Stroke Prediction

[1]Ezedin Mohammed, [2]Goutham Thota, [3]Jaya Surya Thota

Abstract

Stroke is a devastating disease that affects millions of people worldwide and is one of the main causes of death. The likelihood of a patient recovering from a stroke can be considerably increased by early detection and treatment. Traditional stroke prediction techniques, however, demand a lot of manual labor and feature selection. In this study, we use machine learning techniques to predict the risk of stroke. To correctly predict the existence of stroke, we use some classifiers, including K-Nearest Neighbour (KNN), Random Forest, Decision Trees, Support Vector Machines, and Extreme Gradient Boosting (XGB). Random Forest outperformed the other models with an accuracy of 89%. The findings imply that machine learning algorithms have the ability to effectively predict stroke risk and improve patient outcomes by detecting and treating strokes early.

Introduction

Stroke is a terrible illness that kills millions of people annually and is one of the main causes of mortality. The World Health Organization (WHO) estimates that stroke causes 6.3 million fatalities yearly. In addition to the mortality, stroke frequently causes long-term disability and lowers the quality of life for survivors. To enhance patient outcomes and lessen the impact of stroke on society, early detection and timely treatment of stroke is therefore crucial.

Millions of individuals worldwide suffer from the dangerous medical condition of stroke, which is one of the main causes of death. The World Health Organization (WHO) estimates that stroke causes 11% of all fatalities worldwide. When the blood flow to a portion of the brain is cut off, a stroke happens, which can cause death or severe brain damage. Stroke can have serious side effects, such as memory loss, paralysis, and difficulty speaking. Early stroke diagnosis and care can greatly increase a patient's chances of recovery and lessen the condition's long-term repercussions.

Age, blood pressure, and cholesterol levels are a few clinical characteristics that are used in traditional stroke prediction methods to assess the risk of stroke. These techniques are time-consuming and ineffective since they need a lot of manual work and feature selection. We suggest using machine learning approaches to estimate the chance of stroke to solve this problem.

In this study, we use a variety of machine learning classifiers to precisely predict the occurrence of stroke, including K-Nearest Neighbour (KNN), Random Forest, Decision Trees, Support Vector Machines (SVM), and Extreme Gradient Boosting (XGB). These classifiers have a reputation for handling complex data and providing precise predictions. We use a dataset of patient records to train these classifiers. This dataset contains clinical variables including age, gender, blood pressure, smoking status, and other illnesses.

Our main goal is to assess the effectiveness of various classifiers and identify which one offers the most precise stroke risk prediction. Based on each classifier's accuracy, sensitivity, and specificity, we compare the findings. The results of this study will have a big impact on stroke management and prediction, which will improve patient outcomes.

Despite the great potential of machine learning for stroke prediction, there are still a number of obstacles and restrictions that need to be overcome. First, in order to obtain high accuracy and generalizability, ML models need big and varied data sets. However, the quantity and scope of stroke data sets are frequently constrained, which may have an impact on how well ML models function. Second, ML models may be overfitted or underfitted, which can result in incorrect forecasts. In order to guarantee the validity and reliability of ML predictions, adequate model selection and validation are essential.

Related Work

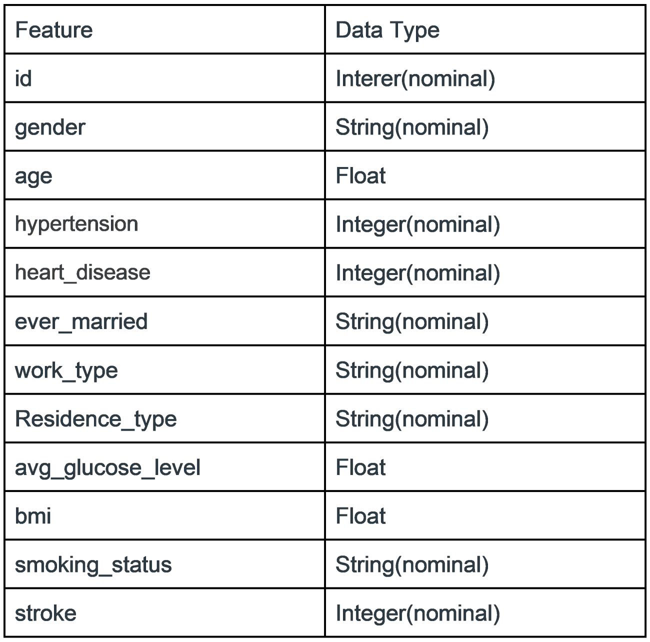
TensorFlow, a popular tool for large-scale machine learning that has been utilized in numerous studies on stroke prediction, is described in detail by Abadi et al. in [2]. The scalability and flexibility of TensorFlow are highlighted by the authors.

The effectiveness of different machine learning techniques for stroke prediction, such as decision trees, k-nearest neighbor, and support vector machines (SVM), is examined by Sailasya and Kumari [3] and Emon et al. [4]. Naive Bayes performed better than other classification algorithms in terms of accuracy, precision, recall, and F1, according to Sailasya and Kumari [3], while Emon et al. [4] employed weighted voting, which had the greatest accuracy of 97% when compared to other machine learning techniques that are frequently used.

According to Jeena and Kumar [5], who concentrate exclusively on the application of SVM for stroke prediction, identifying stroke risk factors has a high accuracy rate. For different kernel functions, the authors utilized an SVM-based technique, and linear function accuracy was 91%. According to the authors, SVM can be a useful tool for predicting the possibility of having a stroke, especially when combined with more clinical and demographic information.

Data

The dataset is **Stroke Prediction Dataset** which was obtained from Kaggle. The Dataset has 5110 observations and 12 features total and Stroke is our target variable. We evaluate responsiveness by stroke. A significant imbalance exists in our Stroke variable.



Feature	Data Type
id	Interer(nominal)
gender	String(nominal)
age	Float
hypertension	Integer(nominal)
heart_disease	Integer(nominal)
ever_married	String(nominal)
work_type	String(nominal)
Residence_type	String(nominal)
avg_glucose_level	Float
bmi	Float
smoking_status	String(nominal)
stroke	Integer(nominal)

Table: 1

Data Preprocessing

Data preprocessing involves removing outliers based on the interquartile range, addressing missing values using the statistical method, and creating a correlation matrix to determine the relationship between the variables.

As the ”id” column was unique to each patient, we removed it. There are three categories in the gender column: male, female, and others. The ”other” class is only present in one row in our data. So, we removed that observation. In our dataset, there are no duplicate values. The Body Mass Index(BMI) column is missing 201 values.

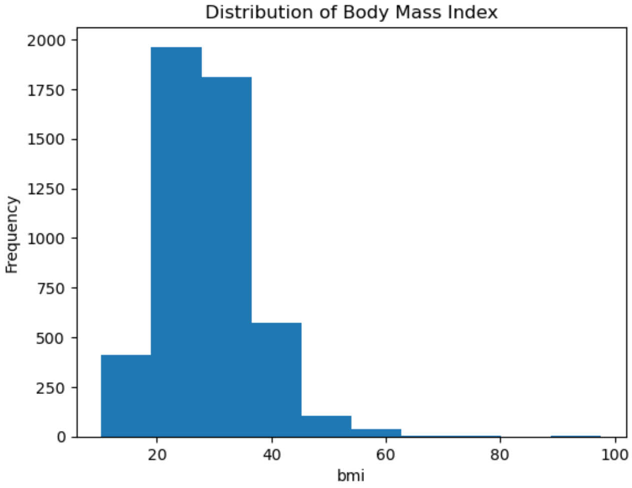


Figure 1: Distribution of Body Mass Index

We can see from the above distribution that most of the data lies from 20 to 40. So, we used the mean to replace the missing values.

We look for outliers to see if our variables have any bad data or not. To better illustrate the outliers for our three numerical variables vs stroke, Box plots are shown below (Figure 2).

These are BMI, age, and average glucose level. Outliers may exist in BMI and average glucose levels as a result of things like genetic predisposition, metabolic diseases, or lifestyle choices. The removal of these outliers could result in the loss of crucial data, which might have an effect on how accurate predictions are made or how well underlying data patterns are understood.

In addition, outliers are typically tolerated and handled well by machine learning techniques like decision trees, random forests, and support vector machines.

Having stated that we did not remove outliers.

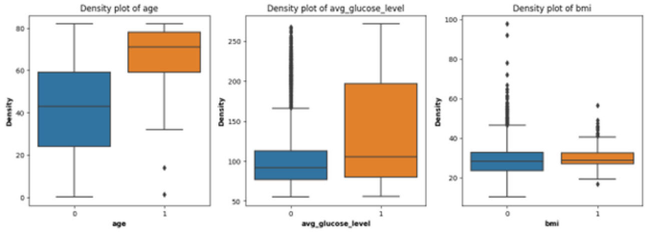


Figure 2: Box plot for outliers

Figure 3 shows the heatmap after preprocessing to help see the relationship; it shows that there is no association between the predictor variables.

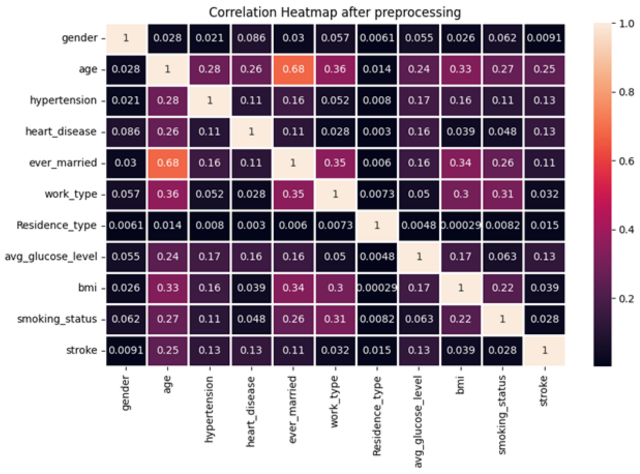


Figure 3: correlation heatmap after preprocessing.

We used bar plots (Figure 4) against each categorical variable to determine the balance of our response variable and discovered it is severely unbalanced.

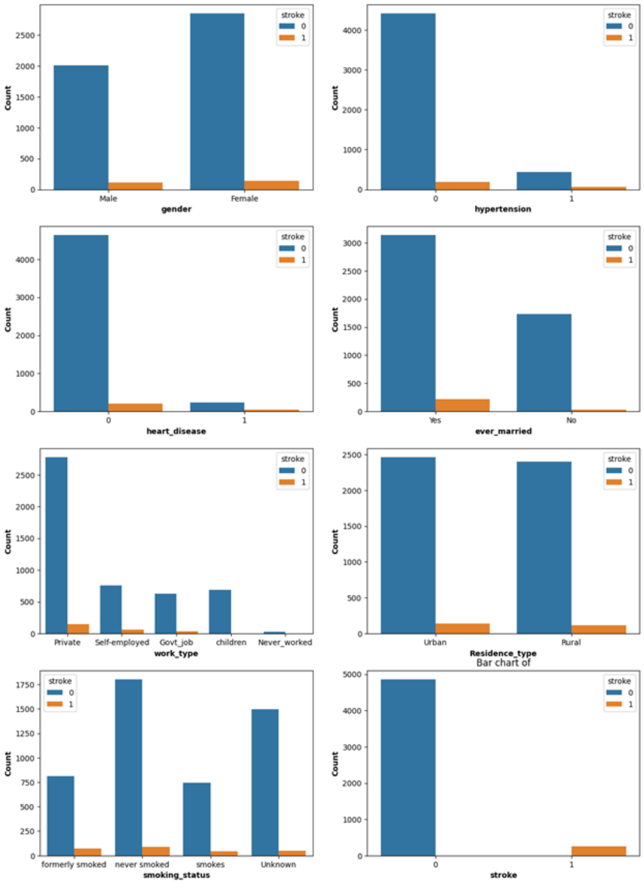


Figure 4: Bar plot of categorical variables.

Label encoding

We utilized label encoding to change the category variables into dummy variables. This function was used for the following variables: gender, ever-married status, work type, residence type, and smoking status.

SMOTE

A common data augmentation technique for imbalanced classification issues is SMOTE (Synthetic Minority Oversampling Technique). By interpolating between examples of the minority class that already exist, it creates synthetic samples from the minority class. To create new instances, the algorithm first chooses a minority instance and locates its k nearest neighbors. The method then linearly interpolates between the chosen instance and its neighbors.

SMOTE is helpful because it may be used to solve the issue of imbalanced data when there are significantly fewer samples in one class than in another. Machine learning models may perform poorly as a result of unbalanced data, particularly for the minority class. By increasing the number of samples in the minority class, SMOTE can aid, which will enhance model performance. Additionally, SMOTE can also help to reduce overfitting and improve model generalization.

So, we have used the SMOTE technique to make our target variable balance.

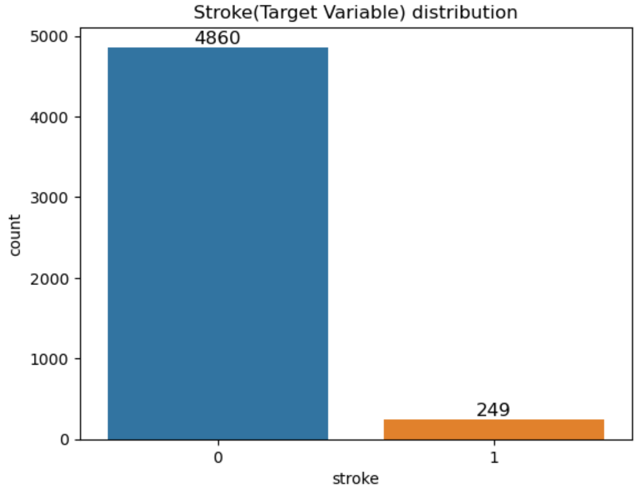


Figure 5: Distribution of Stroke

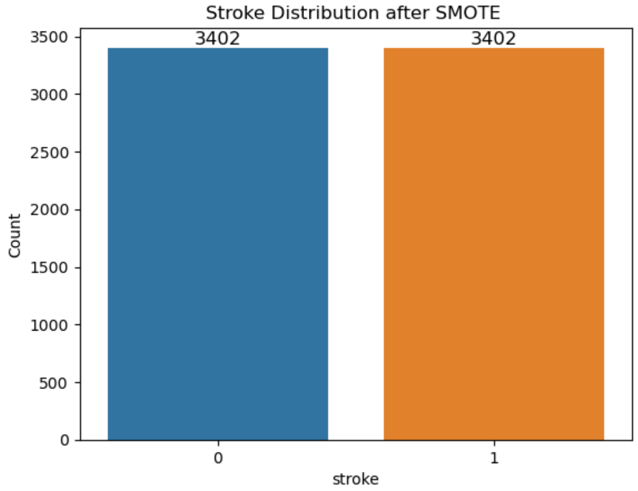


Figure 6: Distribution of Stroke(SMOTE)

Methods

The K-Nearest Neighbour (KNN), Random Forest, Decision Trees, Support Vector Machines, and Extreme Gradient Boosting (XGB) are the five classification models we are currently applying.

Support Vector Machines(SVM)

A deep learning algorithm known as a support vector machine (SVM) uses supervised learning to classify or predict the behavior of groups of data. Supervised learning systems in AI and machine learning provide input and desired output data that are labeled for classification. A learning foundation for upcoming data processing is provided by the classification. Two data groups are sorted using support vector machines according to similar classification. The algorithms create hyperplanes (lines) to divide the groups into various configurations. An SVM needs labeled training data, just like other supervised learning machines. Materials are labeled in groups for classification. SVM training materials are organized into distinct groups and classified separately at various locations in space. SVMs can perform unsupervised learning after processing a large number of training examples.

K-Nearest Neighbour

The supervised learning algorithm K-nearest neighbors (KNN) is used for both regression and classification. By calculating the distance between the test data and all of the training points, KNN tries to predict the correct class for the test data. Then choose the K points that are closest to the test data. The KNN algorithm determines which of the classes of the 'K' training data the test data will belong to, and the class with the highest probability is chosen. The value in a regression situation is the average of the 'K' chosen training points. Both regression and classification predictive issues can be solved using KNN. However, because it performs well across all parameters when applied to industrial problems, classification is where it is most frequently used.

Extreme Gradient Boosting (XGB)

XGBoost is a distributed gradient boosting library that has been optimized for quick and scalable machine learning model training. A number of weak models' predictions are combined using this ensemble learning technique to create a stronger prediction. Extreme Gradient Boosting, or XGBoost, is one of the most well-known and widely used machine learning algorithms because it can handle large datasets and perform at the cutting edge in many machine learning tasks like classification and regression.

Its effective handling of missing values, which enables it to handle real-world data with missing values without requiring a lot of pre-processing, is one of the key characteristics of XGBoost. Additionally, XGBoost includes built-in parallel processing support, enabling the training of models.

Random Forests

The hyperparameters of a random forest are very similar to those of a decision tree or a bagging classifier. Fortunately, using the classifier class of random forest eliminates the need to combine a decision tree with a bagging classifier. Using the algorithm's regressor, you can use the random forest to handle regression tasks as well.

While the trees are growing, the random forest adds more randomness to the model. When splitting a node, it looks for the best feature among a random subset of features rather than the most crucial one. A better model is generally produced as a result of the wide diversity this causes.

Decision Trees

Using a set of features, a decision tree is a supervised learning technique that can be used to categorize or predict outcomes. The data is recursively divided into smaller and smaller groups until each group only comprises individuals belonging to the same class. Beginning at the root node, which stands for the complete dataset, the decision tree is built. The two child nodes that result from the division of the root node each represent a portion of the dataset. Up until each leaf node in the tree represents a single data point, this process is done recursively.

With the help of a dataset containing labeled data, the decision tree is trained. A set of features and an associated class label make up the labeled data.

Model Assessment

To identify the most effective model, the evaluation of machine learning models needs to be validated. Accuracy, precision, and recall are considered in the evaluation.

Experiments and Results

Flowchart

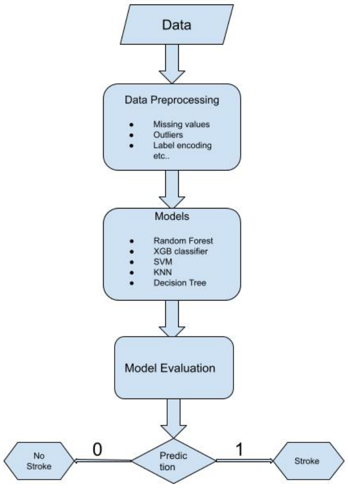
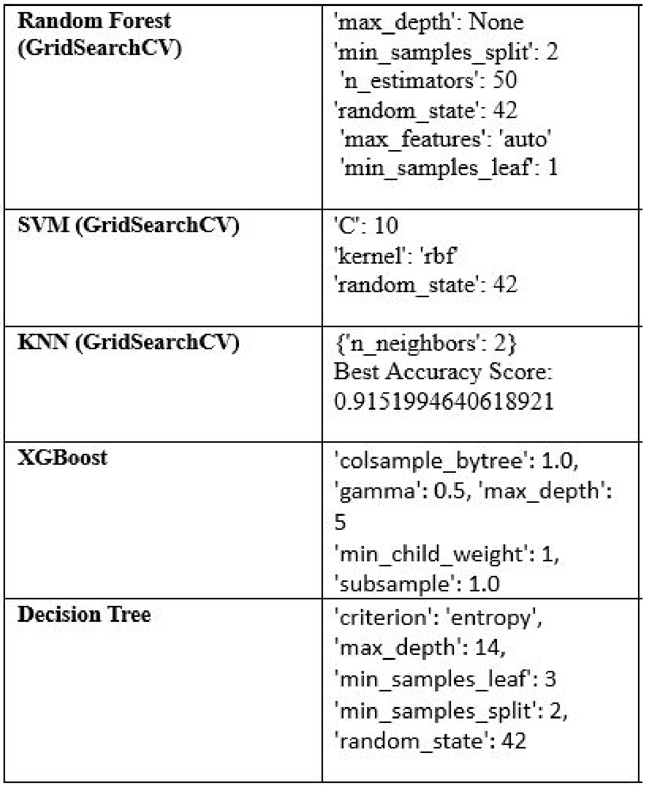


Figure 7: Flowchart

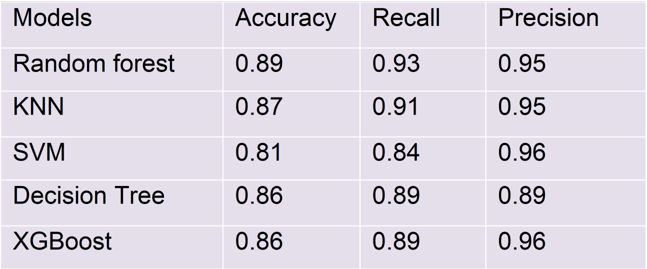
After pre-processing, the first step was to separate the data into training and testing sets. We used a 70:30 ratio to split our data. We choose Stratified 5-fold cross-validation as the method of resampling because of the imbalanced nature of our response variable and to maintain the distribution of the classes. It would split the data so that roughly the same amount of classes would be present in each fold that was created. Predictions were produced using these values on the testing set after all the models were hyper-tuned using the GridSearchCV approach to identify their ideal parameters. These are the best hyperparameters we got for respective models.



Random Forest (GridSearchCV)	'max_depth': None 'min_samples_split': 2 'n_estimators': 50 'random_state': 42 'max_features': 'auto' 'min_samples_leaf': 1
SVM (GridSearchCV)	'C': 10 'kernel': 'rbf' 'random_state': 42
KNN (GridSearchCV)	{'n_neighbors': 2} Best Accuracy Score: 0.9151994640618921
XGBoost	'colsample_bytree': 1.0, 'gamma': 0.5, 'max_depth': 5 'min_child_weight': 1, 'subsample': 1.0
Decision Tree	'criterion': 'entropy', 'max_depth': 14, 'min_samples_leaf': 3 'min_samples_split': 2, 'random_state': 42

Figure 7: Best Hyperparameters

We used these parameters (Figure 7) to predict the test data and reported accuracy, precision, and recall for all the models.



Models	Accuracy	Recall	Precision
Random forest	0.89	0.93	0.95
KNN	0.87	0.91	0.95
SVM	0.81	0.84	0.96
Decision Tree	0.86	0.89	0.89
XGBoost	0.86	0.89	0.96

Figure 8: Classification Report Table

Accuracy Comparison

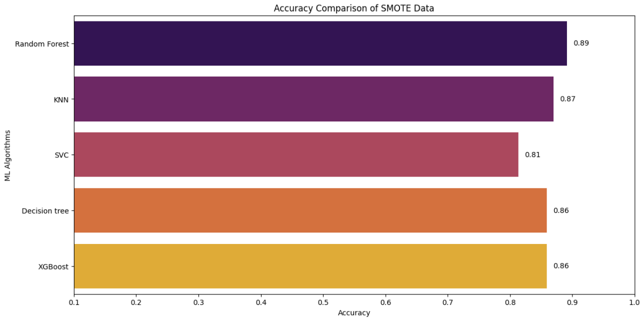


Figure 9: Accuracy comparison for all Models

Conclusion

We can conclude that the Random Forest model beat the other models in terms of accuracy, recall, and precision based on the results of the models examined using SMOTE data. The accuracy ratings for the KNN and Decision Tree models were 0.86 and 0.87, respectively, and both models performed well. While XGBoost had a similar accuracy score and recall score to the Decision Tree model but had a higher precision score, SVM had lower accuracy and precision scores compared to the other models.

Future research could examine several data augmentation methods to see which one is more effective for this specific dataset. ADASYN, Borderline SMOTE, and Synthetic Minority Over-sampling TEchnique with Iterative procedure (SMOTE-IPF) are a few other techniques that could be investigated to see if they perform better than SMOTE, which is a well-known and useful data augmentation technique for imbalanced datasets.

Examining the usage of ensemble models, which aggregate the predictions of various models to improve overall performance, is another potential topic of future research. This could be accomplished by merging the best-performing models' outputs or by employing stacking approaches, which train many models and utilize their outputs as input for a meta-model that learns to integrate them effectively.

According to the results of the evaluation of models employing SMOTE data, the Random Forest model performed the best overall, closely followed by the KNN and Decision Tree models. To further enhance the performance of the models and obtain a better understanding of the elements that affect the chance of stroke, future study could concentrate on feature engineering, hyperparameter tuning, alternative data augmentation strategies, ensemble models, and model interpretability.

References

[1]E Cuadrado-Godia, P Dwivedi, S Sharma, Santiago A Ois, Gonzalez J Roquer, M Balcells et al., ”Cerebral small vessel disease: a review focusing on pathophysiology biomarkers and machine learning strategies” in J Stroke, Random forests. Machine Learning, vol. 20, pp. 302-320, 2018. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6186915/

[2]M Abadi, P Barham, J Chen, Z Chen, A Davis, J Dean et al., ”Tensorflow: a system for large-scale machine learning”, OSDI, vol. 16, pp. 265-283, 2016. https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[3]Gangavarapu Sailasya and Gorli L Aruna Kumari, “Analyzing the Performance of Stroke Prediction using ML Classification Algorithms” International Journal of Advanced Computer Science and Applications (IJACSA), 12(6), 2021.https://thesai.org/Publications/ViewPaper?Volume=12 &Issue=6 &Code=IJACSA&SerialNo=62

[4]M. U. Emon, M. S. Keya, T. I. Meghla, M. M. Rahman, M. S. A. Mamun, and M. S. Kaiser, ”Performance Analysis of Machine Learning Approaches in Stroke Prediction,” 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1464-1469, doi: 10.1109/ICECA49313.2020.9297525. https://ieeexplore.ieee.org/abstract/document/9297525

[5]R. S. Jeena and S. Kumar, ”Stroke prediction using SVM,” 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, India, 2016, pp. 600-602, doi: 10.1109/ICCICCT.2016.7988020. https://ieeexplore.ieee.org/abstract/document/7988020