Stroke Prediction using ML Algorithms

Vedasamhitha Challapalli, R.Amshu Naik, Perumalla Aasrish Vinay

B20CS078 , B20CS046 , B20CS042

Abstract – **The paper reports our experience and efforts in dealing with the dataset of stroke prediction which is basically a machine learning classification problem. I have basically used 7 classification models along with other essential Machine Learning concepts. The ultimate goal of taking up the project is to suggest the best model which predicts stroke given various inputs about the health condition.**

I. INTRODUCTION

According to the statistics mentioned by the World Health Organization (WHO), stroke is the 2nd largest cause of death contributing to 11% of the death rate. And it is a classification problem which makes the research more interesting because there are many algorithms for classification problems and even the prediction rate is more accurate for classification problems. That's why we are going to use machine learning to solve this problem.It will somehow help in decreasing the death rate due to stroke.

Dataset

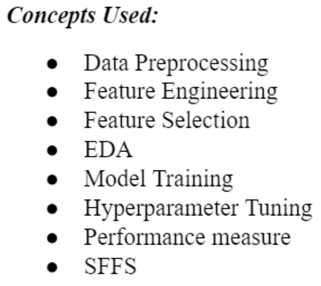
GAN : The file " *healthcare-dataset-stroke-data.csv*" is used as the training dataset.
The train dataset contains 5110 rows and 12 columns, here is a bit description of our dataset:

- Features in our dataset: id, gender, age, hypertension, heart disease, ever married, work type, residence type, BMI, average glucose level and smoking status.
- 1 id column
- 1 stroke column
- 10 latent vector column
- The dataset has been split into train and test with test size of 0.3.

II. METHODOLOGY

Models Used:

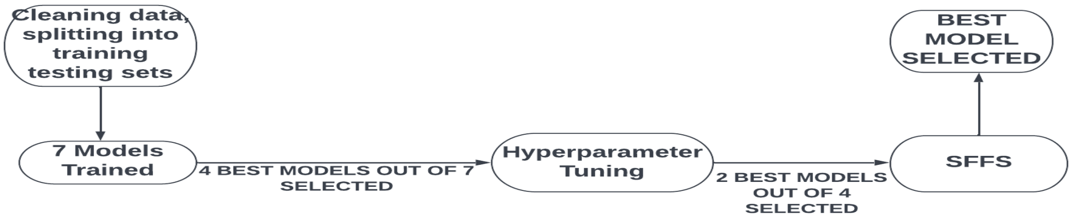
- Logistic regression
- Random Forest Classifier
- Decision Tree Classifier
- Support vector machine (SVM)
- K-nearest neighbor (KNN)
- Naive bayes
- XGB



Concepts Used:

- Data Preprocessing
- Feature Engineering
- Feature Selection
- EDA
- Model Training
- Hyperparameter Tuning
- Performance measure
- SFFS

Process Followed…


Cleaning data, splitting into training testing sets
7 Models Trained
4 BEST MODELS OUT OF 7 SELECTED
Hyperparameter Tuning
2 BEST MODELS OUT OF 4 SELECTED
SFFS
BEST MODEL SELECTED

III.EXPLORING AND CLEANING THE DATASET

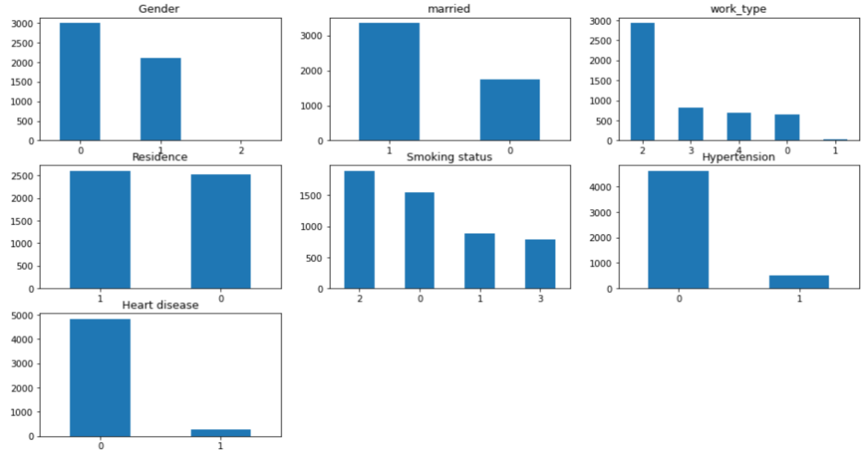
The dataset has 5110 rows and 12 columns.Out of all the columns 5 are object data type,4 are int data type, and 3 are float data type.The 12th column is of "stroke" is our target column.

Preprocessing

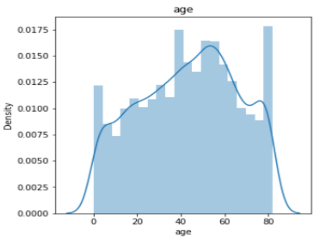
- Empty data has been replaced by mean of the column
- Id column removed
- Label Encoding(Feature Engineering) of the object data type columns/ categorical columns
- Feature Selection
- Splitted dataset into train and test data

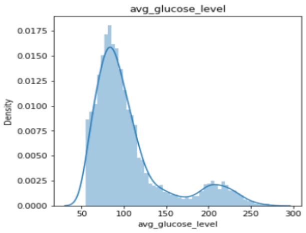
Exploratory Data Analysis:

Conclusions based on EDA:

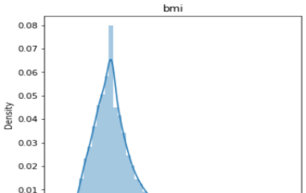


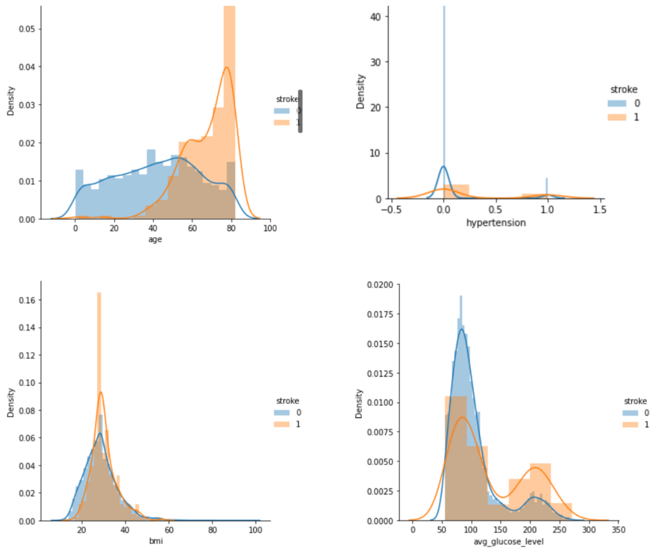
- In the given dataset, there are more people living in rural areas, more people smoking, more people in the private sector, more people in urban areas, more people with hypertension and less people with heart diseases.



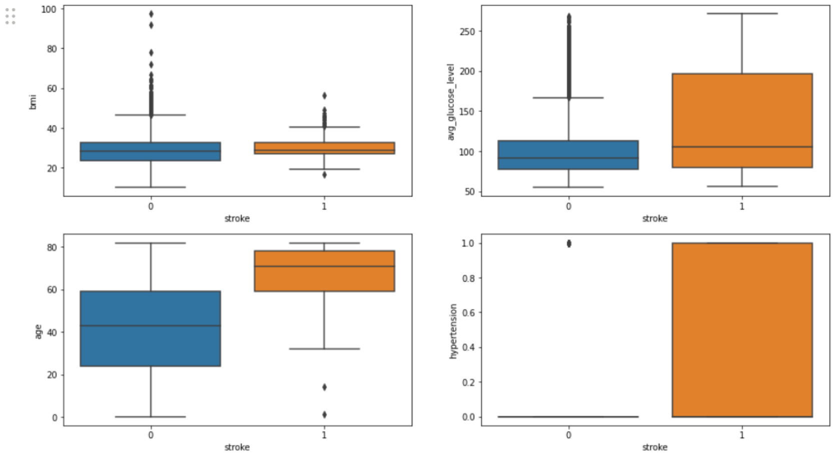


- There are more people from 40-60 age groups
- Average glucose level around 100 is the most common
- BMI of around 30 is the most common.

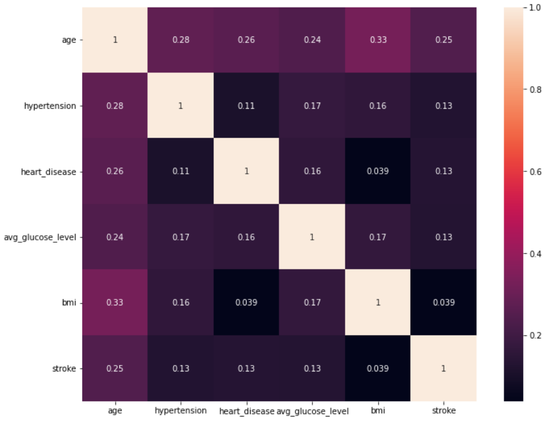




- Ages 60-80 are more prone to stroke than other ages
- Hypertension alone cannot predict stroke.
- BMI around 20-40 is both prone and not prone to stroke, so BMI alone cannot predict stroke.
- Stroke cannot be predicted by average glucose level alone as the distribution is random.



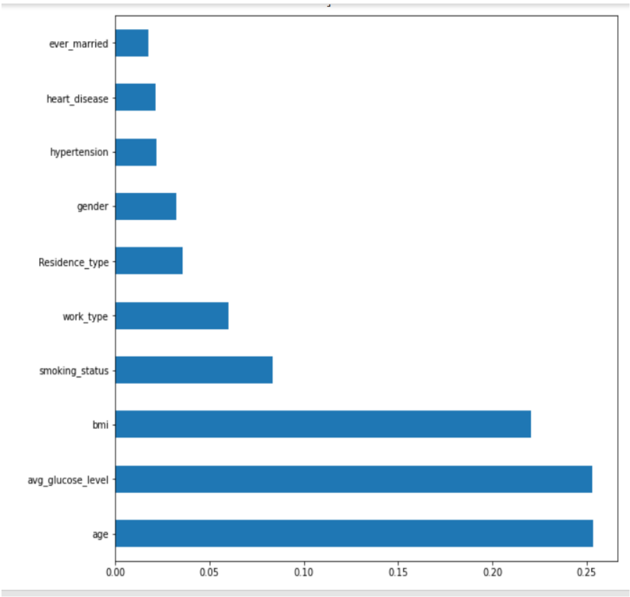
- There are outliers beyond 150 in avg_glucose_level
- BMI above 40 are outliers



- None of the features are too interrelated to each other that we cannot neglect any feature.

Feature Selection:

Conclusion: All features are important from the below graph



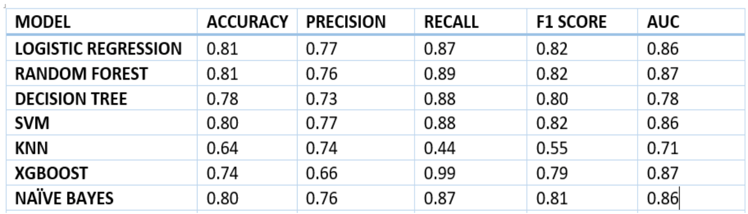
IV. IMPLEMENTATION OF CLASSIFICATION ALGORITHMS

1. ***Logistic regression:*** Logistic Regression Model is mainly used when the data is binary or just two outcomes are there hence, we used the model
    - *Hyperparameter Tuning:* Grid Search CV used to find best parameters
2. ***Random Forest Classifier***: Random Forest Classifier Model takes the average of all the votes of different trees to finally select the class. The advantage is, in this method if there is error, the tree will be overshadowed by the other trees. Due to this advantage, this model has been chosen.
    - *Hyperparameter Tuning:* Randomized Search CV used to find best parameters
    - *Sequential Feature Selection:* SFFS has been used to improve the performance.
3. ***Decision Tree Classification***: We chose Decision Tree Classifier as the most widely used Classification Model is due to its ability to break complex data of high complexity into smaller branches thus reducing complexity.
    - *Hyperparameter Tuning:* Grid Search CV used to find best parameters
4. ***Support vector machine:*** Support Vector Machine is a supervised machine learning algorithm which used kernels that transform data and based on the transformations, optimal boundary is found between possible outputs.We used linear SVM because from pairplot, we can separate classes linearly.
5. ***KNN*:**K-Nearest Neighbors (KNN) is a supervised learning algorithm which classifies based on class most common among its k nearest neighbors. Here k is the number of nearest neighbors to be considered in the majority voting process.
    - We used k = [5,7,10,13,15] to check the optimal k value for classification.
6. ***Naive bayes*:**Naive Bayes is a probabilistic machine learning classifier. It is based on Bayes Theorem. Because the classifier is based on probabilities, we chose the model.
    - *Hyperparameter Tuning:* Grid Search CV used to find best parameters
7. ***XGB*:**XGBoost (Extreme Gradient Boosting), is a gradient boosted decision tree, it provides parallel tree boosting. It is a supervised machine learning algorithm hence, we chose the model.
    - *Hyperparameter Tuning:* Grid Search CV used to find best parameters
    - *Sequential Feature Selection:* SFFS has been used to improve the performance.

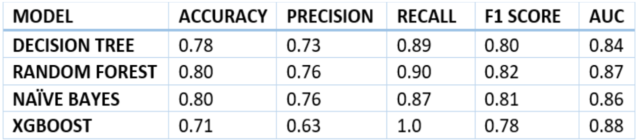
V. EVALUATION OF MODELS

Before Parameter Tuning

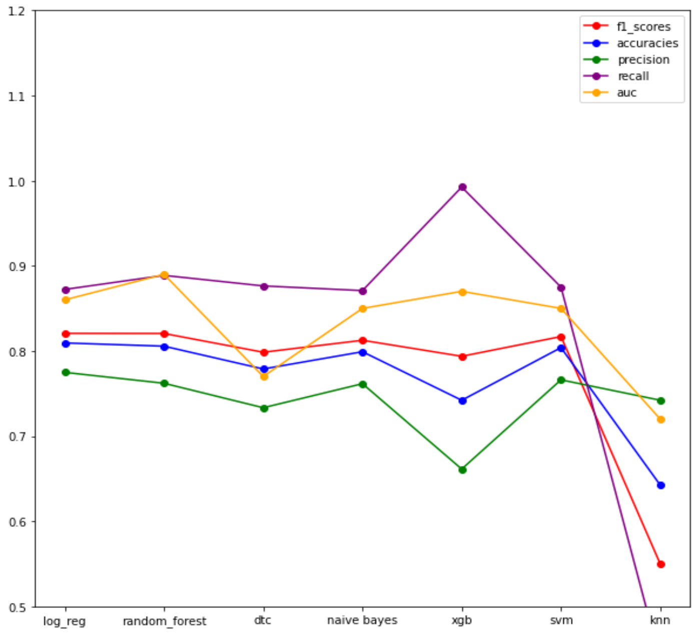
After Parameter Tuning

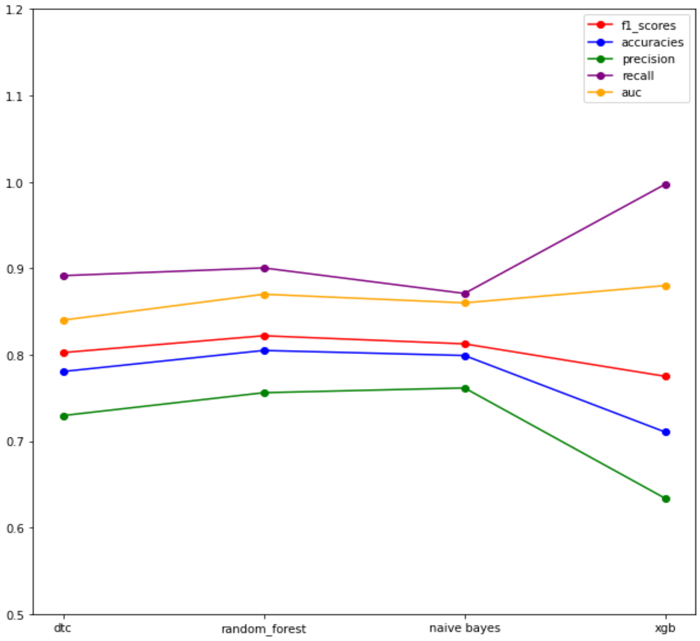


MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE	AUC
LOGISTIC REGRESSION	0.81	0.77	0.87	0.82	0.86
RANDOM FOREST	0.81	0.76	0.89	0.82	0.87
DECISION TREE	0.78	0.73	0.88	0.80	0.78
SVM	0.80	0.77	0.88	0.82	0.86
KNN	0.64	0.74	0.44	0.55	0.71
XGBOOST	0.74	0.66	0.99	0.79	0.87
NAÏVE BAYES	0.80	0.76	0.87	0.81	0.86



MODEL	ACCURACY	PRECISION	RECALL	F1 SCORE	AUC
DECISION TREE	0.78	0.73	0.89	0.80	0.84
RANDOM FOREST	0.80	0.76	0.90	0.82	0.87
NAÏVE BAYES	0.80	0.76	0.87	0.81	0.86
XGBOOST	0.71	0.63	1.0	0.78	0.88





VI. ANALYSIS AND CONCLUSIONS

- ➔ We analyzed the comparison plot we plotted of all the7 models we used.
- ➔ After comparing f1 score, accuracies, precision, recall, auc values we choose 4 models for hyperparameter tuning to get much better results.
- ➔ These 4 models are Decision tree classifier, random forest classifier, naive bayes and xgb.
- ➔ Hyperparameter tuning was done using Grid Search CV & Randomized Search CV.
- ➔ We then analyzed the comparison plot we plotted for the 4 models after the parameter tuning. After comparing their performance, we selected 2 models for Sequential Feature Selection (SFFS).
- ➔ After doing SFFS we got F1 score of *0.96012 for Random forest classifier* and F1 score as *0.9555 for XGBoost classifier.*

We have considered F1 Score to be our prime performance evaluator, as our prediction is a medical condition where we look for highly sensitive classifiers, but sensitivity (recall) turned out to be high for all major classifiers, and hence we have taken F1 score as criteria, as it includes the effect of sensitivity (recall) in it. While not only considering F1 score, we have also taken ROC as an important criteria for the purpose. It is ideal to maximize the true positive rate while minimizing the false positive rate and hence making the ROC value significant in our prediction. After looking at the classification report of the models, we have proceeded and concluded that **Random Forest Classifier** turns out to be the best of all models with good performance in all performance evaluators, whereas **XGBoost** turns out to be best considering Recall & AUC, both concluded after Hyperparameter tuning & Sequential Feature Selection.

VII. CONTRIBUTIONS

***Vedasamhitha Challapalli*:** EDA**,** Logistic Regression, Decision Tree Classifier, Naive Bayes , contributed to the team work, report, video
***R.Amshu Naik*:** Preprocessing**,** Random Forest, XGB, Random Search, SFFS,contributed to the team work , report, video,
***Perumalla Aasrish Vinay*:** EDA**,** SVM, KNN,SFFS,Grid search,contributed to the team work , report, video
All the members actively participated in the project discussion and coordinated well with each other.

IX. REFERENCES

[1] https://www.analyticsvidhya.com/blog/2021/04/mastering-exploratory-data-analysiseda-for-data-science-enthusiasts/
[2] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
[3] https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761
[4] https://www.geeksforgeeks.org/decision-tree-implementation-python/
[5] https://www.analyticsvidhya.com/blog/2021/05/exploratory-data-analysis-eda-a-step-by-step-guide/
[6]https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html
[7]https://vitalflux.com/roc-curve-auc-python-false-positive-true-positive-rate/

Thank you!