forest, Decision Tree, Extreme Gradient Boost classifier, Support Vector Classifier, Bernoulli Naive Bayes, Gaussian Naive Bayes, Logistic Regression and K Nearest Neighbors. For each of these models once they have been trained then they have been tested and then confusion matrix and various other parameters have been calculated along with accuracy. Each of these metrics define the success and failures of the models. The strategy behind calculating all these metrics is that we aim to

build an ensemble model from the best of the tested models as it is well known an ensemble model performs better than any model individually can. It is an efficient form without much need of high computational power. But before doing this we decided to fine tune the model by evaluating the best performing hyperparameter for each model by performing Grid cross validation which is a tuning technique. Once we determine the best performing models along with their parameters we fine-tune them which in return increases the accuracy and other parameters like precision, f1 score and recall. All the results are visualized in a tabular form from which the best five performing models are selected to form an ensemble. For the ensemble model in our project a Voting classifier is being used which is an advanced machine learning ensemble model. The reason for choosing this model is that it predicts an output based on all the models' combined highest probability of chosen class as the output. They are two forms of voting in this classifier that are namely soft and hard. In this project we chose to implement soft voting as it gives the average of probability of the given class that is the inclusion of outputs from all the models that are included in the ensemble. Then once again the parameters like confusion matrix, precision, recall , accuracy and f1 score are calculated for the voting classifier which are definitely better than the models implemented individually. Then the results are analyzed from which we draw the conclusion that Stroke Severity is very strongly correlated to GCS (Glasgow Coma Scale) and SSS (Siriraj Stroke Score). From analysis it was realized that the addictions and disability very slightly impacts the stroke severity. However we also observed that the data is quite imbalanced as there were no young patients found with this condition hence one again it is proved using synthetic data might hinder our machine learning model useless in terms of predicting the severity of the stroke.

5. Code Analysis

Our program starts with importing the required libraries for this project. These libraries include Pandas, NumPy, Scikit Learn, Matplotlib, Sweet Viz, etc. The data is loaded from Github using a URL into a Pandas dataframe. After we load the dataset, all column names are renamed to use the column names in the dataframe at a later stage. We then dropped the insignificant columns which weren’t dependent on the prediction column. Since our dataset had no null or missing values, we could directly proceed to data handling and preparation.

We split the dataset into train and test sections to perform data visualization and exploratory data analysis. As a part of data visualization we generated a report indicating the relationships between all the parameters present in the data set using sweetviz library. Then we have used the info() and describe() functions to understand the properties of data of the dataset. Then we evaluated the number of null values and

unique values in each column. A heat map is generated using seaborn to study the correlation of the dataframe. And we generated the pairplot of the data to get deeper insights. Later we performed preprocessing by using label encoder() for the columns: Sex, Stroke severity and disability. Then came the one hot encoding for the columns addictions and comorbid conditions.

In the model building section, multiple libraries were imported for the necessary models from sklearn library and also few metrics were imported to analyze the efficiency of the machine learning models executed. The models were then trained and tested and the results are stored in variables. All the named models are appended into an array. The metrics considered are confusion matrix, accuracy score, k-fold validation mean accuracy, standard deviation, ROC AUC score, precession, recall and f1 score. All these results are stored in a tabular form in an ascending order of accuracy and k-fold mean accuracy.

Then comes the fine tuning of the models using grid search cross validation, which is a fine tuning technique. The mechanism of the GridSearchCV is that it analyzes all the parameters given in an array or a grid to evaluate the hyper parameters which are best performing for the given model. For example, consider logistic regression for which the parameters considered are C and random state. Post hyper tuning, we understood that the best values for C and random state are 1 and 0 respectively and the model attains 96.88% accuracy. Similarly the best accuracy producing hyper parameters are calculated for all the models individually.

	Model	Accuracy	K-Fold Mean Accuracy	Std. Deviation	ROC AUC	Precision	Recall	F1
6	Random Forest	100.0	96.875	4.192627	1.000000	1.000000	1.000000	1.000000
5	Decision Tree	100.0	96.250	3.061862	1.000000	1.000000	1.000000	1.000000
7	XGBoost	95.0	97.500	3.061862	0.888889	1.000000	0.777778	0.875000
0	Logistic Regression	95.0	96.875	3.125000	0.888889	1.000000	0.777778	0.875000
1	SVM	95.0	96.250	4.145781	0.888889	1.000000	0.777778	0.875000
2	KNeighbors	85.0	85.000	8.003905	0.784946	0.666667	0.666667	0.666667
4	BernoulliNB	75.0	82.500	11.110243	0.681004	0.454545	0.555556	0.500000
3	GaussianNB	65.0	71.250	8.926786	0.774194	0.391304	1.000000	0.562500

After analyzing the output of all the models using the best estimator we determined that k nearest neighbors , random forest , support vector classifier, decision tree and extreme gradient boosting classifier are found to be better performing than the rest and hence they are combined into an ensemble model. We have used a voting classifier ensemble model for this purpose. The results of the ensemble model were analyzed just like it was done for the rest of the models. Then the learning curves are plotted for training score and cross validation score for all the models implemented.

6. Working Plan

Ritvik- Data acquirring and cleaning data. Then building and testing machine learning models namely Decision tree, Random Forest and Extreme gradient boosting classifier.
Alekya- Fine Tuning of all the implemented models by estimating the best hyper parameters and then building an ensemble model. Also performed result analysis.
Murali- Performed Data visualization using Sweetviz and pandas data profiling. Generated heat maps based on correlation matrix. Build and implement a support vector Machine classifier.
Sandeep- Worked on three different machine learning models namely Bernoulli Naive Bayes, Gaussian Naive Bayes and Extreme Gradient Boosting Classifier.
Tulasi- Worked on data preprocessing like one hot encoding and label encoding. Then worked on machine learning models like K nearest neighbor and logistic regression.

7. Team Collaboration

In terms of team collaboration, it was kind of challenging to find time and have consistent meetings as our schedules were very conflicting. We did try to have consistent in person meetings, but when it was not possible, we scheduled zoom meetings. The first few meetings were spent on the project idea, then we performed individual tasks for research. We faced trouble when dealing with the data as it was limited and we spent many weeks on preparing the data and to convert it into a csv file. We managed to figure out all the challenges that we faced together while implementing this project code and we also managed to overcome differences of opinions on how to execute certain steps in the project.

8. Individual Contribution

As a part of the team of predicting acute ischemic strokes I served the purpose of preprocessing the data and building one of the machine learning models known as K-Nearest Neighbor. The term "pre-processing" refers to a technique used to turn raw data into a clean data set. This means that when the data is gathered from different sources it is collected in raw format, making analysis impossible. Ideally, the format of the data should be in a way that will allow the applied model to produce better results in Machine Learning projects. A specific machine learning model needs information in a

specific format, for example, Random Forest algorithm does not support null values, thus null values must be managed from its original raw data set. Furthermore, the data set should be structured in such a way that a few Machine Learning and Deep Learning algorithms can be executed and the best of them is chosen. The term Data Preprocessing generally refers to the process of preparing data so that it can be mined in a more efficient way. Data Preprocessing includes certain number of steps as stated below

1.Importing Libraries

2.Importing the Dataset

3.Handling of Missing Data

4.Handling of Categorical Data

5. Splitting the dataset into training and testing

6. K-fold Cross-Validation

As a part of Data preprocessing importing certain libraries was the primary step that was done which plays a very pivotal role in implementing the code. Now, moving ahead to the second step of importing dataset, importing data sets is a very crucial one again as the very core implementation is done on the data set by uploading or importing the same. To brief a little on the data set that is currently used, it is a patient data set of 200 records who were admitted at the Gandhi Medical Hospital after marking the symptoms of hemorrhage or brain stroke in the year 2020. Since there are very few data points of 200, a resampling technique of K-fold Cross Validation was implemented after splitting the dataset into train and test for lesser variance and bias also to improve the accuracy. As the data that was collected was a perfect and clean one, handling of missing values was not mandatory to implement. Considering the less data points, synthetic data could have been generated, however, generating so would have impacted the accuracy as the target is stroke severity and the factors influencing it were real time ones and the data collected was a real time data due to which synthetic data generation was not considered as an option to generate extra data. Now moving ahead to handling of categorical data there are certain categorical features featured in the data set such as stroke severity which is the target feature, comorbid conditions, addictions, sex, and disability are the factors influencing the stroke severity, to handle these categorical features label encoding, and one hot encoding was done. Label encoding was done to handle features like sex, disability and stroke severity which are the ordinal values, and one hot encoding was done to comorbid conditions and addictions features which was quite a bit challenging as certain patients had multiple comorbid conditions as true. After the categorical to numerical conversion of data train, test split of data was done. It was

done with 80 and 20 where 80% of the data was trained and 20% of it was used for testing the model that was constructed out of it. With this the data preprocessing part of the project comes to an end.

Now moving on to building a K-nearest neighbor model, knn is a supervised machine learning algorithm that uses labeled data as input and learns underlying patterns and gives an appropriate output when unlabeled data is given. This algorithm checks for the similarity while classifying the data. There are several types of distance metrics that can be used, Euclidean distance metric was used here. Several model evaluation parameters are manifested below to measure the efficiency of the model.

```
KNeighbors :
[[28  3]
 [ 3  6]]
Accuracy Score:  0.85

K-Fold Validation Mean Accuracy: 85.00 %

Standard Deviation: 8.00 %

ROC AUC Score: 0.78

Precision: 0.67

Recall: 0.67

F1: 0.67
```

9. References

[1]. M. C. I., H. Z. W. and H. R. L. C., "Using multiclass machine learning model to improve outcome prediction of acute ischemic stroke patients after reperfusion therapy," *International Computer Symposium,* pp. 225-231, 2020.

[2]. Y. Xie, B. Jiang, E. Gong and Y. Li, "Use of Gradient Boosting Machine Learning to Predict Patient Outcome in Acute Ischemic Stroke on the Basis of Imaging, Demographic, and Clinical Information," *American Journal of Roentgenology.,* vol. 212, no. 10.2214/AJR.18.20260, pp. 44-51, 2019.

[3]. R. Trevisani and A. L. C. F. Lehmann, "Carotid intima media thickness measurements coupled with stroke severity strongly predict short-term outcome in patients with acute ischemic stroke: a machine learning study," *Springer,* vol. 36, pp. 1747-1761, 2021.