Final Report for the DS 675 - 005 Project

PREDICTING THE BRAIN STROKE USING MLP

(Krupa Lakhani-kgl, Deep Patel-djp223, Joy Patel-jp2267)

Keywords: Data Preprocessing, Exploratory Data Analysis, Tackling Imbalance, Logistic Regression, Decision Tree, Random Forest, Multi Layer Perceptron, XGBoost, K-Nearest Neighbors Algorithm.

Abstract: This project investigates the development of a predictive model for brain stroke occurrences through the analysis of relevant medical data. Utilizing machine learning algorithms and statistical techniques, the study aims to identify key risk factors and patterns associated with stroke incidence. The dataset comprises a diverse range of patient information, including demographic details, medical history, and lifestyle factors. The model's performance is assessed through rigorous validation techniques, and the results are discussed in terms of accuracy. Ultimately, the findings contribute to the advancement of early detection and prevention strategies for individuals at risk of suffering a brain stroke.

Dataset: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

Code: https://colab.research.google.com/drive/13-zoKOa9eoUheWj0Sgpdh3COjs88JH25?authuser=2#scrollTo=gVztB4WYvpX-

1. INTRODUCTION AND PROJECT STRUCTURE

We divided the tasks for this project into the following parts. This allowed us to tackle tasks in an efficient way and excel at each part individually.

I. Exploratory Data Analysis: Before we started applying any algorithms, we went over the dataset in depth. This step was particularly important because the chosen dataset is extremely imbalanced and large.

II. "Simple" Classifiers + Performance Assessment: With the information found in the EDA, we chose to apply the Decision Tree, Random Forest, KNN, XGBoost, and Logistic Regression on Categorical Features.

III. MLP Classifiers + Performance Assessment: We then trained a MLP Classifier to perform Classification.

IV. Comparison of multiple classifiers + Performance Assessment: We tried to compare the models we created in the previous two steps and performed hyperparameter tuning to improve the performance further.

V. Conclusion: We summarized the findings and decided to create all different models in our project along with the improved results. Furthermore, we went over limitations and discussed future possibility.

2. RELATED WORKS AND REFINEMENT

Before starting our project, we went through various literature reviews and we referred Ms. Saumya Gupta's and Supriya Raheja's approach. There were multiple approaches on stroke prediction using the chosen dataset. All of them had a similar structure but we started with some brief Exploratory data analysis, Data preprocessing, Tackling imbalances by treating null value feature and categorical variables and then built basic classifiers that predicted the Brain Stroke.

In Milestone 2, we particularly focused on the analysis conducted in notebook [2]. The author performed basic exploratory data analysis, focusing mainly on removing outliers. She then built three models, Logistic Regression, Random Forest and KNN, and compared their accuracy scores. The Random Forest algorithm was found to be the most accurate.

Our work is different from previous studies because of the following reasons:

First, we conducted **in depth exploratory data analysis**, When examining the dataset for missing values, it was observed that the "bmi" column had 201 missing values. In order to address this issue, the fillna() method from the **Pandas library** was employed. Second, we could also use **imputation method** but **fillna() method** is sufficient and effective for our dataset.[3] All the preceding work was based only on a single approach, basic classification models such as GNB, DT, and XGBoost.[1] We took a different approach by using **Multlayer Perceptron**. It is true that we took inspiration from previous implementations, but we're the only ones who came up with this model. We worked on defining our own bagging and boosting methods**,** based on the two best models we obtained during parts II and III of this study. Examining both basic and MLP classifiers independently turned out to be a prudent choice for us. This approach enabled us to incorporate resilience and ingenuity into the development of our ultimate model.

3. EXPLORATORY DATA ANALYSIS

We looked into all the variables in the dataset:

-Stroke: Stroke is the target variable of our dataset. Looking at its distribution, we can immediately tell that the dataset is **extremely imbalanced**: The dataset used in this study had 249 stroke cases and 4861 non-stroke cases, resulting in a highly imbalanced dataset with only 4.8% representing the minority class (stroke). To address this imbalance and improve performance on the minority class, Synthetic Minority Oversampling Technique **(SMOTE)** was applied.[4] SMOTE selects a random instance from the minority class and identifies its k nearest neighbors. This doesn't add new information to the model, but it improves performance.

- **Gender**: In the dataset for stroke prediction, we observe a notable gender distribution, with 41% male and 59% female. However, upon closer examination, it is evident that the occurrence of strokes is almost proportional to the gender distribution. **Therefore, this variable will be useful.**

- **Marital Status**: The dataset presents a predominant marital status distribution, with 66% married individuals and 34% single individuals.A comprehensive analysis reveals that marital status alone may not exhibit a substantial correlation with stroke incidence.No missing values are detected in the marital status feature. Therefore, this variable will be useful.

- **Smoking Status:** The feature analysis for smoking status reveals a distribution among the categories: "never smoked" with 1892 occurrences, "Unknown" with 1544 occurrences, "formerly smoked" with 885 occurrences, and "smokes" with 789 occurrences. In terms of percentage, "never smoked" constitutes approximately 38.55%, "Unknown" makes up about 31.41%, "formerly smoked" accounts for around 18.08%, and "smokes" represents roughly 16.96% of the total instances in the dataset. The analysis provides insight into the prevalence of different smoking statuses in the given data.

- **Work Type:** The dataset displays a varied distribution of work types, with 57% in private jobs, 16% self-employed, and 27% categorized as 'Other' (1366 instances).Confirming there are no missing values in the work type feature simplifies the analysis.While the dataset predominantly consists of individuals in private jobs, additional exploration is essential to determine the potential impact of work type on stroke incidence. **Therefore, this variable will be useful.**

- **BMI:** The feature analysis for BMI reveals that there are 110 outliers in the dataset, while the majority, 5000 instances, are non-outliers. Specifically, among the 201 instances with NULL values for BMI, 40 individuals experienced a stroke, translating to approximately 16.06% of the overall dataset. Due to the presence of stroke cases in the NULL values, imputing the missing BMI values with the median is chosen over dropping these instances.

- **Age:** The feature analysis for gender reveals three categories: Female with 2994 occurrences, Male with 2115 occurrences, and a single instance labeled as "Other." Given that "Other" has only one occurrence, it is deemed as an outlier and is removed to streamline the feature. After removal, the dataset now consists of Female and Male categories. The distribution shows that females constitute approximately 58.64% of the dataset, while males make up about 41.35%. Removing the singular "Other" instance helps maintain a more balanced and informative gender representation in the dataset.**This variable is very useful for all the classifiers to predict the stroke.**

- **Residence Type**: The dataset showcases a fairly balanced distribution of residence types, with 51% identified as urban and 49% as rural. Initial examination suggests no substantial correlation between residence type and stroke incidence. The relatively equal representation of urban and rural individuals in the dataset warrants further investigation to determine the impact, if any, of residence type on stroke prediction.

- **Hypertension:** The feature analysis for gender indicates an imbalance in the dataset, with 4612 instances labeled as "People not having" and 498 instances labeled as "People having." This imbalance raises concerns about the feature's significance in stroke prediction, as the minority class is underrepresented. Given the substantial class imbalance, neglecting this feature is a viable approach, as it is unlikely to significantly impact the accuracy of stroke predictions based on this feature alone. Consequently, excluding gender from the analysis may contribute to model simplification without compromising prediction accuracy for strokes. This decision is based on the understanding that other features in the dataset may carry more weight in predicting stroke occurrences.

- **Heart Disease**: The dataset indicates a substantial imbalance in the distribution of heart disease status, with approximately 94.5% of the population being free from heart disease and only 6.5% having heart disease. Initial analysis suggests a potential correlation between heart disease and stroke incidence, considering the notable difference in prevalence. The significant imbalance in heart disease distribution prompts further exploration to understand its impact on stroke prediction.

4. CLASSIFICATION MODELS AND ASSESSMENT

The procedures we implemented in this phase were shaped by the insights gained from our exploratory data analysis. During the EDA, we made determinations regarding the variables we deemed most suitable for inclusion or exclusion in our models. In this section, we validated our assumptions and utilized our discoveries to determine the features incorporated into our models.

We first focused on the categorical data. There are two types of categorical data: binary and multiclass. We decided to start by studying the prediction accuracy obtained using only binary variables and built some models: Logistic Regression, Random Forest, XGBoost, Decision Tree and KNN.

The dataset used in this study had 249 stroke cases and 4861 non-stroke cases, resulting in a highly imbalanced dataset with only 4.8% representing the minority class (stroke). To address this imbalance and improve performance on the minority class, Synthetic Minority Oversampling Technique **(SMOTE)** was applied.

SMOTE selects a random instance from the minority class and identifies its k nearest neighbors. We performed SMOTE[4.1] in order to balance it. Thanks to SMOTE, we are now more likely to obtain better accuracy in our models.

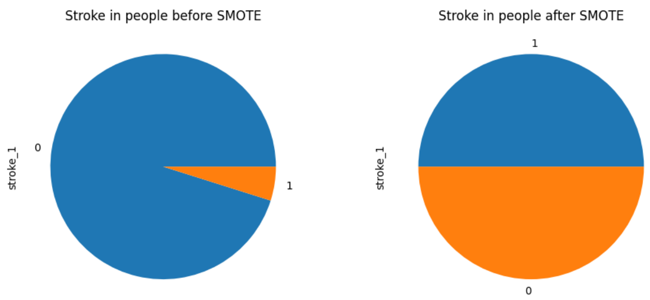
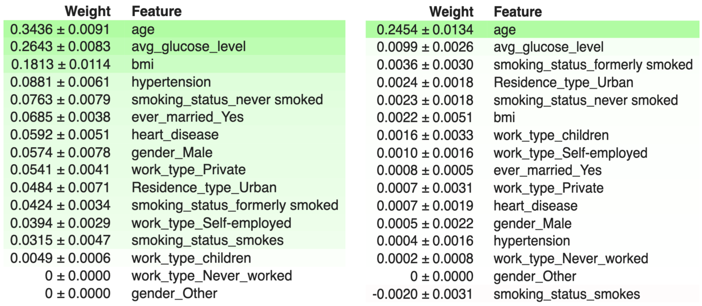


Fig 4.1
L: Data Imbalance before using SMOTE, R: Data Balance after using SMOTE

We performed permutation importance to compare the feature importance for our best and worst models.[5]

The **Logistic Regression model:** age is important feature of this model. Looking at the feature selection. Bmi and Hypertension are least important in this model unlike other models where these features are equally important. Thus this model performs worst predictions.

The **RandomForest classifier:** age is important feature of this model. Looking at the feature selection. work_type is least important in this model. But it considers Bmi and hypertension are important unlike LR model. Thus this model performs best predictions. Below are the comparison of the confusion matrices of these three models in classifying the data.



Weight	Feature
0.3436 ± 0.0091	age
0.2643 ± 0.0083	avg_glucose_level
0.1813 ± 0.0114	bmi
0.0881 ± 0.0061	hypertension
0.0763 ± 0.0079	smoking_status_never smoked
0.0685 ± 0.0038	ever_married_Yes
0.0592 ± 0.0051	heart_disease
0.0574 ± 0.0078	gender_Male
0.0541 ± 0.0041	work_type_Private
0.0484 ± 0.0071	Residence_type_Urban
0.0424 ± 0.0034	smoking_status_formerly smoked
0.0394 ± 0.0029	work_type_Self-employed
0.0315 ± 0.0047	smoking_status_smokes
0.0049 ± 0.0006	work_type_children
0 ± 0.0000	work_type_Never_worked
0 ± 0.0000	gender_Other

Weight	Feature
0.2454 ± 0.0134	age
0.0099 ± 0.0026	avg_glucose_level
0.0036 ± 0.0030	smoking_status_formerly smoked
0.0024 ± 0.0018	Residence_type_Urban
0.0023 ± 0.0018	smoking_status_never smoked
0.0022 ± 0.0051	bmi
0.0016 ± 0.0033	work_type_children
0.0010 ± 0.0016	work_type_Self-employed
0.0008 ± 0.0005	ever_married_Yes
0.0007 ± 0.0031	work_type_Private
0.0007 ± 0.0019	heart_disease
0.0005 ± 0.0022	gender_Male
0.0004 ± 0.0016	hypertension
0.0002 ± 0.0008	work_type_Never_worked
0 ± 0.0000	gender_Other
-0.0020 ± 0.0031	smoking_status_smokes

Fig 4.2 & 4.3
L to R: Random Forest, Logistic Regression

It's also worth mentioning the accuracy score of each model. Logistic Regression gives around 78.32%, and RandomForest around 99.83%. From this we conclude that Random Forest is the best classifier.

During the Exploratory Data Analysis (EDA), we identified a binary categorical feature, namely "Hypertension," that appeared to be less likely to offer relevant information. Consequently, we evaluated the performance of our models both with and without the inclusion of the "Hypertension" feature.

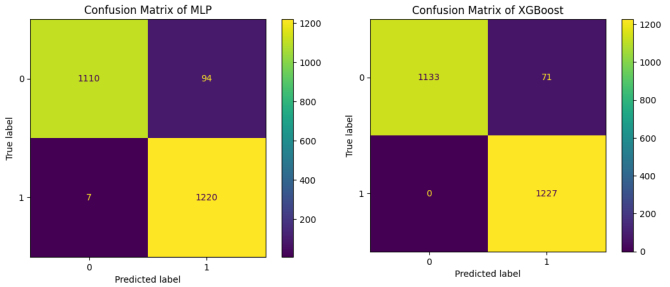


Fig 4.4
Using Hypertensipon Feature
L to R: Multilayer Perceptron, XGBoost

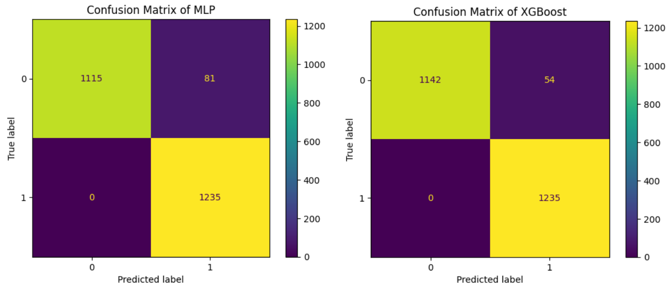


Fig 4.5
Without Using Hypertension Feature
L to R: Multilayer Perceptron, XGBoost

Figures 4.4 and 4.5 illustrate that including or excluding the "Hypertension" variable had negligible impact on the accuracy, confirming our initial assumption of its limited relevance. However, the accuracy results obtained using solely binary categorical data were suboptimal. Upon evaluating the inclusion of all categorical variables, including "age," "bmi," and "Hypertension," we discovered that the best classification performance was achieved when utilizing all features. This underscores the importance of verifying assumptions made during the Exploratory Data Analysis (EDA).

5. Deep Neural Network based Model(MLP)

A Multilayer Perceptron (MLP) with five layers refers to an artificial neural network architecture composed of an input layer, three hidden layers, and an output layer. In this configuration, information flows forward through the network, and each layer contains interconnected nodes or neurons. A dropout rate of 10% after each hidden layer, except the last one, is implemented during training to prevent overfitting. Dropout involves randomly deactivating a fraction of neurons, enhancing the model's generalization capabilities.[6]

For brain stroke analysis, the MLP is trained on a dataset comprising health-related features. The network learns to map these features to the binary outcome of stroke occurrence, utilizing backpropagation and gradient descent for weight adjustments. The achieved accuracy of 96.30% indicates the model's proficiency in correctly classifying stroke and non-stroke instances.

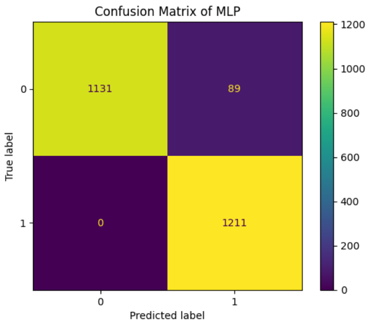


Fig 5.1 Confusion Matrix of MLP

It returned no false negatives and only 94 false positives. Precision, measuring the accuracy of positive predictions, is at 92.33%, suggesting that when the model predicts a stroke, it is correct 92.33% of the time. Specificity, indicating the accuracy of negative predictions, is at 92.00%, showcasing the model's ability to correctly identify non-stroke instances. The recall, measuring the model's capacity to correctly identify all relevant instances, is 100%, indicating that the model successfully identifies all actual stroke cases.

The success of this MLP architecture can be attributed to its ability to capture complex relationships within the data, facilitated by the multiple hidden layers and dropout regularization. Proper tuning of hyperparameters, including dropout rates, contributes to the model's robust performance. The high recall suggests the model's effectiveness in identifying stroke cases, a critical aspect in healthcare applications.

In summary, the presented MLP architecture with dropout regularization demonstrates strong performance in brain stroke analysis, achieving high accuracy, precision, specificity, and recall. This indicates its potential as a reliable tool for predicting and identifying stroke occurrences based on relevant health features.

6. TRAINING THE OTHER CLASSIFICATION MODELS

We utilized several well-known classifiers in our study, including the Decision Tree Classifier, KNearest Neighbours, XGBoost Classifier, and Random Forest Classifier. These classifiers have a solid reputation, allowing us to compare our findings with established results from previous research.

To train our algorithm, we allocated 75% of our dataset, reserving the remaining 25% for evaluating the trained model's performance. We employed 5-fold cross-validation for machine learning model validation.[7] In this technique, the entire dataset is used for both training and testing the classification process, divided into 5 parts or folds. During training, k-1 folds are used to train the ML model, and one fold is designated for testing. This process is repeated 5 times, with each fold serving as a test dataset. One of the advantages of this approach is that it utilizes all samples in the dataset for both training and testing, minimizing high variance.

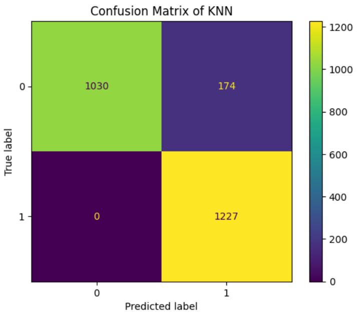
To assess the model's performance, we employed confusion matrices, which help calculate metrics such as accuracy, Area Under the Curve, precision, and Receiver Operator Curve. Through the analysis of these values, we identified the best-performing model for predicting strokes.

a. K-Nearest Neighbours Algorithm

KNN examines the classes of a specified number of training data samples surrounding a test data point to predict the class to which the test data point belongs. The parameter "k" represents the count of the nearest neighboring data samples.

The KNN algorithm categorizes new, unlabeled data by identifying the classes of its nearest neighbors, and this fundamental concept is integral to the algorithm's computations. When presented with a new instance, KNN performs two operations: firstly, it identifies the K points closest to the new data point, and secondly, based on the classes of these neighbors, KNN determines the appropriate class for the new data.[8]

The calculation involves computing the Euclidean distance between the test sample and the specified training samples. It returned no false negatives and only 174 false positives.



$$\sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Fig 6.1 KNN Confusion Matrix & Equation 6.1

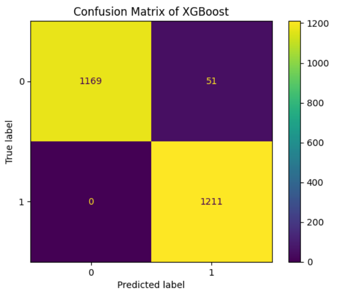
b. XGBoost Algorithm

XGBoost is a supervised learning algorithm that relies on ensemble trees. Its objective is to optimize a cost function, which consists of a loss function (denoted as "d") and a regularization term (represented by an empty set symbol).[9]

$$\Omega(\theta) = \underbrace{\sum_{i=1}^{n} d(y_i, \widehat{y_i})}_{Loss} + \underbrace{\sum_{k=1}^{K} \beta(f_k)}_{regularization},$$

Equation 6.2

In this context, where yiˆ represents the predicted value, n denotes the count of instances within the training set, K signifies the quantity of trees to be created, and fk is an individual tree from the collection of trees, the regularization term is precisely defined as below equation. It returned no false negatives and only 42 false positives.



$$\beta(f_t) = \gamma T + \frac{1}{2}\left[\alpha \sum_{j=1}^{T} |c_j| + \lambda \sum_{j=1}^{T} c_j^2\right],$$

Fig 6.2 & Equation 6.3

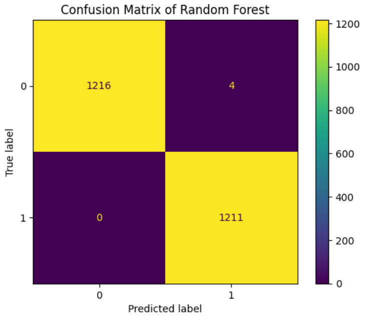
c. Random Forest Algorithm

A Random Forest is composed of a collection of decision trees, where each tree relies on a randomly sampled vector value. These vectors are chosen independently and follow the same distribution across all the trees in the forest. Each tree contributes a voting unit, supporting the most prevalent class in the input.
In the context of Random Forest convergence, improving accuracy is achieved by centralizing the random forests through the determination of a margin function. This margin function plays a crucial role in obtaining more precise results. If we denote the classification ensemble as h1(x), h2(x), ..., hk(x) with a training set randomly selected from the distribution of the random vector Y, X, the margin can be ascertained using the following equation:[10]

$$mg(X,Y) = av_k I\left(h_k(X) = Y\right) \\ - \max av_k I\left(h_k(X) = j\right).$$

Equation 6.4

The indicator function is denoted as I(). The margin function serves to quantify the extent to which the average number of votes in Y, X for a particular class surpasses the average votes for other classes. A greater margin signifies more precise classification results.
Regarding strength and correlation, the maximum limit on the random forest can be determined for generalization error through below equation. It returned no false negatives and only 4 false positives. Since we successfully identified almost all the Stroke, we considered this our best model.



$$PE^* \leq \bar{P}\left(1 - s^2\right)/s^2$$

Fig 6.3 Confusion Matrix of RF & Equation 6.5

d. Decision Tree Algorithm

Broadly speaking, the CART algorithm follows a series of steps to construct a decision tree:

1. Select an attribute to serve as the root.
2. Partition the cases into branches.
3. Iterate through the process on each branch until all cases within the branch share the same class.

In the process of choosing an attribute for the root, the decision is based on the attribute with the highest gain value among the available attributes. The calculation of entropy, which measures impurity, can be determined using the formula presented in equation below.[11]

$$Entropy(s) = \sum\nolimits_{i=1}^{n} P_i * \log_2 Pi$$

Equation 6.6

$$Gain(S,A) = Entropy\ (S) - \sum\nolimits_{i=1}^{n} \frac{|S_i|}{|s|} * \ Entropy\ (S_i)$$

Equation 6.7

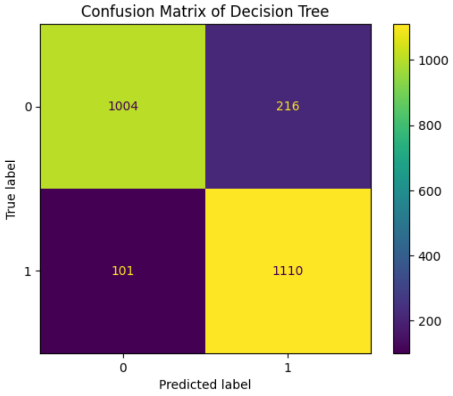


Fig 6.4 DT Confusion Matrix

Decision Tree returned 101 false negatives and only 216 false positives.

e. Logistic Regression Algorithm

Logistic regression is a statistical method commonly used in brain stroke prediction analysis. It is a binary classification algorithm designed to predict the likelihood of an individual experiencing a stroke based on given input features. Logistic regression models the relationship between these features and the binary outcome of whether a person is at risk of a stroke or not.

By estimating the probability of stroke occurrence, logistic regression aids in identifying key risk factors and informing preventive measures for individuals at higher risk. This Classifier gives the worst accuracy among all the classifiers. It returned 220 false negatives and 320 false positives.[12]

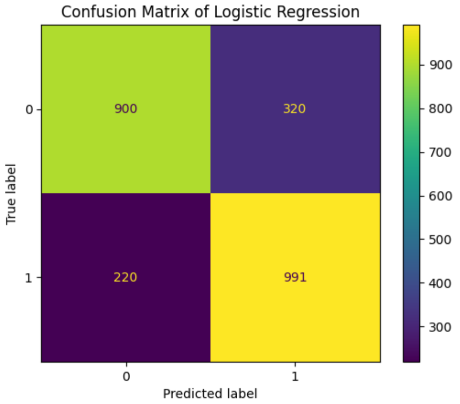


Fig 6.5 LR Confusion Matrix of LR

7. HYPER PARAMETER TUNING

We then tried to see if we could improve the performance even further using **hyperparameter tuning.**

- **Tuning our Model MLPClassifier with Adam Optimizer:** Number of hidden layers are three, and neurons(units). We have used **Relu** as an **activation function** for all the layers accept the last layer which has **Sigmoid**. **Loss function** is binary cross entropy. Batch size is 256 and 100 epochs. **Drop out regularization** of 10% after each layer. We used a **early stopping** method for more accurate predictions. For the most accurate prediction the model stops training at 52 out of 100 epochs.
- **Tuning our Model XGBoostClassifier:** We used **GridSearchCV** to iterate over all the hyperparameters and find the best fit from these. The result gives maxdepth of 7, and estimators equal to 200.
- **Tuning our Model KNN Classifier (used GridSearchCV):** Algorithm used is Ball_Tree, and number of Neighbors is equal to 9, P = 2(Euclidean), and weights are considered distances.
- **Tuning our Model DT Classifier (used GridSearchCV):** Max_depth = 10, max_feature = sqrt, min_sample_leaf = 2, min_sample_split = 5
- **Tuning our Model RF Classifier (used GridSearchCV):** Max_depth = 30, max_ feature = srqt , min_sample_leaf = 1, min_sample_split = 2, number of estimators = 2.

By performing the hyperparameter tuning the accuracy of Random Forest, XGBoost and Multilayer Perceptron is increased.

8. RESULTS

Classification	Accuracy	Precision	Recall	Specificity
Logistics Regression	77.78	75.59	0.81	73.77
Decision Tree	86.96	83.71	0.91	82.29
K-Nearest Neighbors	90.62	84.15	1	81.31
Multilayer Perceptron	96.33	93.15	1	92.7
XGBoost	97.90	95.95	1	95.81
Random Forest	99.83	98.0	1	98.16

We found that the **best model was Random Forest Classifier with 99.83% of accuracy and recall of 100%**. It returned no false negatives and only 4 false positives.

9. CONCLUSION

During this project, we had some limitations derived from the dataset we were working with. Our data was imbalanced and it had a 201 missing fields. This makes sense since there is no standard template that all features follow to predict stroke, which makes it harder to compare them. However, we were able to overcome them using techniques like SMOTE method.

Thanks to starting with more basic algorithms and progressively using more advanced techniques we were able to achieve 99.83% classification accuracy using Random Forest Classifier. Furthermore, the only errors made by our model were False Positives, which means it was able to predict all stroke.

The selection of the Random Forest model for future predictions was based on its exceptional accuracy and high performance across various metrics. Achieving the highest accuracy at 99.13%, Random Forest demonstrates proficiency in classifying data, particularly with incomplete attributes and large sample sizes. It excels in obtaining a perfect ROC and AUC score of 0.99, along with a Precision of 0.98. However, the model encounters challenges when handling invalid (semantically) values for attributes like age, BMI, and average glucose levels. While increasing the number of trees can enhance performance, it may lead to slower real-time usage.

10. REFERENCES

[1] Reference Paper
https://ieeexplore.ieee.org/document/9734197

[2] Kaggle Notebook Reference, used in Milestone 2.
https://www.kaggle.com/code/alexisbcook/categorical-variables

[3] Kaggle Notebook Reference, Data Cleaning
https://www.kaggle.com/code/dansbecker/handling-missing-values

[4]Applying SMOTE.
https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

[5]Permutation Importance
https://www.kaggle.com/code/dansbecker/permutation-importance

[6] MLP.
https://learnopencv.com/implementing-mlp-tensorflow-keras/

[7] Cross Validation
https://www.kaggle.com/code/alexisbcook/cross-validation

[8]KNN
https://www.kaggle.com/code/amolbhivarkar/knn-for-classification-using-scikit-learn

[9]XGBoost classifier
https://www.kaggle.com/code/alexisbcook/xgboost

[10]Random Forest Classifier
https://www.kaggle.com/code/dansbecker/random-forests

[11]Decision Tree
https://www.kaggle.com/code/stieranka/decision-trees

[12]Logistic Regression
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html