How accurate can a machine learning model be in predicting whether someone may expereince a stroke?

Orfeas Gkourlias - 420172

2022-11-08

1 Preface

The project and paper were written for an assignment given by Hanzehogeschool Groningen. With the educational goal of providing a learning experience regarding machine learning and some Java. We were given the opportunity to choose any data set to our liking for the project, adhering to the requirements set, of course.

This made it so that the entire experience was more interesting, because of the agency in choosing something that interested me. Sadly, the first data set I had chosen did not meet the requirements, resulting in me lagging behind for a little while. Luckily, I was able to catch up with another interesting subject, concerning stroke patients. This subject intrigues me tremendously, because of people in my environment who have had the misfortune of suffering a stroke. Considering the ever growing importance and emphasizes on preventative care, and the significance of strokes on people, I thought this would be a perfect candidate for the machine learning project. I would like to thank my teachers for the guidance they offered in execution of the project.

2 Abstract

Every 40 seconds someone in the united states suffers from a stroke. With around 1 out of 5 people passing away as a result. The outlook for stroke patients are also varying in severity for the survivors. With extensive aftercare almost being a guarantee in all cases, strokes are also the leading cause of serious, long-term disabilities and impairments. With only 1 out of 10 patients making a full recovery.

The severity and prevalence of strokes cannot be understated. While modern aftercare usually provides favorable prognosis, the complete prevention of not only strokes, but many cardiovascular conditions has long been advocated for. The importance of preventative healthcare keeps being re-emphasized as people grow older. There are many preventative measures available and still being developed.

Considering the above, the ability to predict whether someone may suffer from a stroke would be an invaluable tool. This project has aimed to do just that. By taking patient data, examining that and then training a machine learning model to make predictions based off it. Using the powerful machine-learning and data analysis program called Weka, a prediction model was developed and trained to be 92 percent accurate in its classification. With emphasis on avoiding false negatives, the model will be identify high risk patients with utmost care.

If this model could be made even more accurate, then it might serve as a primary detector of high risk stroke patients. Although a wide array of different methodologies and algorithms were tried, there might still be facets of the project which could have been done differently. Further examination of the model would maybe be able to push it past even the 95 percent accuracy mark.

To compliment the model, a java program was also developed. This program takes the model, new instances of data with similar structure, and makes a prediction on all of the instances. This way many entries from big data-sets can be classified at once.

3 Abbreviations

SMOTE : Synthetic Minority Oversampling Technique BMI : BOdy mass index

4 Table of Contents

Contents

5 Introduction

Stroke is one of the leading causes of death and disability worldwide, ranking at second for amount of deaths and third for death and disabilities combined World Stroke Organization (WSO): Global Stroke Fact Sheet 2022. The best solution to this worldwide problem will be preventative care. As high as 80% of all the stroke cases are preventable ones. Preventing Stroke Deaths The correlation between stroke risk and attributes such as BMI and blood pressure are already well observed. These metrics are useful in predicting a person's likelihood of suffering from a stroke. While this may be done on an individual basis, a computer could in theory also make these predictions, given the required metrics. This is where machine learning and statistical inference could be paramount in preventative care. The total amount of stroke patients may be reduced significantly,by applying a machine learning model and predicting for multiple people whether they are at risk of suffering from a stroke.

5.1 Objective

The objective of this project is to develop and train a model, which can accurately predict whether a person is likely to suffer from a stroke. This can be achieved using machine learning algorithms, which are already present in the program named Weka. This final model may then be used in a Java program, which predicts for every given person whether they are likely to suffer from a stroke.

5.2 Theory

As mentioned before, preventing strokes will be the most efficient way to combat them. This can be done in many ways, but detecting that on a large scale is already quite an undertaking. Doctors can already provide sufficient diagnosis of attributes such as elevated blood pressures and increased BMIs. But to do this for every single individual would require too many people. Patients are often already unaware of elevated blood pressures, hence why it is called a silent killer. Application of a machine learning model allows for the diagnosis process to be applied on a greater scale. Taking thousands of patient and predicting their individual risks is faster and more efficient.

Machine learning can be a complicated process, with many different ways of approaching the training of a model. First training data needs to be collected. In this case, data was taken from kaggle. This data contains 12 attributes, with one being the classifier: stroke history. 5110 instances are present in the main dataset.

Column	Unit	Description
ID	Number	Unique patient identifier
Gender	Text	"Male", "Female" or "Other"
Age	Number	Age of patient
Hypertension	Boolean	Whether patient has hypertension
Heart_disease	Boolean	Whether patient has a heart disease
Ever_married	Boolean	Whether patient has ever been married
Work_type	Text	Occupation status of patient
Residence_type	Text	Patient living enviroment
Avg_glucose_level	Number	Average glucose level in blood
BMI	Number	Body mass index of patient
Smoking_status	Boolean	Whether patient smokes or not
Stroke	Boolean	Whether patient has ever experienced a stroke

Some of the attributes can directly be identified as correlated to an increased risk of having a stroke. Other attributes are less obvious contenders, such as the marriage status. The last attribute is the one for which will be predicted. In this case, it is a binary classification, with stroke either indicating yes or no. This was an important consideration when choosing an algorithm, because not every algorithm is as effective as another when it comes to classification problems. The accuracy, meaning how many of the instances a model predicts correctly is of high importance. Something else which was closely examined was the false negative rate. This rate must be as low as possible without compromising much of the accuracy, because false negatives are more destructive to a person than a false positive in this case. The ROC curve was therefore also an important facet of the end result.

While the data itself is of good quality, there was one concern to be had. The classifier was quite uneven in distribution. With 95% of the instances being classified as no, with the remaining 5% being a no. While maybe realistic, it would skew the final model too much, resulting in a lower accuracy. Bringing the balance closer to 70/30 would be better for training purposes. To achieve this, the synthetic minority oversampling technique (SMOTE) was used. As the technique implies, it increases the number of total cases in a balanced way.

The final data is then prepared for model training, which as mentioned before will be aimed at increasing the accuracy while reducing the false negative rate.

6 Materials and methods

The entirety of this project, including both the machine learning and java part , was developed in a github repository. The ml directory contains the machine learning section, with the java directory containing the entire java program. This project was also performed by one student.

6.1 Material

The project consists of multiple programs and tools. The versions of the software and tools are important, especially in the case of the java program. A table will be shown with all the software which was used.

Software	Version	Function
R	4.2.1	Statistical analysis language.
Weka	3.8.6	Machine learning platform.
Java	17.0.5	Application programming language.
Gradle	7.4	Application builder.

The R and Java software also require specific packages. Packages required for R are as following.

R Package	Version	Function
readr	2.1.2	Read rectangular text data.
ggplot2	3.3.6	Creating visualizations.
tidyr	1.2.1	Tidy messy data.
lemon	0.4.5	Ggplot2 exstension.
Rweka	0.4-44	Weka interface for R.
DMwR	0.4.1	Functions for data mining.
Hmisc	4.7-1	Harrell miscellaneous.
caTools	1.18.2	Statistical tools.

Followed by the Java packages used in development.

Java Package	Version	Function
Weka	3.8.0	Weka API for Java.
Apache Commons Cli	1.5.0	CLI Api for Java.

A unix based system is recommended in combination with the above software.

6.2 Data

The data set was taken from kaggle. It can be downloaded in a .csv format. The file itself has been renamed to stroke-data.csv. The theory behind the attributes from the dataset is explained in the theory section.

6.3 Methodology

This project consists of multiple methods which had been developed prior. The data pre-processing contains SMOTE, which is a widely used technique for imbalanced data. The remaining pre-processing operations were done with R and the aforementioned packages. The data was already very usable on its own. The only thing performed besides SMOTE for cleaning purposes was the deletion of the ID column. This was not relevant for the algorithm training.

The order in which things were done as follows. First, the data was downloaded from the kaggle link. The data was then loaded into the journal.rmd file. EDA was first performed in this file. The results can be found in the corresponding journal.pdf file. Statistical analysis was done with standard t-tests and functions from the base R language. The analysis was done to find correlations in attributes before the data would be used in Weka.

After the analysis and pre-processing steps, Weka comes into play for the machine learning section. The new cleaned up data (train.arff) was loaded into the weka program. First, with the explorer to individually test different algorithms. The algorithms themselves are all built into the Weka program. The explorer was used in the program to get a feel for what might be feasible and what might not be. All algorithm tests in this and the following steps were done with 10-fold cross validation.

After deducing the most likely algorithm candidates from the explorer, they were all ready to be tested against each other. This was done in the weka experimenter. The weka experimenter allows for multiple algorithms to be selected at once, so that they may be tested and compared with eachother. This was done with the following algorithms: ZeroR, OneR, NaiveBayes, SimpleLogistic, SMO, IBk, J48 and RandomForest. This was done with default settings, with ZeroR being the comparison baseline. Because of the importance of false negatives, all the above algorithms were tested again but with cost sensitivity. The scoring matrix being:

0	2
1	0

Which indicates that the false negatives will be weighted more than the other hits.

Once the best two algorithsm were found, that being OneR and RandomForest, they were once again compared in the experimenter. But this time with differing settings. Every default setting was changed from 0 to 1 in turn. The results were underwhelming, and the default settings were therefore retained after examination. This resulted ina model with 92% accuracy, with usage of the RandomForest algorithm. This model was exported to a .model file, which can be found in resources under the java directory.

6.4 Java

The java program was developed with the intent of allowing for new classifications, using the model from the last step. This model can be loaded into java with the Weka api. The weka Api contains functions which can both read and write to .arff files. Once the user input was taken as a .arff file with the Apache api, the classification of new instances was performed by loading the .model file with the Weka api and that. The output of this process is saved in a new .arff file, with the new classifications for every row/instance.

The java program was built using gradle and the shadow plugin. The shadow plugin allows for building of fatjars, which don't require the user to have downloaded the dependencies beforehand. The entirety of the program can be found under the java/wrapper directory path.

7 Results

The results of the project will be shown here, with the following structure. First the data exploration and its results will be explained and shown. Following that, the results from the machine learning and model training will be displayed. Ending with a display of the final java program which allows for new classifications.

7.1 Data Analysis

Following the analysis of the data set and the attributes within, some important correlations can be observed. Correlations relevant to the research question mostly consist of heart and health attributes. The first plot regarding heart disease differences in non-stroke and stroke patients will be shown.

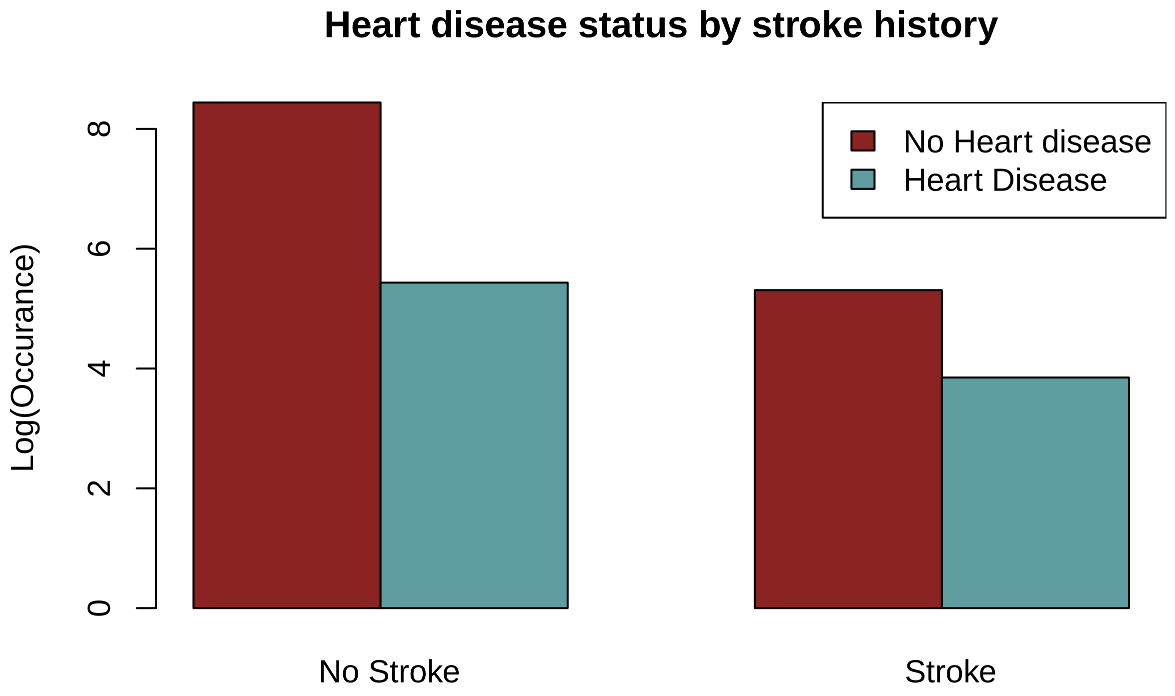

Figure 1: Barplot of heart disease status by stroke history.

This plot of normalized data shows that there are proportionally more heart disease patients in the stroke group, than there are in the no stroke group. This ratio can be translated to a percentage. The percentage of patients who have not had a stroke is 4.71%. This percen tage is higher for the group of patients who have experienced a stroke. For that group the percentage is 18.88%. This means that it is 4 times as likely for someone who has had a stroke, to also have a heart condition. The literature on this subject seems to have come to a common consensus: There is a correlation between having a heart disease and a higher risk of a stroke, This paper published by the American Heart Association being one of the bigger published works on the topic: Heart Disease and Stroke Statistics—2020 Update: A Report From the American Heart Association. Keep in mind, the risk of developing heart disease after stroke is also quite high.

The same can be seen in the case of hypertension, because that is also a heart condition.

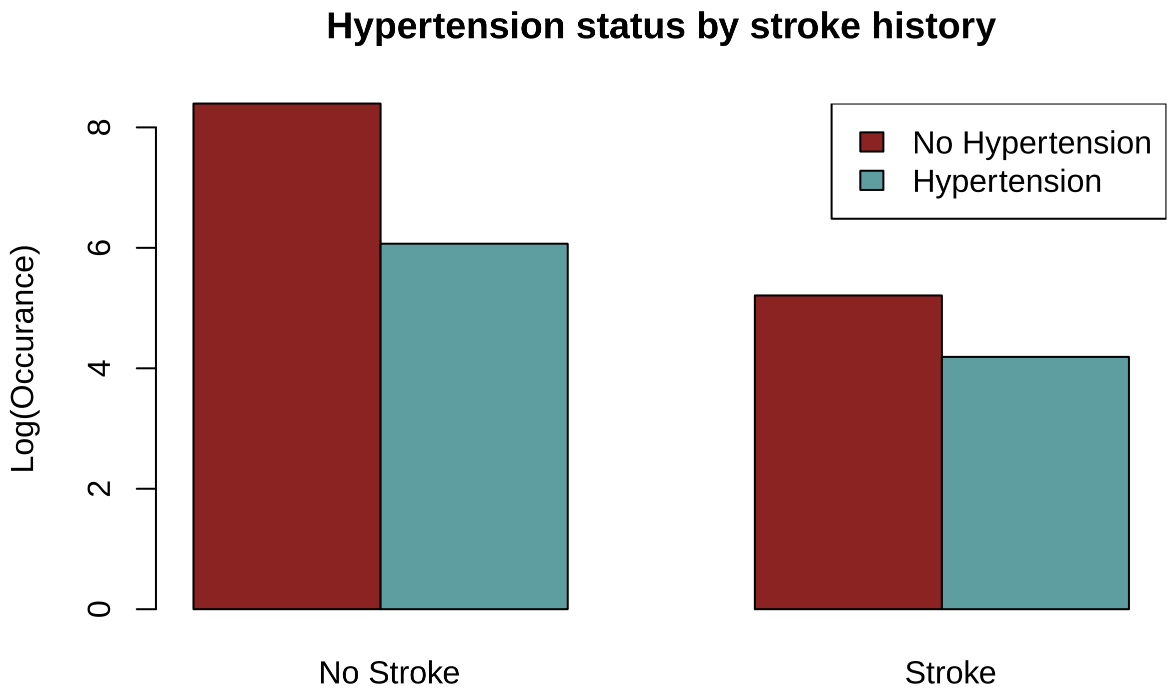

Figure 2: Barplot of hypertension status by stroke history.

This plot is very similar to the prior one. That makes sense, because quite some heart conditions are accompanied by hypertension. Hypertension on its own is already classified as a heart condition too.

Age has been observed to also be an important attribute. The older a person beco mes, the higher the likelihood of a stroke. Especially when combined with other attributes, such as hypertension or heart conditions. According to this paper, the odds of having a stroke double for every ten years a person lives. Aging and ischemic stroke

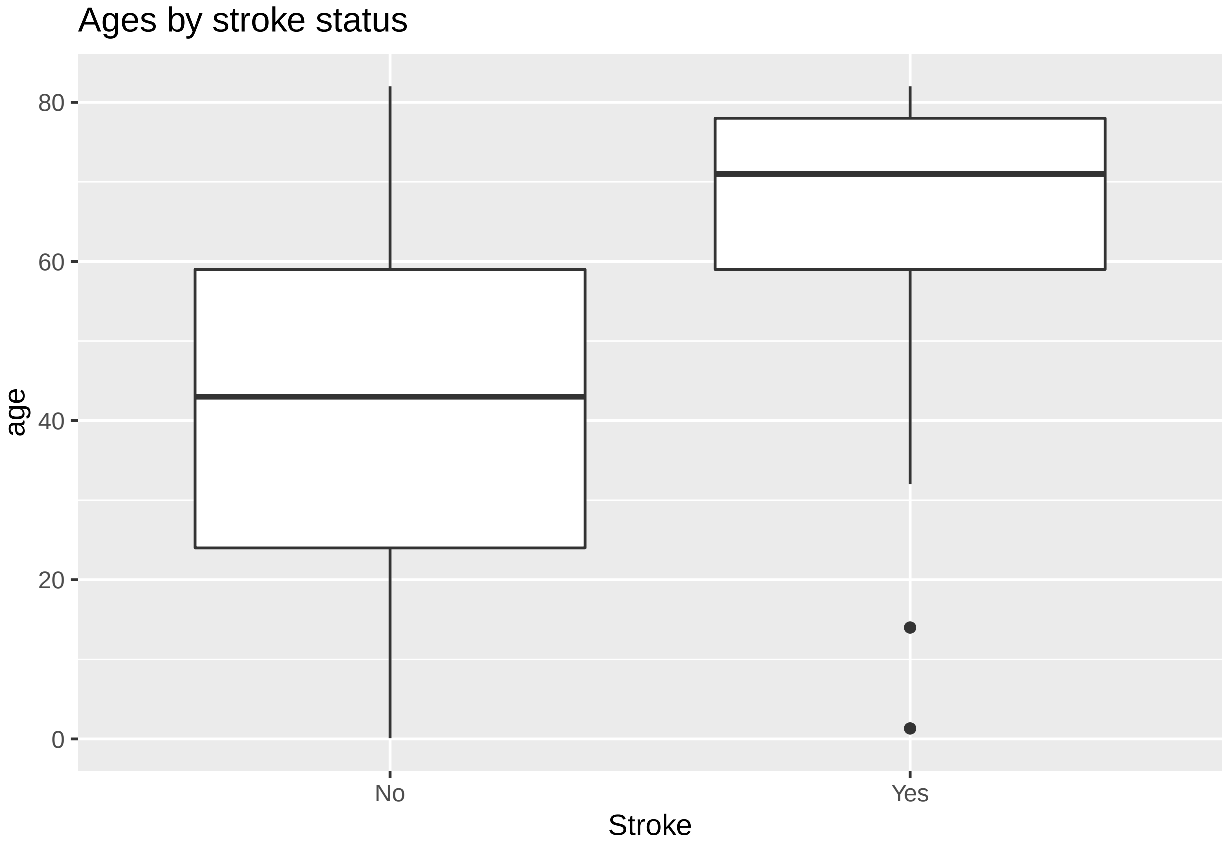

Figure 3: Boxplot of age by stroke status.

This shows that the older a person is, the more likely they are to have experienced a stroke. This would of course make sense, considering that the older someone gets, the higher the chance one might encounter health issues.

Another health related observation that may be relevant is that of the BMI means for stroke and no stroke patients. This can be visualized through a similar box plot as the prior one.

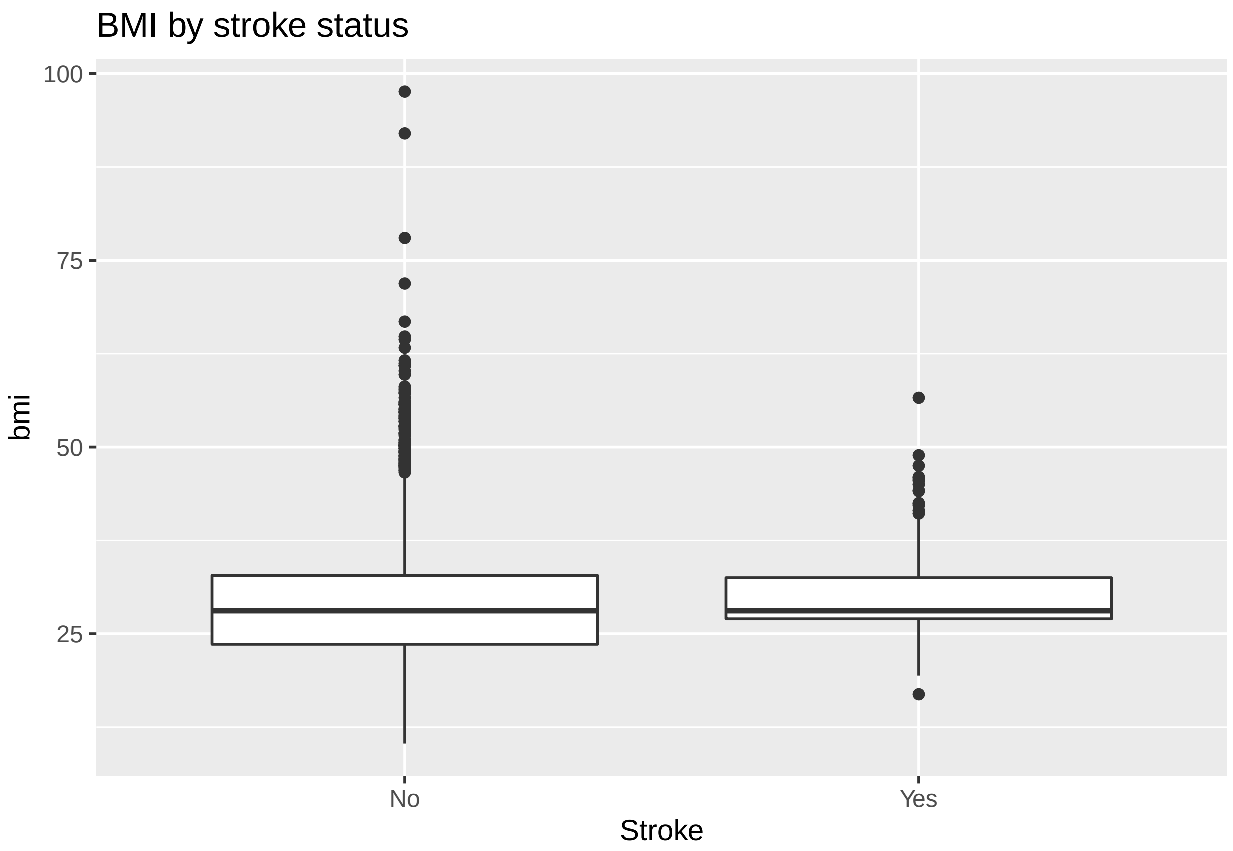

Figure 4: Boxplot of BMI by strike status.

The patients who have had a stroke only have a slightly higher BMI. The no stroke group also has more extreme values on both ends.

7.2 Imbalance

Something important which can be observed in the data is the imbalance between stroke instances. A bar plot will be used to demonstrate this imbalance

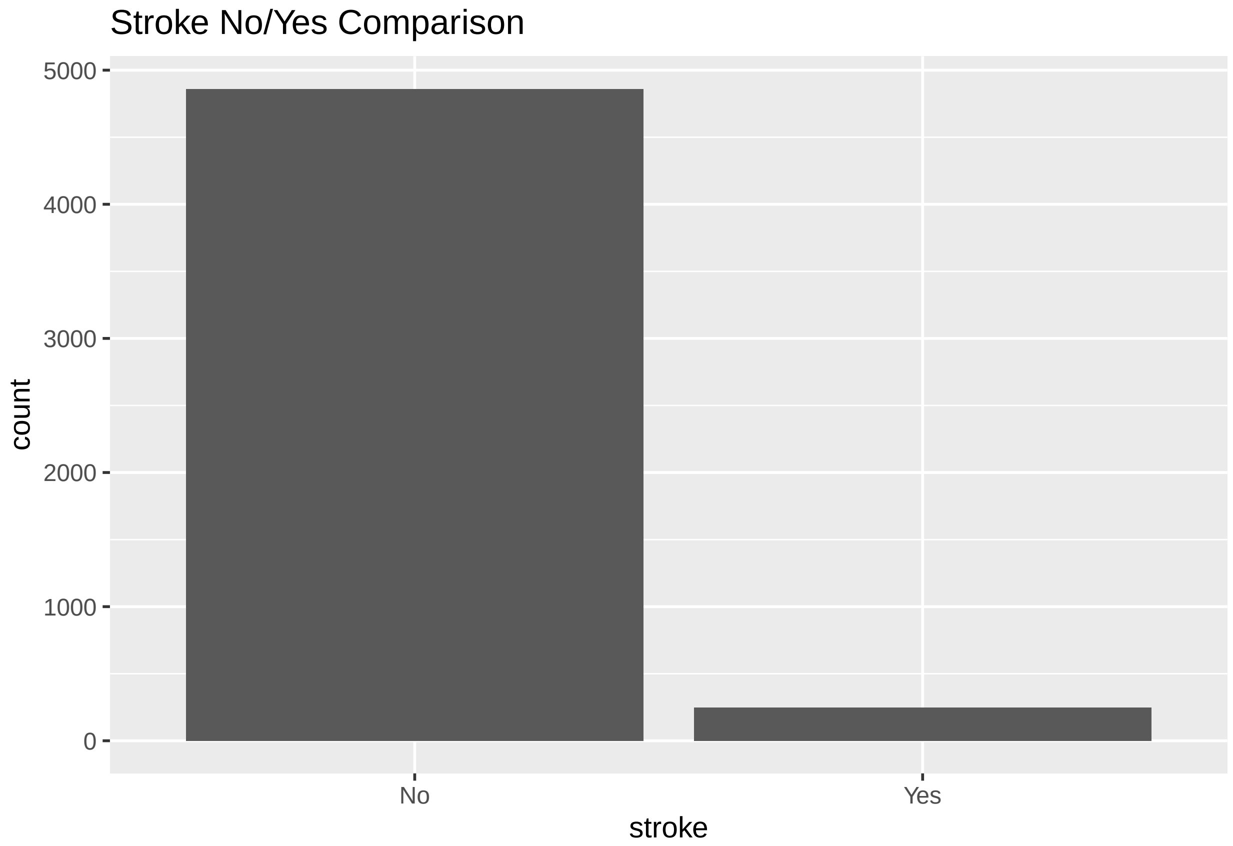

Figure 5: Barplot of classifier balance.

As can be seen, there is a great imbalance in the amount of patients who have had a stroke those who have not. To be exact: 0% of the data, consists of rows where stroke is equal to 0. The remaining 5% has experienced a stroke. This will be very important when selecting for different machine learning algorithms, because the weighting of occurrences will most likely have to be changed in this case.

7.3 Machine Learning

Multiple algorithms were performed, which all netted different results. A table will be shown with the differing algorithm metrics for them all.

Table 6: Weka experimenter algorithm with cost sensitive classifier results showing accuracy, false negative rate and the AuROC. ZeroR being the basline for which improvement or degradation is shown.

Algorithm	ZeroR	OneR	NaiveBayes	SimpleLogistic	SMO	IBk	J48	RandomForest
Accuracy	65.45	69.92 ○	81.19 ○	79.09 ○	78.47 ○	89.85 ○	86.25 ○	91.67 ○
False Negative Rate	0.00	0.08 ○	0.15 ○	0.07 ○	0.06 ○	0.05 ○	0.05 ○	0.03 ○
AuROC	0.50	0.60 ○	0.88 ○	0.90 ○	0.72 ○	0.88 ○	0.89 ○	0.98 ○

○, ● statistically significant improvement or degradation

These are the final results of the algorithm experimenter in Weka. The goal of the project was to find the most accurate model, with the least amount of false negatves. This means that the RandomForest algorithm is the best out of all of the algorithms. This is also supported by the high AuROC value.

7.4 Java Program

The finished java program can predict for multiple new instances whether a stroke is likely or not. The program can be executed from a .jar file. By giving the new instances which have yet to be classified, and the desired output file/location, a new file will be generated. This is an .arff file which shows the predicted classification for every instance/row. The program itself is built out of three classes. Which are all called upon in the main class.

8 Discussion and Conclusion

While the final product is quite accurate, it might be possible that with another approach to the algorithm testing, an even better model may be trained. Future research on the same data set maybe prove that something was not considered or recognized in this project, which would lead to higher accuracy if adjustments are made. But until proven otherwise, the initial goal of creating an accurate model has been reached.

8.1 Data

Now that the most important results have been shown, some points of contention could be considered. The aim of this project is to answer the original research question using machine learning. This may be done with varying algorithms and considering different attributes. Looking at the results, some of the attribute s may seem irrelevant at first, and some would argue these could be removed before applying any machine learning algorithms. But in this case, none of the attributes will be discarded. The program being used, Weka, has multiple functions which allow for the automation of this process. By allowing Weka to select the most influential attributes systematically, biases may be avoided.

In addition to the data selection, the imbalance is also important to consider. One way to tackle this problem is by generating samples in the minority class. That class being the 1 instances under the stroke attribute. Another option was to use SMOTE (Synthetic Minority Oversampling Technique). This method was chosen because it is effectively a more accurate version of oversampling. SMOTE will generate synthetic data points, so that the absence of balance will be remedied. Whether this should be remedied might also be debatable. Some may argue that the raw data ratio would be an accurate representation. But considering that the difference is this extreme, SMOTE was chosen regardless.

8.2 Machine learning

As mentioned before, the accuracy of the final model is quite impressive. But there may be room for improvement. While the settings were played with in this project, not every combination was tried. Although unlikely, it is possible that a novel combination of settings may favorably change the result. The scoring matrix may also be changed, since the approach in this project was slightly rudimentary. Further statistical testing may help find a better scoring matrix.

8.3 Medical Aplicaton

With the rising importance of preventative care, this program can surely help in predicting, and therefore preventing a lot of strokes. But it is worth noting that only using this model is most likely not enough. While the accuracy and error rate are respectively high and low, the seriousness of the condition warrants further examination. Individual diagnosis from doctors would most likely always still be a requisite, for good reason. But the program may help as reminder or warning sign to the people it classifies for. It might push those people to find further help with a doctor.

9 References

10 Appendices