CDS503: Machine Learning
Academic Session: Semester 1, 2022-2023
School of Computer Sciences, USM, Penang

Machine Learning to Aid in the Forecasting of Brain Strokes

Project Group 03
Austin Smith - P-COM0084/22
Ahmed Adel Sanhan Al-Haidary - P-COM0113/22
Samira Elsamad- P-COM0147/22

Table of Contents

Abstract

Strokes have become one of the top causes of death in recent years, impacting the central nervous system. Ischemic and hemorrhagic strokes cause the most significant damage to the central nervous system. Globally, 3% of people suffer from subarachnoid hemorrhage, 10% from an intracerebral hemorrhage, and 87% from ischemic stroke, according to the World Health Organization (WHO). In this study, machine learning techniques are used to identify, categorize, and predict brain stroke using medical data.

1.0 Project Background

1.1 Background of the Problem Domain

Stroke is now a significant health issue (Merino, 2014). Known alternatively as a cerebrovascular accident, a stroke is a brain condition that can happen as a result of ischemia (due to the lack of blood flow) or bleeding of the brain's arteries typically results in a variety of functionally detrimental motor and cognitive deficits (*Stroke - What Is a Stroke?*, 2022). Stroke signs and symptoms can include difficulty speaking or understanding, dizziness, or a loss of vision on one side of the body. They can also include difficulty moving or feeling on one side of the body. When a stroke occurs, signs and symptoms frequently show up quickly. Every year, stroke affects around 16 million people worldwide and has significant economic repercussions. Recently, ML (Machine Learning) has rapidly expanded and changed in several applications across numerous healthcare systems. Because stroke has a high fatality rate, it is regarded as a serious health problem by the American Heart Association (Benjamin et al., 2018).
Additionally, there is an increased need for improved technology that can aid in clinical diagnosis, treatment, predictions of clinical events, recommendations of promising therapeutic approaches, rehabilitation programs (Almeida et al., 2020), etc. due to the rising costs of stroke hospitalization (Di Carlo, 2009). For a stroke to be effectively treated, early detection is essential, and ML can be very helpful in this process. Machine Learning (ML) is a cutting-edge technology that enables health practitioners to make clinical judgments and predictions to achieve that.
The organization of this paper is as follows: chapter 2 literature review. The methodology is described in chapter 3. Section 4 delineates the experimentation and results from the analysis. Finally, Chapter 5 concludes the paper with a discussion of the best algorithm for predicting brain strikes.

1.2 Issues and Problem Statement

Brain stroke claims the lives of a significant number of individuals each year, and the rate is rising in emerging nations (R, 2001). The various forms of stroke are controlled by several risk factors.

Understanding the connection between these risk factors and various types of strokes is made easier by predictive algorithms. Through early identification and treatment, the machine learning algorithm can enhance the health of patients. From a patient's clinical report and statistical data, we have employed several machine learning algorithms to identify the likelihood of brain stroke that may occur in the patient or have already happened. We have collected the dataset from the Kaggle website. After that, the dataset was prepared for the machine learning algorithms.

1.3 Project objectives and motivation

According to many neurologists, there is currently no treatment that can totally heal a stroke. Instead, we offer supportive palliative care, which would undoubtedly increase a person's longevity. In developing nations, there were ten times as many people who lost their lives to strokes, and by 2030, this number is expected to double globally. A study that was conducted in two phases by the Canadian Institutes of Health Research and the Heart and Stroke Foundation showed the influence of a patient's risk factor on the likelihood of stroke (WHO, 2004). A higher percentage of patients were discovered to be impacted by ischemic stroke, according to the report provided by the Asian Stroke Advisory Panel (Suwanwela et al., 2016). We need to forecast the possibility of a brain stroke as soon as feasible in order to lessen the severity of symptoms in such diseases. The primary objective of this study is to acquire a stroke dataset and classify the chances of stroke by using machine learning algorithms. The brain stroke data is tagged and classified in order to accomplish the main goal.

2.0 Literature Review

There has been plenty of research conducted recently into employing various machine learning methods to predict strokes:

Sudha et al. (Sudha et al.,2012) employed Decision Tree, Bayesian Classifier, and Neural Network classification algorithms to predict strokes and give some related features. The dataset contained 1000 observations. Principal component analysis (PCA) algorithm was used to reduce dimensions and specify the most valuable attributes. Neural Network, Naive Bayes classifier, and Decision Tree algorithm achieved 92%, 91%, and 94% accuracy respectively.

Amini et al.(Amini et al., 2013) proposed a stroke classification model based on k-nearest neighbor and C4.5 decision tree algorithm. Data on 50 stroke risk factors including diabetes and smoke consumption was collected from 807 healthy and unhealthy patients. It was concluded that both k-NN and C4.5 decision tree algorithm can successfully predict strokes as k-NN had 94% accuracy and C4.5 decision tree algorithm had 95%.

Adam et al. (Adam et al.,2016) study developed an ischemic stroke classification model utilizing both the decision tree algorithm and k nearest neighbor(k-NN) [7]. The data used to build the model was collected from various hospitals in Sudan. It consisted of 15 features and 400 observations. In this study, the decision tree algorithm performed better than k-NN. Their results were determined to aid medical professionals in classifying ischemic strokes.

R. Jeena et al.(R. Jeena et al.,2016) used Support Vector Machine (SVM) for stroke classification [41]. The data was collected from the International Stroke trial Database and consisted of 12 features and 350 observations. They applied several kernel functions including polynomial and linear functions. The linear kernel was determined to give the highest accuracy of 91%.

Govindarajan et al. (Govindarajan et al.,2019) applied text mining tools along with machine learning algorithms to build a stroke classification prototype. The machine learning algorithms used were Artificial Neural Networks (ANN), Support Vector Machine (SVM), Decision Tree and Logistic Regression. Data containing observations from about 507 stroke patients was collected in an Indian hospital. Ischemic stroke affected 91.52% of the patients, while hemorrhagic stroke affected the other 8.48%. An artificial Neural Networks with stochastic gradient descent learning algorithm performed the best out of the mentioned machine learning algorithms with an accuracy of 95.3% in stroke classification.

Kansadub et al. (Kansadub et al., 2015) conducted a study on stroke prediction using Decision Tree, Naïve Bayes and Neural Network. The algorithms were evaluated on their accuracy and area under ROC curve (AUC). The Decision Tree algorithm was the most accurate and Naïve Bayes was best in AUC.

Singh et al. (Singh et al.,2017) conducted a study utilizing artificial intelligence. Data was selected from the cardiovascular health study (CHS) dataset. The decision tree algorithm was used to extract features and a neural network classification algorithm was used for model building. The model had 97% accuracy.

Chin et al. (Chin et al.,2017) was performed to provide an automated early ischemic stroke detection system. The system was developed using a convolutional neural network (CNN) deep learning algorithm. The CNN model was trained and tested using 256 patch images. The patch images are prepossessed CT images of the brain with data prolongation. This CNN experiment has accuracy higher than 90%.

3.0 Methodology

For this project, group 3 will be using the CRoss Industry Standard Process for Data Mining (CRISP-DM) process. This process is very common in the machine learning community and will aid us in proper work flow and model preparation. The first step in the CRISP-DM process is to understand the background of the project, determine the requirements of the project, establish the objectives of the project, and design a plan for the project which can meet those objectives. The second step of CRISP-DM is to collect, understand, explore, and ascertain the quality of the data. This second step is shown below in section 3.1. The description for the data will include where we obtained the data and the source. The third step in the CRISP-DM process is the data preparation which can be read in section 3.2. In section 3.2 of the proposal, we discuss how we plan to clean the data by removing outliers, replacing empty data, and filtering out data to bring the dataset more into balance. Not included in this proposal are the fourth and fifth steps which involve modeling the algorithm, evaluating the algorithm, and tuning the algorithms. These steps will be discussed in sections 3.3, 3.4, and 4.1 respectively and completed for the final presentation of the findings. The final phase of CRISP-DM of deployment will be executed via the oral presentation given later this semester.

3.1 Project Framework

The group aims to implement 2 machine learning algorithms that model and predict a heart stroke with greater than 50% chance of success. If successful, this knowledge could assist those in the global community with stroke prevention via understanding which factors correlate with strokes and which subsets of the population should be more aware of potential strokes.

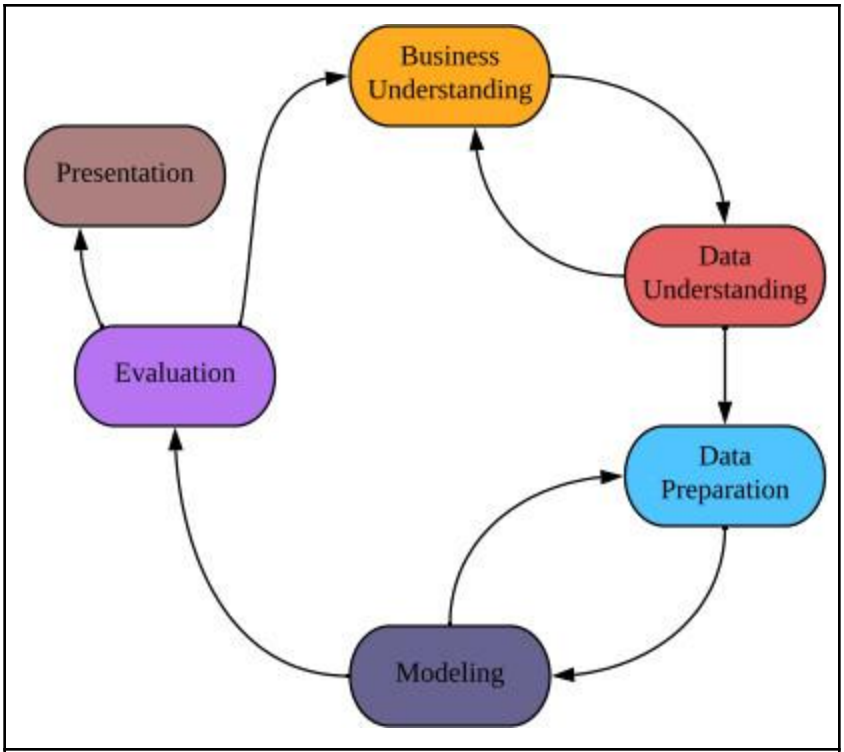


Figure 3.1 - CRISP-DM Methodology

3.2 Dataset and Preparation

For our project, we found a stroke prediction dataset on kaggle at the following link: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset . The dataset author did not disclose when, where, or how the dataset was collected.The dataset contributor Fedesoriano claimed that " because of the sensitive content the source of the data is going to remain confidential".
In the original dataset, there were twelve different attributes with 1554 instances. Those twelve attributes are as follows:

1. Unique patient identifier: random integer
2. Gender: "Male" [41%], "Female" [59%], "Other" [<0.02%]
3. Age: positive integer [min: 0.08, mean: 43, max: 82]
4. Hypertension: 1 (yes) [90%], 0 (no) [10%]
5. Heart disease: 1 (yes) [5%], 0 (no) [95%]
6. If they were ever married: "Yes" [66%], "No"[33%]
7. Type of work: "children" [13%], "Never_worked" [0.43%], "Private" [57%]. "Self-employed" [16%], "Govt_job" [13%]
8. Type of residence: "Urban" [51%], "Rural" [49%]
9. Average glucose level: continuous [min: 55, mean: 106, max: 272]
10. Body mass index (BMI): continuous [min: 10, mean: 29, max: 98]
11. Smoking status: "formerly smoked" [17%], "never smoked" [37%], "smokes" [15%], "unknown" [30%]
12. Stroke Experience: 1 (yes) [5%], 0 (no) [95%]

There will need to be pre-processing as there are missing values within the BMI which can be filled in with the median BMI from a person of that age group. There is one entry with the gender of “Other” and that can be dealt with via removing the entry. Since the smoking status of “unknown” is not helpful, we can remove those entries as the unknown status only applies to less than 20 percent of those who had a stroke and 32 percent of the non-stroke cases. Removing these cases of “unknown” smoking status will help bring the dataset more into balance and remove noise from the dataset. The data class distribution is not balanced as there are approximately 4800 people who have not had a stroke and only 200 people who have had a stroke. This data is imbalanced and can be adjusted via removing those entries with an “unknown” smoker status, removing entries whose age is less than a set threshold, and other filtering yet to be decided upon.

3.3 Machine Learning Algorithms

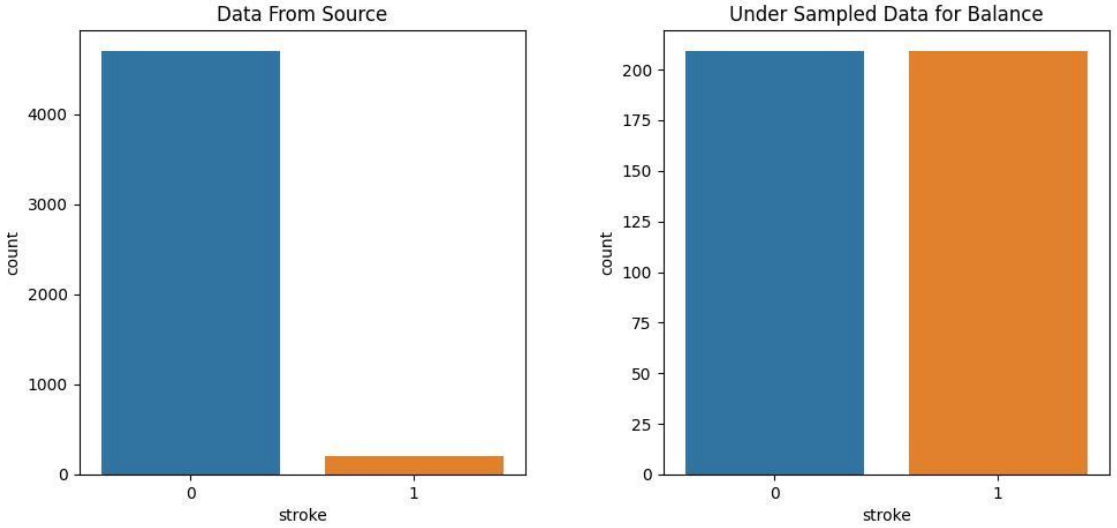
When considering which model to use, we compared parametric classifiers with themselves and non-parametric classifiers with themselves. Parametric classifiers normally need less training data and assume some form of the data. On the other hand, with non-parametric classifiers, the classifier algorithm attempts to discover the form of the data from the training data. We compared the following parametric classifiers: Naive Bayes classifier, Linear Support Vector Classification (SVC), and Logistic Regression parametric classifiers against each other. When comparing nonparametric classifiers, we compared the following: K-Nearest Neighbor, decision tree, and Support Vector Classifier using the Radial Basis Function (SVM-RBF) Kernel.

In order to select a parametric and non-parametric algorithm, testing of all of the algorithm’s accuracy was performed. For this comparison, we did not try to optimize the algorithms and parameter tune. Instead, we used a general approach for each algorithm, and after comparing the results, we will choose the best algorithms (one parametric and one nonparametric) and tune those best-performing algorithms so as to maximize their accuracy.

Second, when attempting to compare the classification algorithms, we reduced the number of variables being considered so as to give each classification algorithm its best performance. When analyzing the effectiveness of the classifying algorithms, we were able to train and then test each single variable within the database. The highest accuracy from each test was recorded so that we could compare.The assumption was that using the best singular variable test for each variable would be a good place to start for comparing classifying algorithms without parameter tuning.

Third, to calculate algorithm accuracy we used a cross-validation with 9 folds. Cross validation minimizes overfitting by breaking the data into 9 groups. Each of the 9 groups is then used as the test data while the other 8 groups are used as the training data. The model’s parameters from each of these 9 runs are then averaged. This averaging of the model parameters helps reduce overfitting of the data. Then after cross validation, the average accuracy of those 9 test sets was recorded. After all features were individually tested for each classifier, the best feature and its accuracy for stroke prediction was saved.

When comparing accuracy of the classification models, it is important to use a balanced dataset (equal stroke and no-stroke cases) so as to not overfit the model to one case or the other. Second, because the accuracy calculation is less sensitive to false negatives, the balanced dataset helps the measurement to be more true. Because this dataset was moderately imbalanced in favor of “no stroke” with a ratio near 22:1, we undersampled the dataset from the “no stroke” class so as to train our models with balanced data. For this initial algorithm comparison, all of the positive stroke cases were used along with the same amount of “no stroke” cases selected from the data at random.

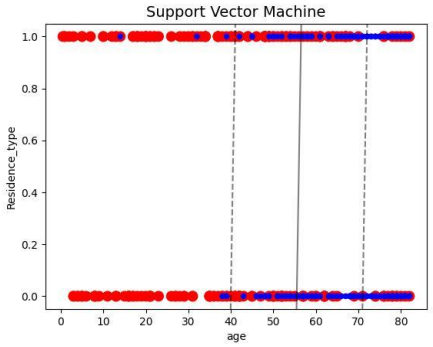


3.3.1 Analysis Of Parametric Classifiers

The first parametric classifier for analysis is the Naive Bayes Classifier. The classifier works by applying Bayes Theorem to our data. Bayes Theorem helps describe the probability of one event happening based on other possibly related conditions. For example, using our dataset, Bayes Theorem will attempt to describe the probability of having a stroke given a person’s age, bmi, smoking status, work type… etc. After the training data is provided, we can then use the test data to see if the probabilities are consistent across all samples. For the classifier, we used the Gaussian Naive Bayes classifier as we are using many continuous datasets. The best accuracy was 78% from Naive Bayes using only data from the “age” feature.

The second parametric classifier for analysis is the Linear Support Vector Classification. In general, this classifier works by plotting the “n” features into n-dimensional space (each n represents a different feature). For our simple algorithm comparison, we will be using 1 feature at a time, and thus we will be using a 1 dimensional space. Given n-dimensional space, a hyperplane is then passed through the data attempting to differentiate between “stroke” and “no stroke” classes. For this initial analysis, the “C” value was set to 1. Which (compared to a value of 100), enables the algorithm to look for a larger-margin separating hyperplane at the expense of misclassifying points. When using Linear SVC, it is not recommended to use large

values of C. Hard margins (large values of C) may not have a solution, and (compared to a softer margin) are less robust to outliers. Not only will this plane pass through the classes, but also will attempt to maximize margin between these two groups. After analysis, Linear Support Vector Classification scored an accuracy of 77.98% by using age as its singular best performing features. Below is shown an example of SVM in 2D space using “residence_type” and “age” as the two features.



The third parametric classifier for analysis was the Logistic Regression classifier. This classifier operates off of a “maximum likelihood” principle and is commonly used when two-class algorithms (such as our own stroke vs. no-stroke). The algorithm calculates the likelihood of something occurring and classifies the result accordingly. Specifically, this algorithm uses a sigmoid function to map real numbers between 0 and 1. If the outcome of the sigmoid is greater than 0.5, then it is labeled positive (stroke), and if the outcome of the sigmoid is less than 0.5, it is labeled negative (no-stroke). The training data is used to formulate the parameters of the sigmoid, and then the testing will yield the results of the training. Unlike a linear regression that uses an R2 factor to indicate goodness of fit, logistic regression uses maximum likelihood. When considering all of the data from the balanced dataset (not just one variable at a time), the logistic regression classifier generated the following coefficient table.

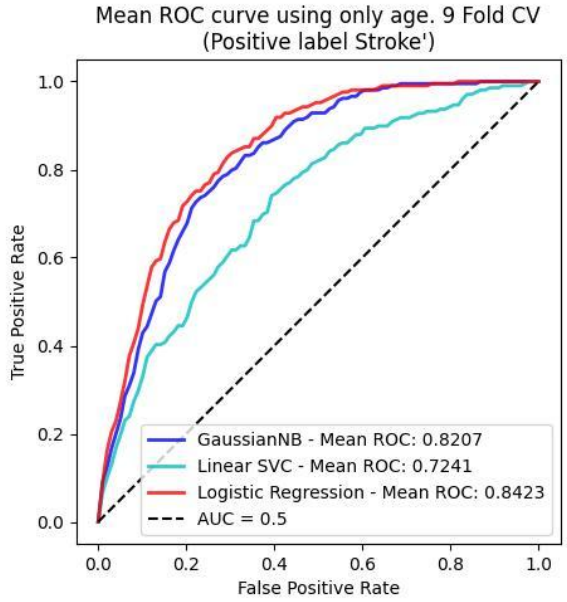
Variable	coef	P>|z|
age	0.0637	0
bmi	-0.0529	0
smoking_status	-0.3516	0.001
work_type	-0.5052	0.002
hypertension	0.8569	0.012
heart_disease	0.4321	0.277
gender	-0.2083	0.395
Residence_type	-0.176	0.461
avg_glucose_level	0.0006	0.798
ever_married	0.0219	0.953

The p-values in the table above indicate if the defined coefficients within the table are reliable for use in calculating the likelihood.When using the p-table above, it is common practice to only trust the estimated coefficient values from variables with p-values less than 0.05. Those variables would be as follows: age, hypertension, work_type, bmi, and smoking status. For logistic regression, the variables such as “ever_married” or their average glucose level, did not have a large influence on the maximum likelihood. For

our rough singular variable analysis comparison, we saw the best singular feature performance from “age”. Given age, the logistic regression was accurate 78% of the time.

Below are the accuracy results from the three parametric algorithms

Algorithm	Single Feature Accuracy	Best Single Feature for Accuracy
Naive Bayes (Gaussian)	78.47%	age
Linear SVM	77.981	age
Logistic Regression	77.992	age

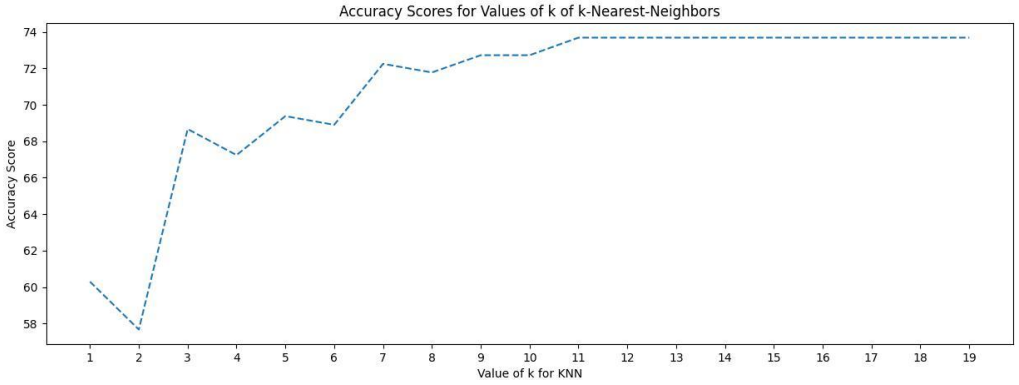


Using the balanced dataset, the above figure shows the Area Under the Curve (AUC) Receiving Operating Characteristics (ROC) for all three algorithms. The chart shows the true positive rate plotted against the false positive rate. As the chart shows, as the algorithms become more sensitive to true positives they become more sensitive to false positives as well. The more area under the curve (AUC) (area between the curve and the x-axis), the better the algorithm is said to have performed. Thus, the ROC curves that are closest to a true positive rate of “1.0” while maintaining a 0.0 false positive rate are best. For this study given the balanced dataset, the Logistic Regression algorithm performed the best.

As a result of classifier analysis, we have decided to go with the logistic regression classifier as it has the highest accuracy with limited variables. We expect with better data pre-processing and filter tuning, the accuracy will increase.

3.3.2 Analysis Of Non-Parametric Classifiers

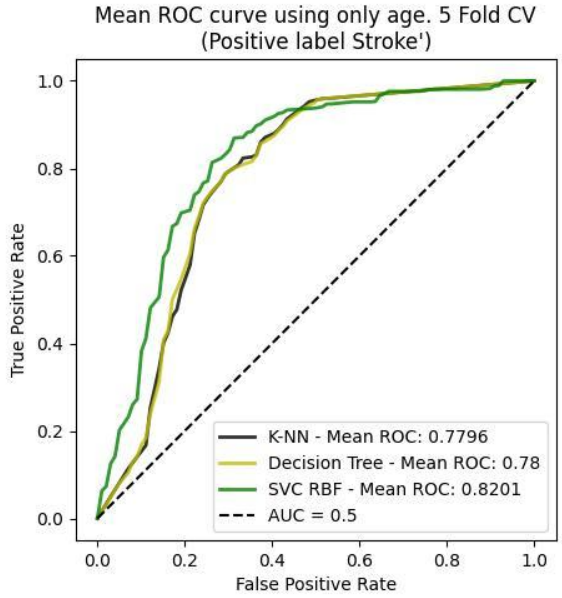
The first non-parametric classifier is the k-nearest neighbor (KNN) classifier. Being non-parametric means that it does not know or assume a form for the data. KNN does not assume a linear form (or any form for that matter), instead classification of the data is done through analyzing the known class of data points near the point of classification. KNN measures the distance between the point to be classified and all other data points. After that, a classification is assigned to the new data point according to the majority of its “k” closest neighbors. For the rough approximation of the algorithm performance, we did a quick search for the optimal k value and set it to 11. Using the decision tree classifier, the single best feature accuracy was found at 74% and age was the best performing feature.



The second non-parametric classifier is the Decision Tree classifier. The decision tree classifier uses the training data to design a set of rules that will classify the test data. Each rule generated during training will split the data in an attempt to properly classify the data point. For example, if the classifier generated the rule that all ages over 65 were positive for stroke, then if the new datapoint contained the age of 70 years old, then the data point would be classified as positive. Using each variable by itself for training and testing, the best single variable was “age”, and using age, the decision tree classifier had an accuracy of 74%.

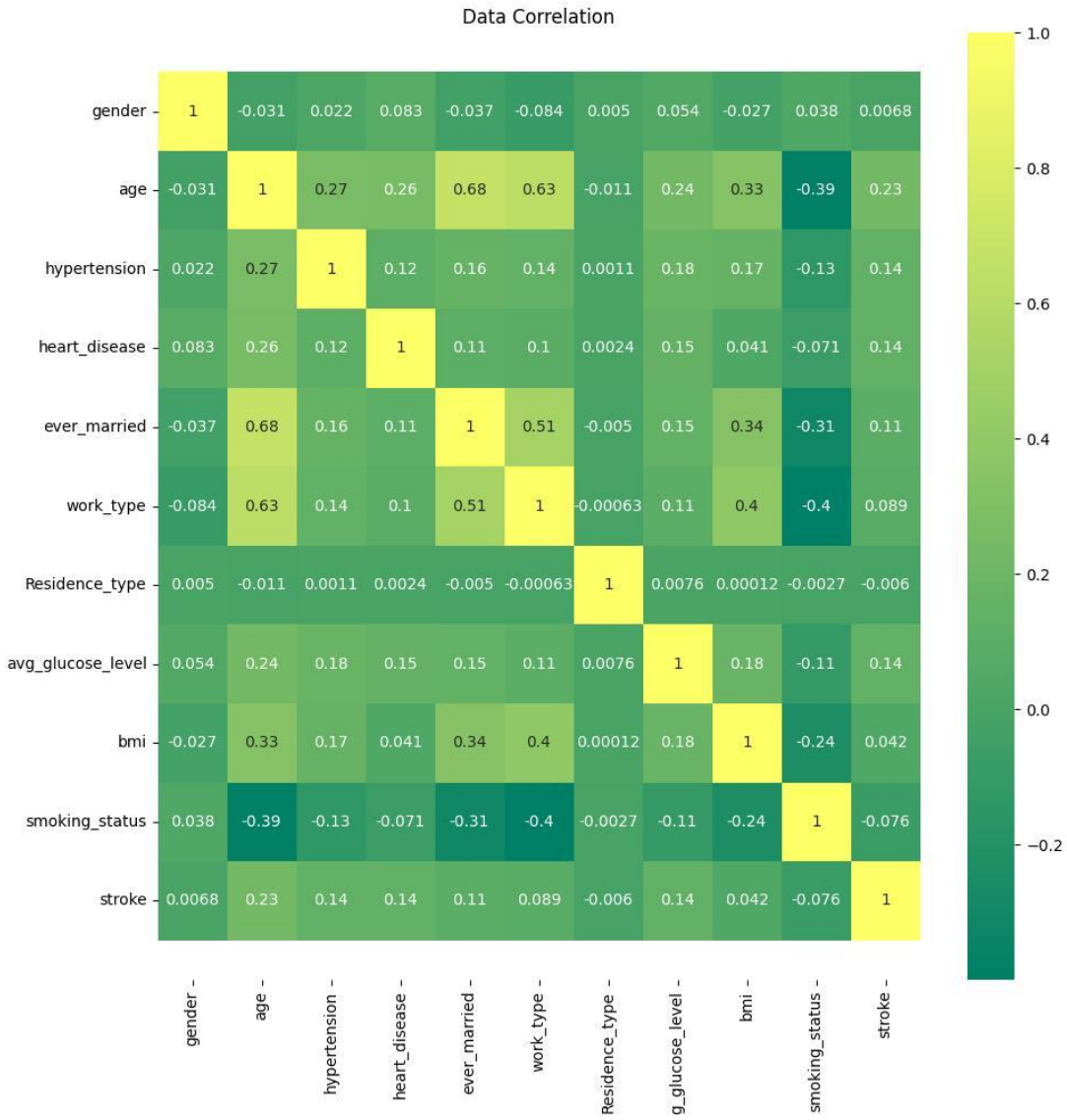
The third and final non-parametric classifier to analyze was the Support Vector Classifier using the Gaussian Radial Basis Function (RBF). This is one of the more popular kernels used in Support Vector Machines (SVM), and is most often used with non-linear data. For this SVM RBF kernel, there are two main parameters. The first parameter is “C” which controls how soft the dividing hyperplane is; higher values of C lead to a harder (less generalized) margin. The parameter of “C” is not specific to the RBF kernel. The second parameter is “gamma” which controls how close two points have to be in order to have similarity. The higher the value of gamma, the closer points must be to share similarity. To prevent the model from overfitting, we must be careful as higher values of “gamma” can lead to overfit data. As gamma increases, the region of similarity decreases. To think of this like a K-NN classifier, the higher that “gamma” becomes the lower the “k” value is. When compared to all other feature’s performance, the feature with the highest accuracy for predicting stroke was “age”. Using only the “age” and a balanced dataset, SVC-RBF kernel had an accuracy of 78%.

Algorithm	Accuracy	Best Single Feature for Accuracy
KNN	74%	age
Decision Tree	74%	age
SVM RBF	78%	age



After analyzing the non-parametric classification algorithms, the algorithm with the highest accuracy was the SVM RBF classifier. The AUC-ROC curves can reflect this characteristic as the SVC-RBF AUC was significantly higher than the other two algorithm. This is not a surprise as this is a very popular and common algorithm used in the field of data science.

Below you can see the correlation table for the entire (imbalanced) dataset. As can be seen from the chart, the highest correlation to stroke is “age”. Thus, it was no surprise that each and every algorithm found its best success with the “age” feature.



3.4 Dimensionality Reduction Techniques

Dimensionality reduction techniques are used to lower the dimension of the dataset with the purpose of increasing the predictive power of the model or improving processing capabilities by reducing the amount of space required by the data. It also can significantly increase performance and the speed of model training simply because the dataset is smaller and the model has less features to train on. In this project, we have chosen to use the feature selection technique. We believe that the features we have are enough to conduct our experiment and build good models.

We opted for the filter method instead of the wrapper or intrinsic method as both these methods involve some degree of selecting features best suited for a certain learning model and in this project, we wanted to apply the same data with the same features on multiple machine learning models and compare their predictive abilities.

Feature selection can be done on many basis. Logical deduction and how the features relate to the output is a valid method of feature selection. The features that

were deemed noise or not really relevant to the target variable or output were the gender, residence_type and ID. Logically, they don't seem to be strongly related to the output. Our reasoning for removing the features of gender and residence_type from our data analysis seems to be confirmed by the correlation table presented in part 3.2 of this report. We see that lowest correlation coefficients between the output or target attribute (stroke) and all other attributes are with residence_type (0.006) and gender (0.0068). These correlation coefficients confirm that these two features are the most weakly related out of all the other features to the output, so they can be removed. ID was removed because it is the unique identifier for a patient and has no meaningful connection to whether a person experiences a stroke or not. We decided that the other features are meaningful to the problem we are trying to address and cannot be reasonably discarded. Removing features based on correlation or logical deduction is a valid approach that we could take in reducing the dimension.

In this project we decided to apply the SelectKBest method from the scikit-learn library which aids in feature selection. The F-value between the output class and the is calculated to determine the best features in the dataset. The more informative a feature is, the higher its F value. 'K' represents the number of best parameters you choose to run your model on. It was set to 6 in this case, and these the six features with the highest F scores are selected. In this case, they are age, ever_married, work_type,hypertension, avg_glucose_level and heart_disease.

	Feat_names	F_Scores
0	age	195.245105
3	ever_married	32.307129
4	work_type	27.917520
1	hypertension	24.128139
5	avg_glucose_level	17.444857
2	heart_disease	11.467345

4.0 Experiment and Analysis

4.1 Experimental Setup

To start our model training, we have to convert the categorical value into the numerical value to perform Naive Bayes and RBF SVM models. These models are based on mathematical calculations and statistical analysis, typically performed on numerical data. Naive Bayes is a method that uses probability to predict the class of an input based on specific features. It is based on counting the frequency of different class and feature values in the training data. To perform Naive Bayes, it is necessary to convert any categorical variables to numerical values. On the other hand, RBF (Radial basis function) models are based on mathematical functions that use distance measures to calculate similarities between data points. The input features need to be numerical to perform this calculation.

We have to treat the BMI null values before training a model. It is essential to ensure that the model is trained on accurate data and to prevent errors or decreased performance. It is crucial to have a balanced dataset before performing any analysis. This is because imbalanced datasets can lead to several problems, such as bias, poor performance on the minority class, overfitting, sensitivity to the algorithm, and misleading metrics. A biased model can result in inaccurate predictions and poor performance of the minority class. Overfitting occurs when the model becomes too specialized to the majority class and cannot generalize well to new, unseen data. Many machine learning algorithms are sensitive to class imbalance and may perform poorly on imbalanced datasets. Some performance metrics, like accuracy, can be misleading when the data is imbalanced. Balancing the dataset can provide a fair representation of different classes to avoid these problems and improve generalization. We used the resample library to make a blanched dataset by giving the same size for each category.

To compare the result before balancing the dataset, we applied the grid search with the RBF algorithm to know the optimal C and gamma. After that, we resampled the data to make it balanced and used the same C and gamma values for the RBF algorithm.

Furthermore, we used the SelectKBest method from the scikit-learn library to perform feature selection on the training data. The method uses the f_classif score function based on the F-value between the label/feature for classification tasks. The number of features to select is specified by the 'k' parameter, which is set to 6. This method is used to determine the most informative features that have a higher F-value score. This will help in reducing the dimensionality of the dataset and also improve the performance of the classifier.

Since our dataset aims to predict stroke patients, the accuracy is not enough to evaluate the accuracy of the machine learning model. We have chosen the confusion matrix to evaluate our model by considering the: Precision for class (1), Recall for class (1) , FP Rate, and Accuracy.

4.2 Result Analysis

Gaussian Naive Bayes algorithm

The Naive Bayes model showed an accuracy of 91.5%. Furthermore, the model could predict only 27.6% of stroke patients correctly. The model's performance is considered poor since it is chosen that the model is biased toward the non-stroke patient since it is the majority. We can see the model is overfitting as the accuracy of the model is high, but the ability to differentiate between the classes is not well. The lousy performance of Naive Bayes classifiers is because the classifier is more likely to predict the majority class due to its higher prior probability. Also, the independence assumption of the features may not be able to capture the relationship between features in unbalanced datasets.

Table 4.1: Gaussian Naive Bayes with default configuration			
Precision	Recall	FP Rate	Accuracy
22.1%	27.6%	72%	91.5%

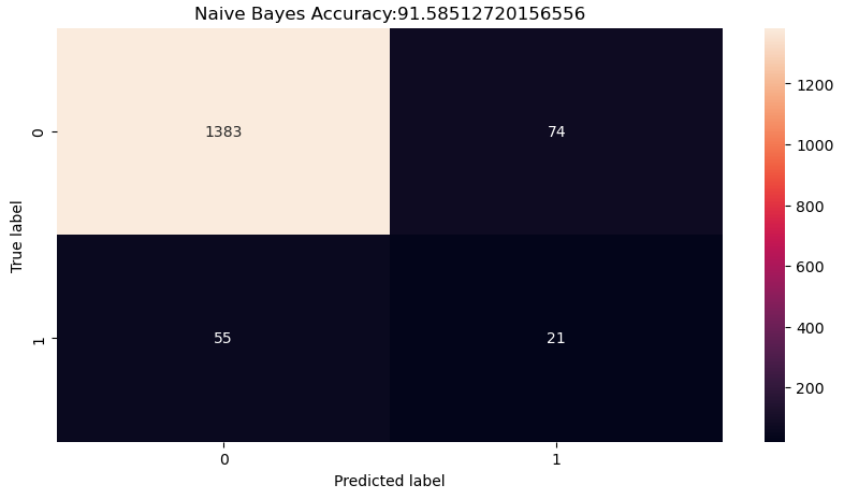


Figure 4.1: Gaussian Naive Bayes with default configuration

RBF SVM algorithm

The RBF shows a higher accuracy with 94.9% compared to the naive Bayes model. However, the model's recall is very bad as it identified only one patient with stroke over 76 patients. The model assumed that all the data belonged to the same class due to the imbalanced dataset. The RBF classifier will be more likely to predict the majority class, regardless of the input features. This is because the model tries to maximize the margin between the two classes, and the majority class is likelier to be within the margin. This can result in the majority class being predicted more often, leading to a high false positive rate (FPR).

Table 4.2: RBF SVM with default configuration

Precision	Recall	FP Rate	Accuracy
33.3%	1.4%	98.6%	94.9%

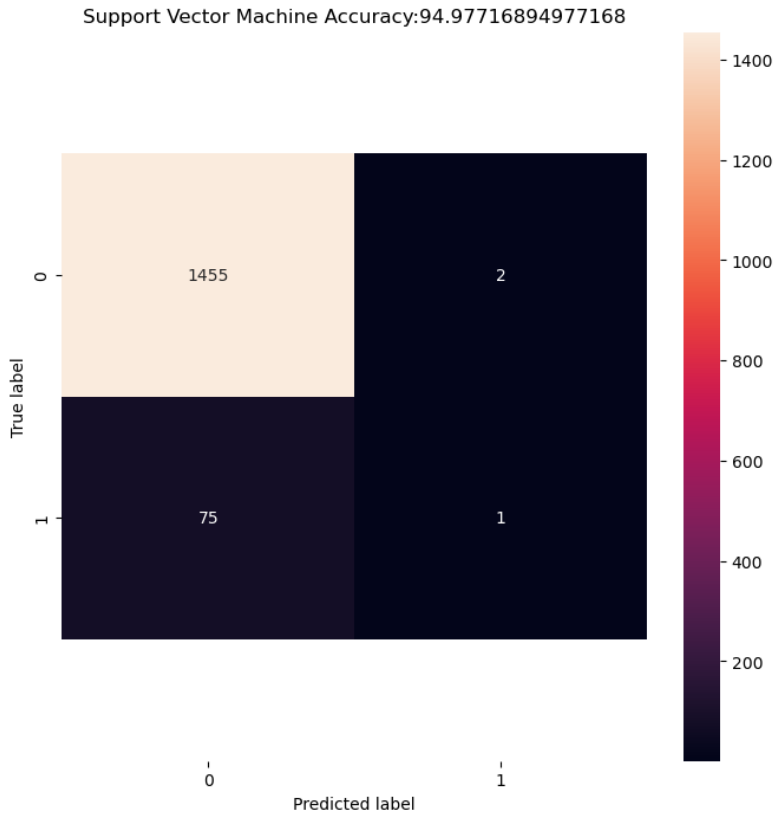


Figure 4.2: RBF SVM with default configuration

Gaussian Naive Bayes algorithm with balanced data

The recall of the Naive Bayes increased to 88% after balancing the dataset, showing that the model can predict 88% of stroke patients correctly. The accuracy has dropped by more than 10% due to reducing the number of examples for the majority class, which can decrease the classifier's ability to predict the majority class accurately. Therefore, the classifier may not generalize well to new examples from the majority class, which can lead to a decrease in overall accuracy.

Table 4.3:Balanced Gaussian Naive Bayes

Precision	Recall	FP Rate	Accuracy
73.6%	88%	12%	78%

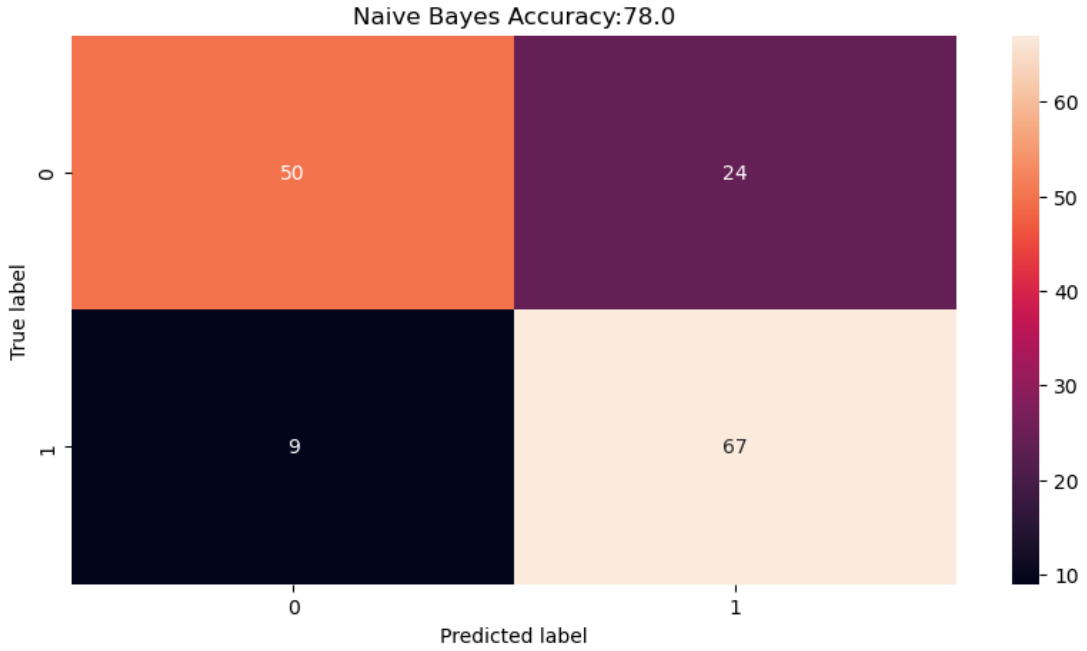


Figure 4.3: Gaussian Naive Bayes with balanced dataset

RBF SVM algorithm with balanced data

The recall of the RBF model increased to 59%, which is a good improvement but not good enough to use it as a model to predict stroke patients. The accuracy has dropped significantly from 94% to 54 due to the complexity of the features and reducing the number of samples. So we have to reduce the complexity of the model to improve its recall and accuracy.

Table 4.4: Balanced RBF SVM			
Precision	Recall	FP Rate	Accuracy

54.2%	59.2%	40.8%	54%

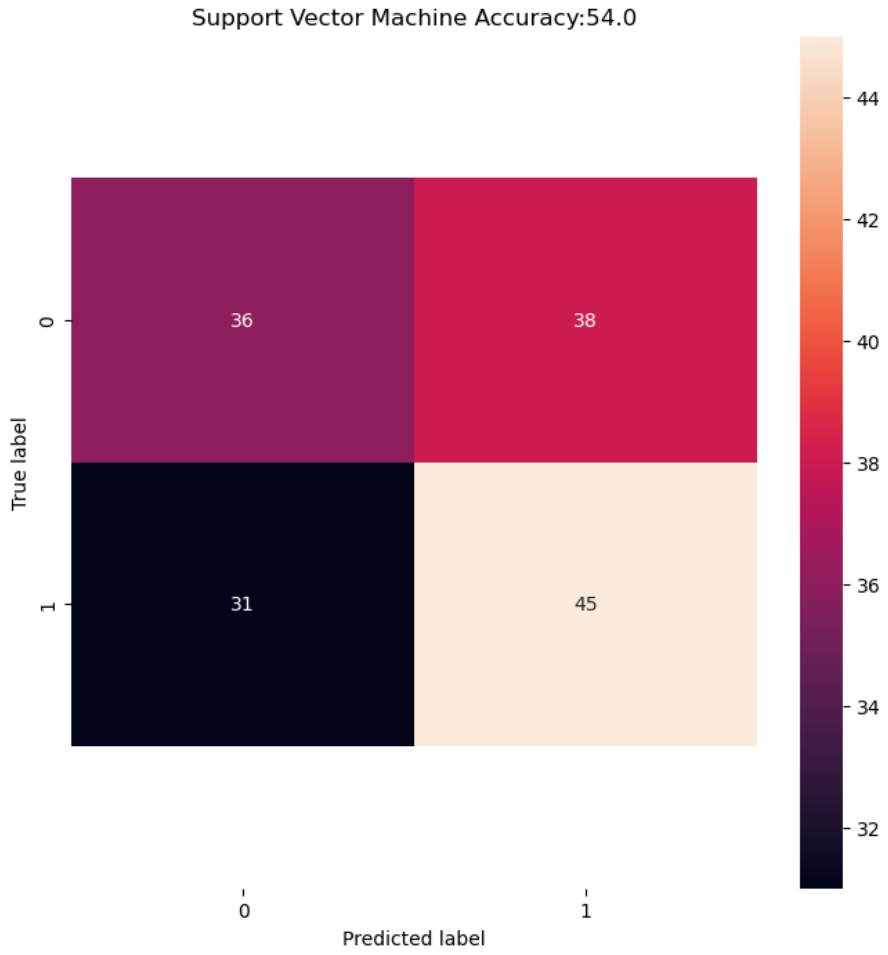


Figure 4.4: RBF SVM with balanced dataset

Gaussian Naive Bayes algorithm with balanced data and feature selection

After performing the feature selection, the accuracy increased by 0.6%. However, the specification dropped by 5%. Removing features that are not informative can help the classifier to focus on the most relevant features, which can increase the accuracy of the classifier. Additionally, suppose the removed features are correlated with the minority class. This could lead to a false positive rate (FPR) increment as the classifier will not rely on those correlated features to classify the data. Overall the Naive Beyes can predict 83% of the patients with stroke with an accuracy of 78.6%, which is considered a good model that doesn't suffer from underfitting or overfitting.

Table 4.5: Balanced Gaussian Naive Bayes with feature selection

Precision	Recall	FP Rate	Accuracy
76.8%	83%	17%	78.6%

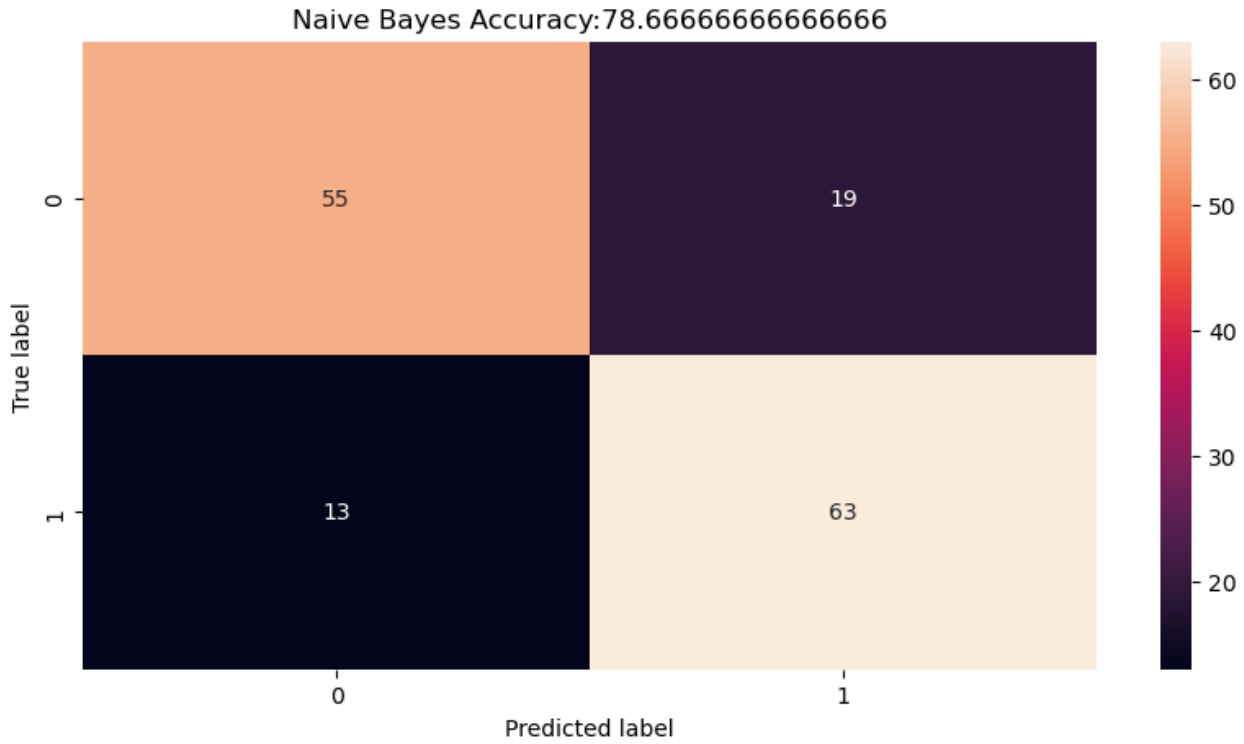


Figure 4.5: Gaussian Naive Bayes with balanced dataset and feature selection

RBF SVM algorithm with balanced data and feature selection

The recall of the RBF increased dramatically from 59% to 85.5% after performing the feature selection. Moreover, the accuracy jumped from 54% to 78.6%. Removing some non-informative features from the dataset can improve the classifier's performance by reducing the dimensionality of the data, reducing the chances of overfitting or underfitting the data, and focusing on the most relevant features, which can increase the accuracy and reduce the FP Rate.

Table 4.6: Balanced RBF SVM with feature selection

Precision	Recall	FP Rate	Accuracy
75.6%	85.5%	14.5%	78.6%

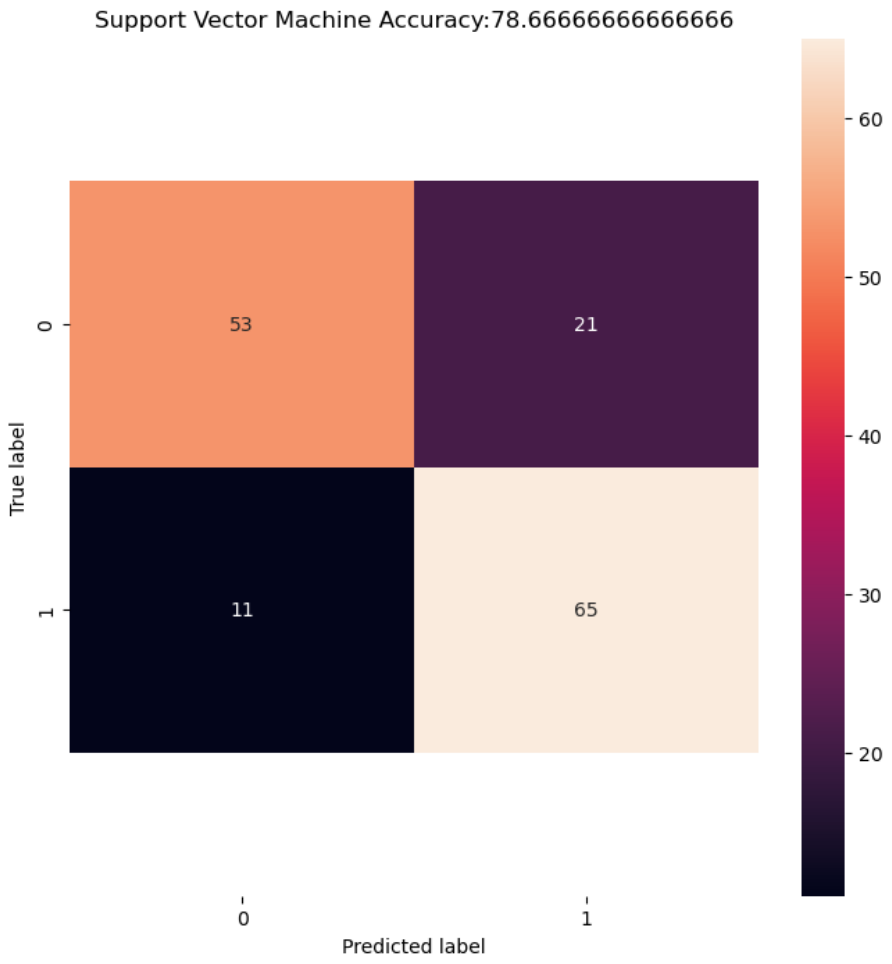


Figure 4.4: RBF SVM with balanced dataset and feature selection

4.3 Comparing Gaussian Naive Bayes and RBF SVM after data Balancing and features selection

RBF Support Vector Machine (SVM) provides better results than the Naive Bayes algorithm with a balanced dataset and feature selection.
One primary reason for this is the ability of RBF SVM to create non-linear decision boundaries, which can better separate the classes in the dataset compared to the linear decision boundaries of Naive Bayes. The RBF kernel in SVM allows for a more complex and flexible model that can capture complex patterns in the data that a linear model like Naive Bayes might miss.

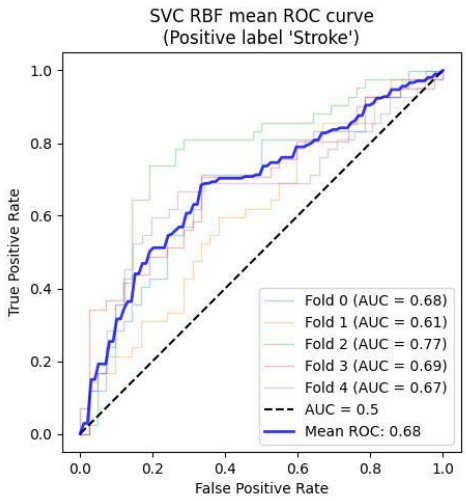
Additionally, SVM is less sensitive to outliers and noise in the data, which can lead to more robust performance. This is particularly important when working with datasets that contain outliers or noise, which can negatively impact the performance of a linear model like Naive Bayes.
SVM is also better suited for handling high dimensional data, as it tries to find the best boundary between classes. On the other hand, Naive Bayes assumes independence between features and can struggle with high-dimensional data.
In summary, SVM is less sensitive to outliers and noise and can handle high-dimensional data better. The non-linear decision boundary of the RBF kernel also allows SVM to separate the classes in the dataset better. These factors combined make SVM a more suitable algorithm for achieving high accuracy and a low false positive rate compared to Naive Bayes, even with a balanced dataset and feature selection.

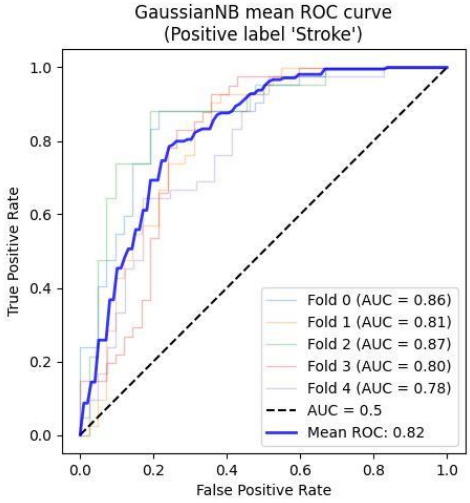
5.0 Conclusion

In this paper, two machine learning models for the prediction of brain strokes were presented. Brain strokes are a dangerous and common condition with a high fatality rate. Improving the process of diagnosing the threat of brain strike for a patient is imperative. This project was conducted with the goal of building machine learning models that are capable of predicting whether a patient has experienced a stroke before or not based on some personal and medical data. An objective of this project was that the model is able to predict brain strokes in patients before they happen, therefore lessening the medical risks associated with the condition and the cost of the treatment. The dataset was explored comprehensively and pre-processing options were also explained. A parametric algorithm: Naive Bayes and a non-parametric algorithm: RBF

SVM were chosen to conduct the experiment.The choice of machine learning models and dimensionality reduction were justified with the respect to the dataset.
In the experiment and analysis, the results were stated clearly and the two algorithms were compared. RBF SVM was determined to be a better fit for our dataset and project goal.

In the future, this model can be applied to other datasets and see if the models' predictive power holds up on different and larger data sources. It is our hope that such models as the ones presented in this paper can actually be of use to medical professionals and help prevent unnecessary casualties or physical suffering of patients. Instead, preventative measures can be taken to avoid that pain if a patient is predicted to have a high chance of having a stroke.





References

References

Almeida, Y., Sirsat, M., & Fermé, E. (2020). *Publication Details*. SciTePress. Retrieved November 21, 2022, from https://www.scitepress.org/Link.aspx?doi=10.5220/0009369108450853

Benjamin, J, E., & Virani. (2018). Heart Disease and Stroke Statistics—2018 Update: A Report From the American Heart Association. *Heart Disease and Stroke Statistics—2018 Update: A Report From the American Heart Association* , *137*(Circulation), e67--e492. https://www.ahajournals.org/doi/full/10.1161/CIR.0000000000000558#

Di Carlo, A. (2009, Jan). *Human and economic burden of stroke*. PubMed. Retrieved November 21, 2022, from https://pubmed.ncbi.nlm.nih.gov/19141505/

Merino, J. G. (2014). *Clinical stroke challenges: A practical approach*. PubMed. Retrieved November 21, 2022, from https://pubmed.ncbi.nlm.nih.gov/29443247/

R, G. (2001). *To Err Is Human. Eds. L. T. Kohn, J. M. Corrigan and M. S. Donaldson.* (Vol. 6). National Academy Press Washington. 10.1017/S095026880100509X

Stroke - What Is a Stroke? (2022, March 24). NHLBI. Retrieved November 21, 2022, from https://www.nhlbi.nih.gov/health/stroke

Suwanwela, N. C., Poungvarin, N., & Asian Stroke Advisory Panel. (2016). *Stroke burden and stroke care system in Asia*. PubMed. Retrieved November 21, 2022, from https://pubmed.ncbi.nlm.nih.gov/26954968/

WHO. (2004). *The global burden of disease : 2004 update*. World Health Organization (WHO). Retrieved November 21, 2022, from https://www.who.int/publications/i/item/9789241563710

A. Sudha, P. Gayathri, & N. Jaisankar (2012) . *Effective analysis and predictive model of stroke disease using classification methods*. International Journal of Computer Applications, vol. 43, no. 14, pp. 26– 31. Retrieved November 23, 2022 from https://www.ijcaonline.org/archives/volume43/number14/6172-8599

L. Amini, R. Azarpazhouh, M. T. Farzadfar, S. A. Mousavi, F. Jazaieri, F. Khorvash, R. Norouzi, & N. Toghianfar (2013). *Prediction and control of stroke by data mining*. International Journal of Preventive Medicine, vol. 4, no. Suppl 2, pp. S245–249. Retrieved November 23, 2022 from https://pubmed.ncbi.nlm.nih.gov/23776732/

S. Y. Adam, A. Yousif & M. B. Bashir (2016). *Classification of ischemic stroke using machine learning algorithms.* Int J Comput Appl, vol. 149, no. 10, pp. 26–31. Retrieved November 23, 2022 from https://www.ijcaonline.org/archives/volume149/number10/26035-2016911607

R. Jeena & S. Kumar (2016). *Stroke prediction using svm*. 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 600–602, IEEE. Retrieved November 23, 2022 from https://ieeexplore.ieee.org/abstract/document/7988020

P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi, R. Patan, P. Jayaraman & R. Manikandan(2019) . *Classification of stroke disease using machine learning algorithms*. Neural Computing and Applications, pp. 1–12. Retrieved November 25, 2022 from https://link.springer.com/article/10.1007/s00521-019-04041-y

T. Kansadub, S. Thammaboosadee, S. Kiattisin & C. Jalayondeja (2015). *Stroke risk prediction model based on demographic data*. 2015 8th Biomedical Engineering International Conference (BMEiCON), pp. 1–3. Retrieved November 25, 2022 from https://ieeexplore.ieee.org/document/7399556

Singh, M. S., & Choudhary, (2017). *Stroke prediction using artificial intelligence.* 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON). Retrieved November 25, 2022 from https://www.semanticscholar.org/paper/Stroke-prediction-using-artificial-intelligence-Singh-Choudhary/73a8b4778bc802c17019a664df3d2781a4d01896

Chiun-Li Chin, Bing-Jhang ,LinGuei-Ru Wu, Tzu-Chieh Weng, Cheng-Shiun Yang, Rui-Cih Su, Yu-Jen Pan (2017). "An automated early ischemic stroke detection system using CNN deep learning algorithm". 2017 IEEE 8th International Conference on Awareness Science and Technology. Retrieved November 25, 2022 from https://ieeexplore.ieee.org/document/8256481

Appendix

Group Contribution

Ahmed Adel Sanhan Al-Haidary- Abstract
Ahmed Adel Sanhan Al-Haidary- Abstract1.0 - 1.3
Samira Elsamad- 2.0
Austin Smith - 3.0 - 3.3
Samira Elsamad - 3.4
Ahmed Adel Sanhan Al-Haidary- 4.0 - 4.3
Samira Elsamad - 5.0
Samira Elsamad - Poster