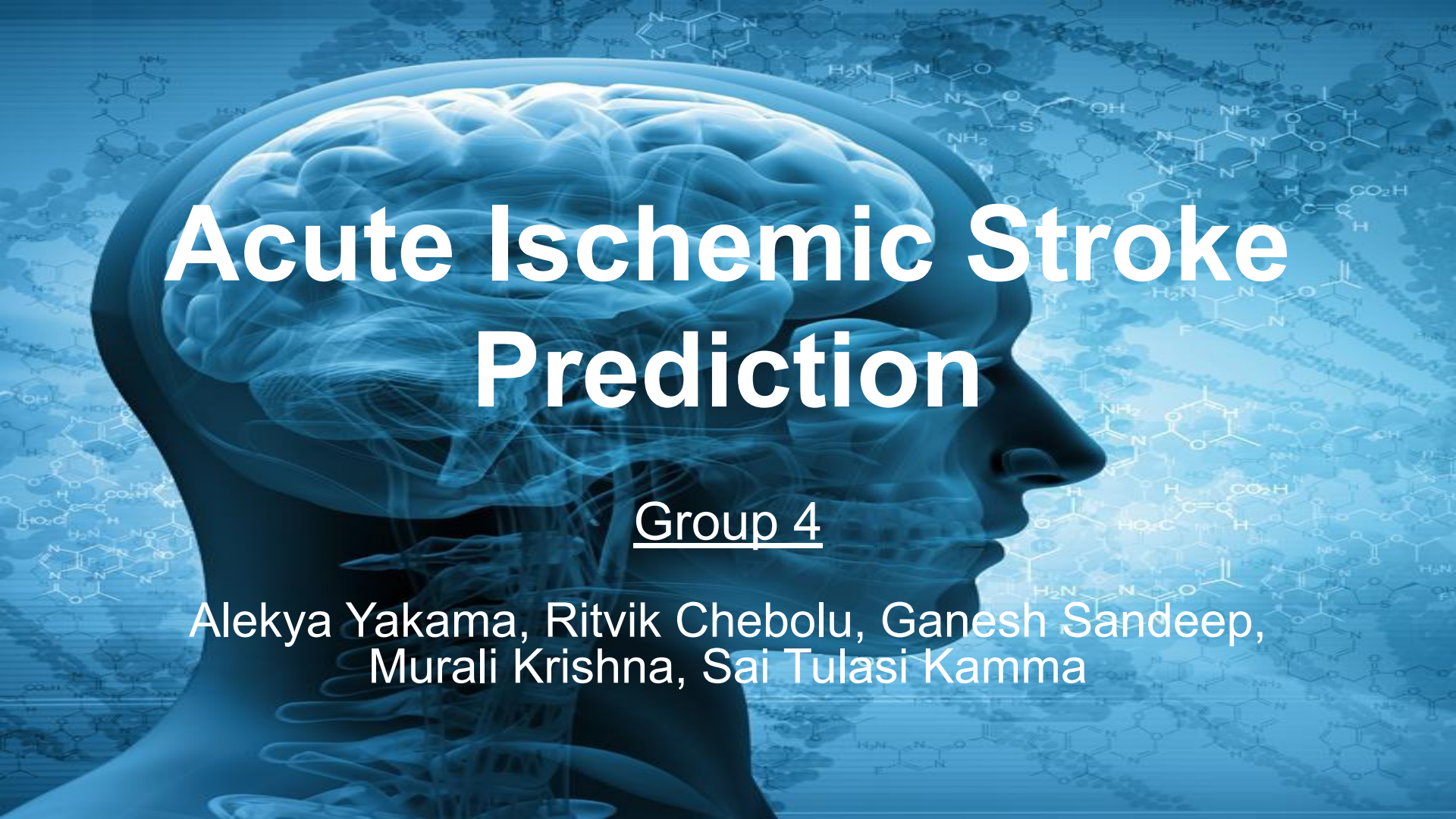


Acute Ischemic Stroke Prediction

Group 4

Alekya Yakama, Ritvik Chebolu, Ganesh Sandeep, Murali Krishna, Sai Tulasi Kamma

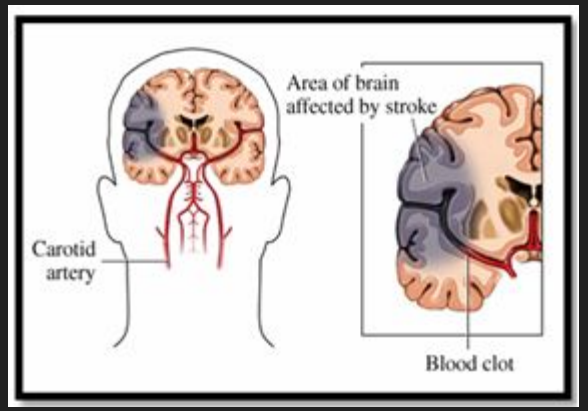
Table Of Contents

Introduction

- An acute ischemic stroke occurs when blood flow through a brain artery is blocked by a clot, a mass of thickened blood.
- Clots can either be thrombotic or embolic, depending on their location of development inside the body.
- A thrombotic stroke, the most common of the two, occurs when a clot forms within an artery in the brain.

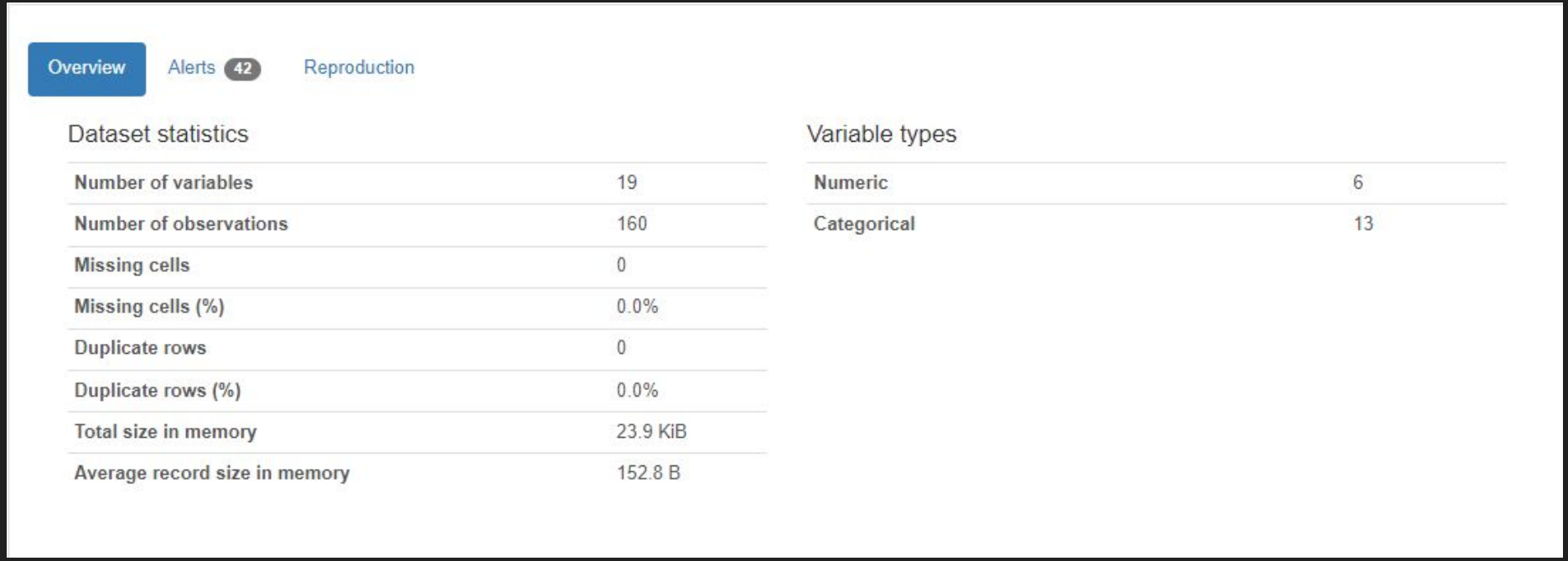
Problem Statement

Cerebrovascular accident or stroke is a life-threatening neurological disorder. It accounts for more than 50% of people hospitalized. Recent studies predict that in the coming decade, strokes might emerge as the second leading cause of morbidity and mortality in several developed countries. Our goal is to predict the acute ischemic stroke severity using medical features like GCS, Serum Albumin and SSS Score .



About our Dataset

- Our dataset is a compilation of 200 data-points that was collected by post-graduate doctors from a hospital (Gandhi Medical Hospital, Hyderabad) in India, in the year 2020.
- This data was collected from patients who reported to have experienced an acute ischemic stroke (brain stroke in simple terms) and were admitted in the hospital.
- The dataset contains several medical factors that could possibly be the critical factors influencing the severity of these hemorrhagic stroke in patients.



The above image indicates all the necessary information regarding the dataset at a glance.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Sl. No.	Patient ID	Name	Age	Sex	Lesion	GCS	Serum Albumin	SSS Score	Stroke Severity	MRS Score	Disability	Comorbid Conditions
2	1	20577	SAYAVVI	70	F	1	5	4.2	6	Severe	6	Death	A
3	2	20578	CHANDRA	70	F	4	14	4.2	30	Moderate	4	Moderate	A
4	3	20579	INDU	60	F	2	11	3.6	32	Moderate	4	Moderate	A,B
5	4	20580	KAVU	70	F	1	5	4.2	10	Severe	5	Severe	A,B,C
6	5	20581	LAKSHMI	55	F	1	9	3.2	14	Severe	5	Severe	E
7	6	20582	KAMALA	85	F	2	9	2.6	4	Severe	5	Severe	D
8	7	20583	H LAKSHMI	63	F	2	15	4.2	26	Moderate	4	Moderate	A
9	8	20584	THULASI	58	F	1	9	4.1	20	Severe	5	Severe	E
10	9	20585	SHALLU	85	F	1	5	3.8	2	Severe	6	Death	A,C
11	10	20586	ZABBERUNIKA	37	F	1	7	2.7	12	Severe	6	Death	F

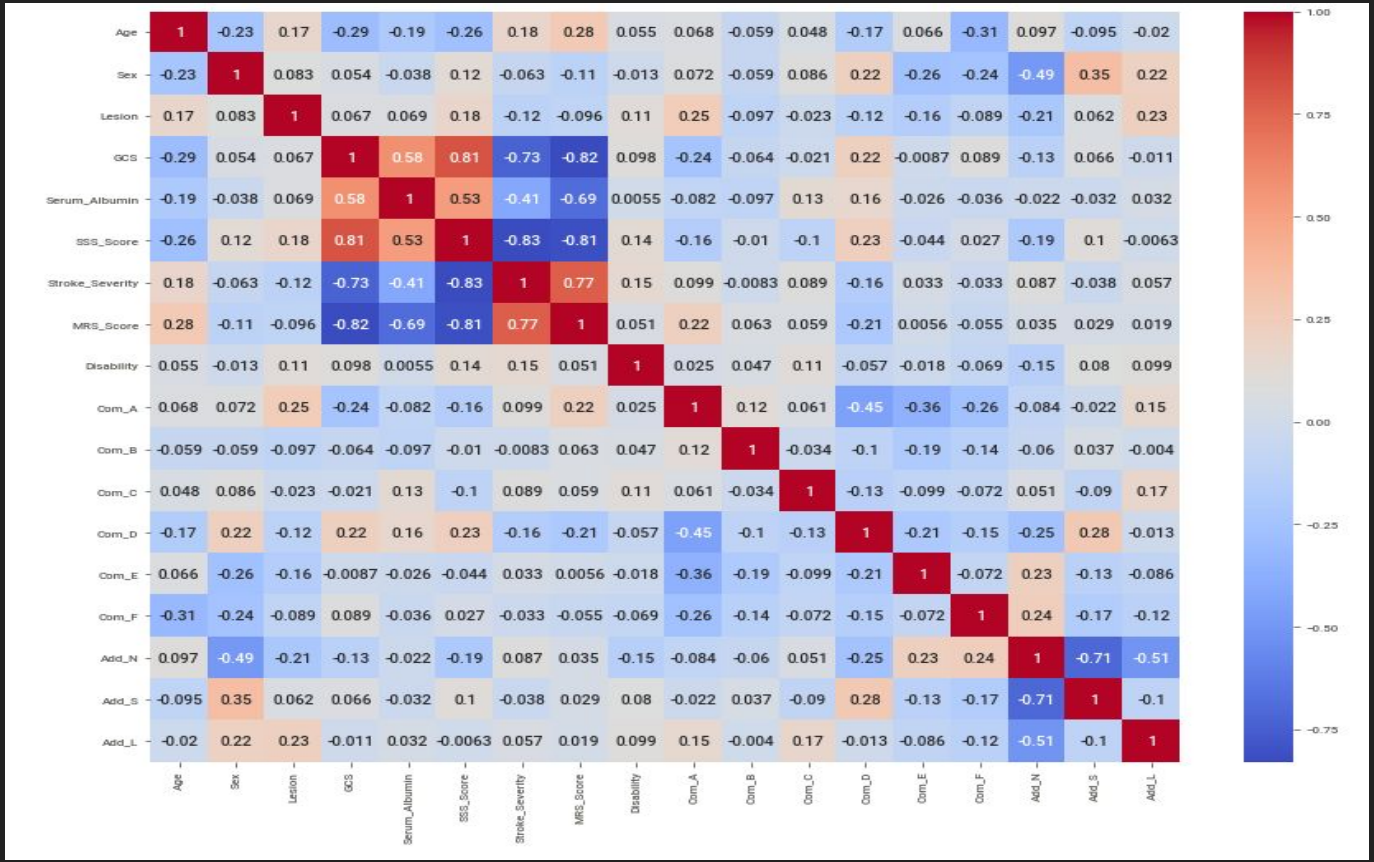
A snippet of first ten records of dataset.

Data Wrangling

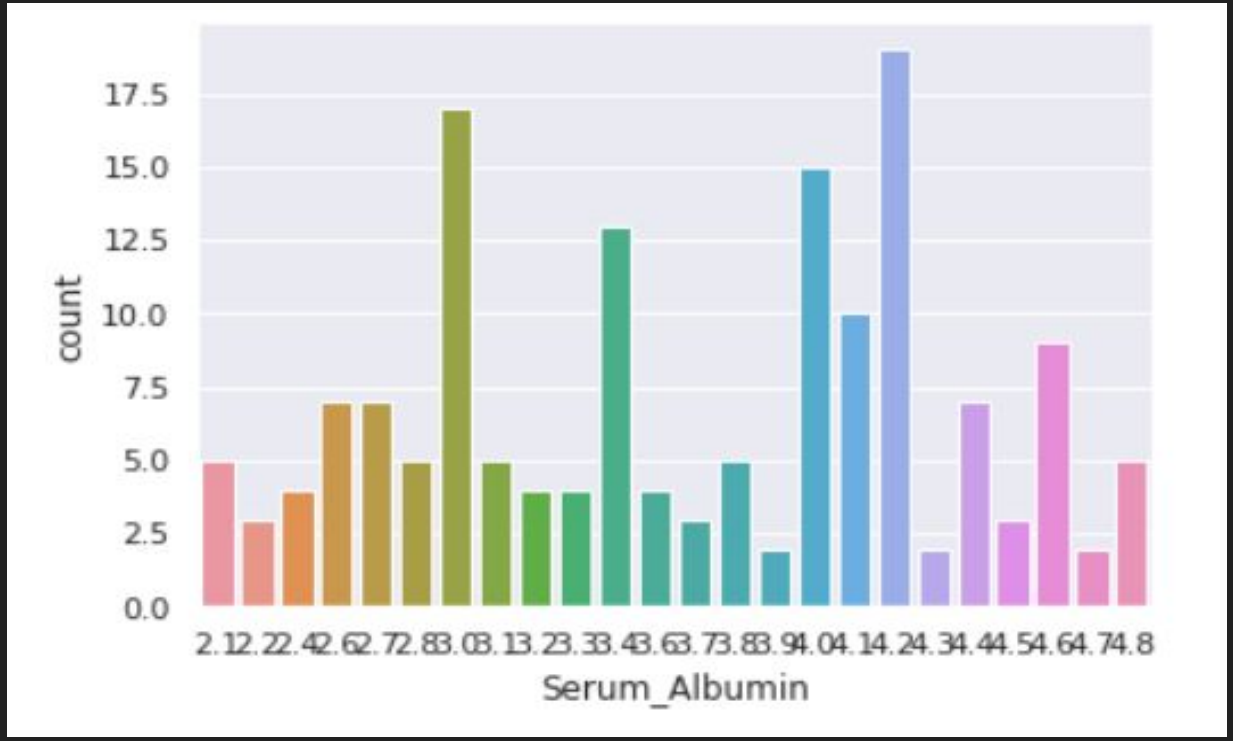
- We prepare the data to train the ML model firstly by converting the categorical columns into numerical columns.
- The data provided is of real time and it is utmost perfect without existence of missing values or duplicates.
- Then, we randomly split the data for training and testing in the ratio 75:25.
- Our next step was feature engineering, where we dropped a few insignificant columns like Name, Patient ID, Sl No, Addictions and Comorbid conditions.
- Though the data seemed to be clean, we had a troublesome time while applying One Hot Encoding to a addictions and comorbid conditions columns in the dataset.
- Label encoding has been performed on the required columns like Sex and Stroke_Severity.
- Since we had a small of 200 patients, we used k-fold cross validation so that we do not lose a portion of the data for training.

Data Visualization

- Used sweet viz, seaborn, matplotlib, pandas_profiling libraries for data visualization of the features present in dataset
- Through data visualization correlations , missing values and interactions of variables are acquired
- Heat maps are generated for entire data frame to analyze relationships between the multiple features at a higher dimensional space



Heatmap showing correlation coefficients between all features of the dataset



Distribution of Serum Albumin, a feature that greatly impacts the Stroke severity

Model Building

Models Used to predict Severity of Stroke

- Logistic Regression Classifier
- K Nearest Neighbors Classifier
- Gaussian Naive Bayes
- Bernoulli Naive Bayes
- Support Vector Classifier
- Decision Tree Classifier
- Random Forest Classification
- Extreme Gradient Boosting Classifier

Logistic Regression

- Logistic regression model identifies a relationship between the predictor variables and response categorical variables.
- It helps to estimate a probability of falling into a certain level of the categorical response based on the given set of predictors and assigning weights to each feature.

```
Logistic Regression :
[[31  0]
 [ 2  7]]
Accuracy Score:  0.95

K-Fold Validation Mean Accuracy: 96.88 %

Standard Deviation: 3.12 %

ROC AUC Score: 0.89

Precision: 1.00

Recall: 0.78

F1: 0.88
```

K-Nearest Neighbors

- K-Nearest Neighbours(KNN) is an relatively simple classifier which uses similarity function to perform classification.
- The similarity is defined according to a distance metric between two data points like euclidean distance whose mathematical formula is as follows.

$$Dist((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2}$$

```
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
```

Gaussian Naive Bayes

- A Naive Bayes classifier works basis on the Naive Bayes mathematical algorithm so the Gaussian Naive Bayes is derived as one such classifier but with Gaussian distribution.
- Gaussian is the easiest to find out the mean and the standard deviation from the training data and it is also suitable to continuous data.

```
GaussianNB :
[[17 14]
 [ 0  9]]
Accuracy Score:  0.65

K-Fold Validation Mean Accuracy: 71.25 %

Standard Deviation: 8.93 %

ROC AUC Score: 0.77

Precision: 0.39

Recall: 1.00

F1: 0.56
```

Bernoulli Naive Bayes

- Bernoulli Naive Bayes model is an acquisition of Naive Bayes classifier .
- It predicts the probability of input being classified for all the classes.
- This model is based on Bayes theorem and conditional probability.
- All features are given equal importance in this classifier. But each feature should be independent of the rest of the features.

```
BernoulliNB :
[[25  6]
 [ 4  5]]
Accuracy Score:  0.75

K-Fold Validation Mean Accuracy: 82.50 %

Standard Deviation: 11.11 %

ROC AUC Score: 0.68

Precision: 0.45

Recall: 0.56

F1: 0.50
```

Support Vector Classifier

- An SVM model is a general representation of different classes in a hyperplane in multidimensional space. Hyperplanes are generated in an iterative manner in order to reduce the error.
- The end goal of an SVM classifier is to divide the dataset into classes to find a maximum marginal hyperplane.
- SVM algorithm is implemented with kernel that transforms an input data space into the required form.

```
SVM :
[[31  0]
 [ 2  7]]
Accuracy Score:  0.95

K-Fold Validation Mean Accuracy: 96.25 %

Standard Deviation: 4.15 %

ROC AUC Score: 0.89

Precision: 1.00

Recall: 0.78

F1: 0.88
```

Decision Tree Classifier

- This tree directly aims to reduce the entropy (or simply improvement in MSE) at each node split.
- Solely focuses on achieving low bias (trying to maximally overfit the training data), whereas bagging methods like Random Forest tries to achieve low variance.

```
Decision Tree :
[[31  0]
 [ 0  9]]
Accuracy Score:  1.0

K-Fold Validation Mean Accuracy: 96.25 %

Standard Deviation: 3.06 %

ROC AUC Score: 1.00

Precision: 1.00

Recall: 1.00

F1: 1.00
```

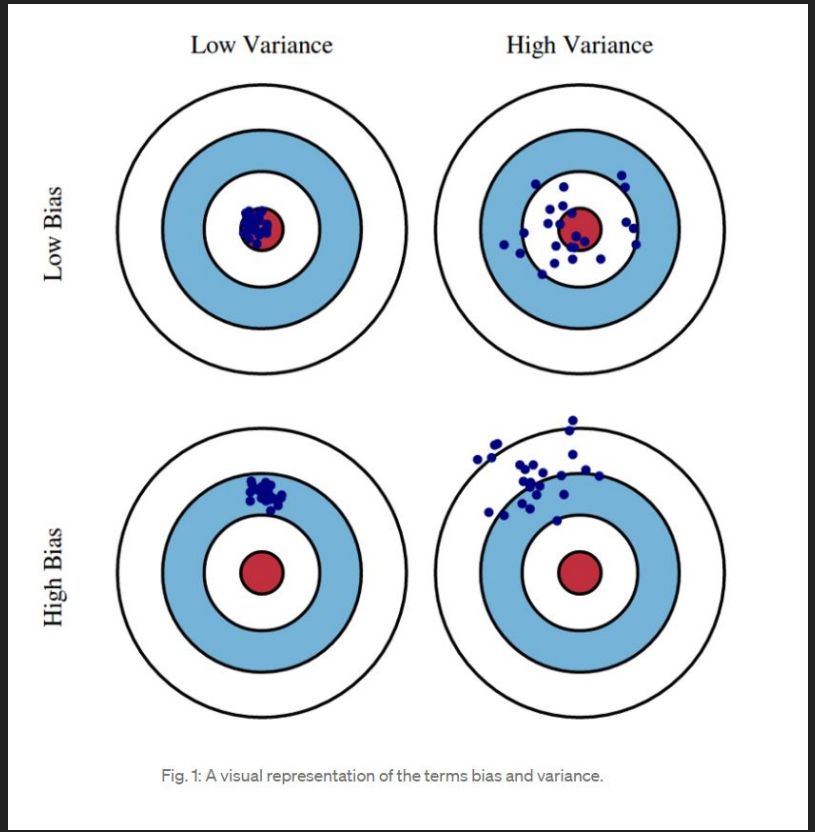


Fig. 1: A visual representation of the terms bias and variance.

Bias and Variance comparison [1]

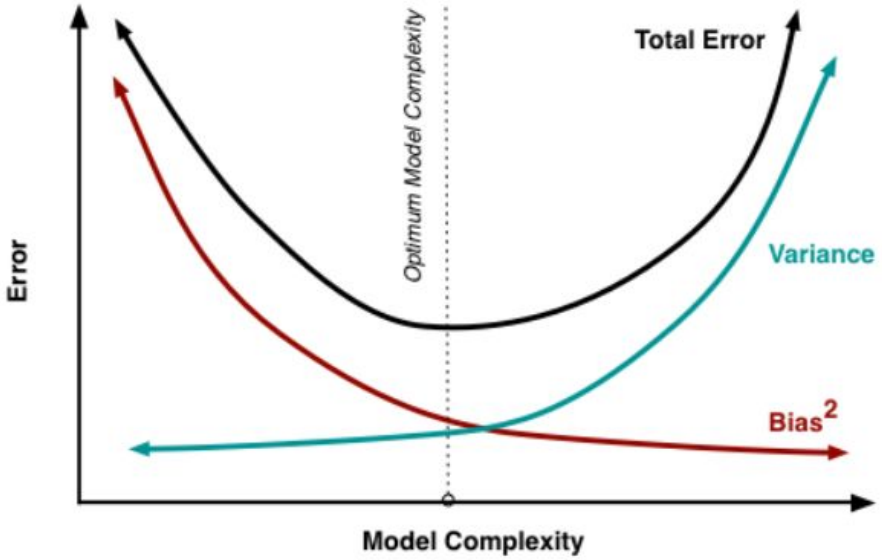


Fig. 2: A curve of squared bias vs variance showing the inverse correlation that is typical of the relation between the two as the model gets more complex. It is not uncommon for the resulting Total Error to follow some variant of the U-shape shown in the figure.

Bias - Variance tradeoff [2]

Random Forest Classifier

- Bagging method (bootstrapping) where several trees with all possible splits are considered as models (with given tree depths) and aggregated to build a model with enhanced accuracy.
- Unlike the greedy decision tree, random forest considers all possible scenarios for split at each node in the trees and aggregates with weights for trees with greater accuracy.

```
Random Forest :
[[31  0]
 [ 0  9]]
Accuracy Score:  1.0

K-Fold Validation Mean Accuracy: 96.88 %

Standard Deviation: 4.19 %

ROC AUC Score: 1.00

Precision: 1.00

Recall: 1.00

F1: 1.00
```

Extreme Gradient Boosting Classifier

- Boosting method where the output of one tree is input into another to make a stronger ensemble model.
- Several weaker models are joined together to eventually build a model with better performance.
- Typically used for accurate results in cases of high bias.

```
XGBoost :
[[31  0]
 [ 2  7]]
Accuracy Score:  0.95

K-Fold Validation Mean Accuracy: 97.50 %

Standard Deviation: 3.06 %

ROC AUC Score: 0.89

Precision: 1.00

Recall: 0.78

F1: 0.88
```

Hyper-parameter Optimization or Fine -Tuning

- These are adjustable parameters that must be tuned in order to obtain a model with optimal performance.
- For example, these include max tree depth and n estimators for random forest and entropy for decision trees.
- A parameter grid will be built which includes a range of possible values for the said hyperparameters in order to check at which value the maximized accuracy can be obtained by the model.

Grid Search CV

- A tuning technique that attempts to compute the optimum values of hyperparameters.
- This method takes in grids with designated spacings and computes the accuracy of the ML models at all these grid points to arrive at the best model and makes a note of its hyperparameter.

```
---------------

DecisionTreeClassifier():
Best Accuracy : 96.88%
Best Parameters :  {'criterion': 'entropy', 'random_state': 0}

---------------

RandomForestClassifier():
Best Accuracy : 96.88%
Best Parameters :  {'criterion': 'gini', 'n_estimators': 100, 'random_state': 0}

---------------

XGBClassifier():
Best Accuracy : 97.50%
Best Parameters :  {'eval_metric': 'error', 'learning_rate': 0.1}

---------------
```

Creating an Ensemble out of the tuned ML models…

We now have the hyper-parameter optimized ML models to make predictions.

BUT WHAT IF WE COULD GET MUCH IMPROVE THE ACCURACY BEYOND WHAT THESE MODELS CAN ACHIEVE ALONE?

This is a method of boosting where multiple weak models (lesser accuracy) are considered and boosted by aggregating these less accurate models to end up with better predictions than all the weaker models.

The Ensemble Model - A Voting Classifier

- A Voting Classifier is a machine learning model that trains on an ensemble of various other models and predicts an output based on their highest probability of chosen class as the output.
- It aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting.
- Soft voting the output class is the prediction based on the average of probability given to that class.
- NOTICE HOW THE ACCURACY, F1 & RECALL INCREASED COMPARED TO OUR BEST MODEL (XGBOOST)?

```
[[31  0]
 [ 1  8]]

Accuracy Score:  0.975

K-Fold Validation Mean Accuracy: 97.50 %

Standard Deviation: 3.06 %

ROC AUC Score: 0.94

Precision: 1.00

Recall: 0.89

F1: 0.94
```

Analysis of this Study

- From the correlation heatmap, we find that Stroke Severity is very strongly correlated to GCS (Glasgow Coma Scale) and SSS (Siriraj Stroke Score).
- Also, we find that the Stroke severity is dependent on a few other factors like Serum Albumin, age and MRS Score (Modified Rankin Score).
- This study also supports to some extent that there is a correlation between nicotine addiction and the stroke severity.
- Also, the Disability which is an after-effect of the stroke, seems to be moderately correlated to the Stroke severity.
- It might seem that the data is imbalance as there are no patients with smaller data. However, it is unlikely that such a stroke is observed in the younger population.

Inferences from this Project

- An array of 9 ML models were created and trained on the training data to make predictions.
- XGBoost Classifier shows max accuracy among all the models due to its boosting technique (but could also lead to overfitting).
- We then move on to build an ensemble model (Voting Classifier) using the ML models that are better predictors (those with higher accuracy).
- This ensemble model eventually serves the purpose of achieving better F1 score, accuracy, precision and recall.

	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

Summary of all 8 ML models

Conclusion

- Through analysis and data visualisation it was understood the various unheard features impacting the severity of a brain stroke.
- In this project machine learning ensemble is build to perform classification through which we attained at 97.5% accuracy in predicting the different levels of severity of it like mild, severe and death.
- Generally in projects related to healthcare the expected accuracy is 99% but we tried to achieve high accuracy with just 200 records of data by combining five well performing machine learning classifiers.
- Future possibility is to build and train various other models and gather a larger dataset to create a more powerful ensemble in order to attain higher accuracy.

References

1. Random Forests and the Bias-Variance Tradeoff | by Prratek Ramchandani
2. Bias-Variance Tradeoff | TowardsDataScience

Thank You