ST310 Machine Learning Project:

Understanding and Predicting Stroke Occurrences in Imbalanced Data

Supervised by: Professor Joshua Loftus
Candidate Numbers: 22220, 17915, 16923

Due: 7th May 2021
Word Count: ~5160 (aim: 5000)
Note 1: Word Count does not include Titles, Subtitles, Tables, References or Graphs.

Note 2: R Markdown document contains our code as well as explanations, but is not repeatable due to a lack of set.seed() in certain areas due to the fact this script was a conglomeration of several different pieces of code. Therefore, results may differ from what is stated in this report.

Table of Contents

1. Introduction

1.1 Abstract

The World Health Organization (WHO) defines a stroke as "rapidly developing signs of disturbance to cerebral function, with no apparent cause other than of vascular (blood system) origin"[1], usually caused by the disruption of blood to the brain. This may be due to a blood clot (ischaemic stroke - 85% of strokes) or rupture of blood vessels (haemorrhagic stroke)[2]. Evidence suggests that "better prevention and management of strokes may improve long-term survival rates"[3] and thus a key aspect of our report is the understanding and prediction of strokes, which we hope will aid hospitals and patients prevent as well as survive strokes.

Initially, we would like to provide consolidatory evidence on the potential risk factors for strokes which can both be advertised to generate awareness of the risk factors researched as well as easily interpreted by the wider public (better prevention). We would then seek to provide a more predictively focused machine learning frameworks that can allow hospitals as well as interested members of the public to accurately pre-empt this life-threatening condition based on given characteristics, and thus put into place the necessary infrastructure to make sure the risk of fatality is minimized (better management). Stroke research reports tend to have largely imbalanced data, with most patients recorded likely to have not had a stroke before. Therefore, this report will also evaluate the differences between different methods used to deal with an imbalanced dependent variable, and thus how future reports on imbalanced medical conditions can be integrated with machine learning frameworks to produce the most informative models possible.

This research is particularly important because according to the WHO Global Health Estimates (GHE) 2016[4] strokes are the second leading cause of death (after ischemic heart disease) and the third leading cause of disability worldwide. As per the WHO Atlas of Heart Disease and Stroke[5] ,15 million people suffer a stroke annually. Of these, 5 million instances are fatal with a further 5 million left permanently disabled. The above journal also indicates that "non-fatal instances of stroke were associated with a 5-fold increase in risk of death for up to a year afterwards". This indicates that strokes are not only common, but also fatal, with the prevention and prediction of this condition playing a key role in the mortality of at-risk patients in particular.

The potential risk factors for Strokes have been well-documented over several pieces of research. A key preventative measure for strokes according to the Centres for Disease Control and Prevention are through healthy lifestyle changes, with up to 80% of strokes possibly prevented through controlling factors such as cigarette smoking, glucose level as well as hypertension[6]. Our more interpretable models seek to provide consolidatory evidence for these risk factors, with our more predictively focused models using other factors such as Age and Body Mass Index (BMI) to accurately guess stroke occurrences, allowing at-risk patients to suitably change their lifestyle characteristics accordingly. Our report will also focus on the wider issue of dealing with imbalanced data in medical research, a

[1] https://www.ahajournals.org/doi/full/10.1161/STR.0b013e318296aeca
[2] https://www.ncbi.nlm.nih.gov/books/NBK535369/
[3] https://www.ahajournals.org/doi/full/10.1161/hs0901.094253
[4] Global Health Estimates 2016: Deaths by Cause, Age, Sex, by Country and by Region, 2000-2016. Geneva, World Health Organization; 2018. https://www.who.int/healthinfo/global_burden_disease/estimates/en/
[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5321635/
[6] https://www.betterhealth.vic.gov.au/health/ConditionsAndTreatments/stroke-risk-factors-and-prevention

common occurrence and one which is not especially well documented in prediction contexts especially.

1.2 Previous studies related to the topic

A number of previous papers have attempted to predict strokes. Our research can be tightly interlinked and motivated by Wu & Fang (2020)[7], where they focused on stroke prediction amongst an (imbalanced) older Chinese demographic between 2012 and 2014 using logistic regressions, support vector machines and random forests. They found that all methods performed poorly when presented with imbalanced training data, however, when data-balancing techniques were applied such as Synthetic Minority Oversampling Technique (SMOTE), that machine learning algorithms in combination with these methods were effective tools for stroke prediction.

Other papers such as Singh et al (2020)[8] or Lin et al (2020)[9] use other methods such as Artificial Neural Networks (ANN) for classification, as well as using regional directory-based datasets from India and Taiwan respectively. Key things to note in this research are that most prediction errors come from the most severe stroke patients. As a result, we plan on differentiating our research using different (oversampling) data-balancing algorithms from the ones stated above to tackle this problem, as well as an extra machine learning algorithm not included in the above research. Other papers add to the repertoire of algorithms used to accurately predict strokes, with Heo et al (2019)[10] as well as Fan & Wu (2019)[11] amongst others using Deep Neural Networks (DNN) in order to predict stroke outcomes, particularly in large, imbalanced data. Our dataset is not as extensive; therefore, we will avoid using DNN and instead focus on using methods based on Tree and Gradient Descent algorithms, with Hung et al (2020)[12] performing research that suggests these methods are favourable when using data that isn't 800,000 observations deep.

It should be noted that many of these above papers are area specific, with the most "general" area we could find by KGM Moons (2002)[13] where the dataset was based on a European cohort from Cardiff, Kuopio and Rotterdam, and concludes that: (using multivariate logistic regression) age, hypertension, smoking and diabetes are linked with stroke occurrences. We seek to verify the above claims using our data, as well as other machine learning methods. Other non-region-based papers such as Letham et al (2015)[14] takes a Bayesian approach to stroke prediction, concluding that the probability one suffers from a stroke is due to cholesterol levels, smoking status and blood pressure amongst others. The importance of region-based models is reaffirmed by the stroke rate in different countries - In 2016, 2.1 strokes occurred in the UK per 1000 people, whereas in China 4.1 strokes occurred per 1000 people[15], therefore, without knowing the exact origin of the dataset, we refrain from making conclusions about the generalisability of our findings.

Our report seeks to add onto the above, by using several machine learning methods to both verify the risk factors associated with stroke occurrences as well as accurately predict them. We introduce an

[7] https://www.mdpi.com/1660-4601/17/6/1828/htm
[8] https://link.springer.com/chapter/10.1007/978-981-15-4018-9_9
[9] https://www.sciencedirect.com/science/article/abs/pii/S0169260719314361
[10] https://www.ahajournals.org/doi/full/10.1161/STROKEAHA.118.024293
[11] https://www.sciencedirect.com/science/article/pii/S0933365719302295
[12] https://ieeexplore.ieee.org/abstract/document/8037515
[13] https://pubmed.ncbi.nlm.nih.gov/11815642/
[14] https://arxiv.org/abs/1511.01644
[15] https://www.who.int/healthinfo/statistics/bod_cerebrovasculardiseasestroke.pdf

analysis of methods used in the treatment of imbalanced data (specifically focusing on Undersampling vs Oversampling), using different Oversampling algorithms based on research by Lin et al (2020) to better represent the severe stroke cases in our final training data, as well as introducing Extreme Gradient Boosting, a method which is relatively new but remains unused in a stroke prediction study, and will thus give our research an aspect which is innovative and unseen as well as informative and consolidatory.

2. Data Description

The dataset used for the analysis is a 'Stroke Prediction Dataset' from Kaggle. It has the 12 variables listed below, from 5110 individuals. It should be noted that the source of this dataset is confidential. So, as mentioned in the section on previous studies, we refrain from generalising the results and drawing new inferences, and more so consolidating studies that already exist, as well as providing a resource for our experiences on using the techniques for dealing with imbalanced data as well as some novel machine learning algorithms.

Variable	Type	Values
Stroke	Binary Categorical	0, 1
Gender	Categorical	'Male', 'Female', 'Other'
Age (Years)	Numeric	$0.08 \leq a \leq 82.00$
Hypertension	Binary Categorical	0, 1
Heart Disease	Binary Categorical	0, 1
Ever Married	Binary Categorical	'Yes', 'No'
Work Type	Categorical	'children', 'Govt_job', 'Never_worked', 'Private', 'Self-employed'
Residence Type	Binary Categorical	'Rural', 'Urban'
Average Glucose Levels	Numeric	$55.12 \leq a \leq 271.24$
BMI	Numeric	$10.30 \leq b \leq 97.60$, NA: 201
Smoking Status	Categorical	'Formerly smoked', 'never smoked', 'smokes', 'unknown'

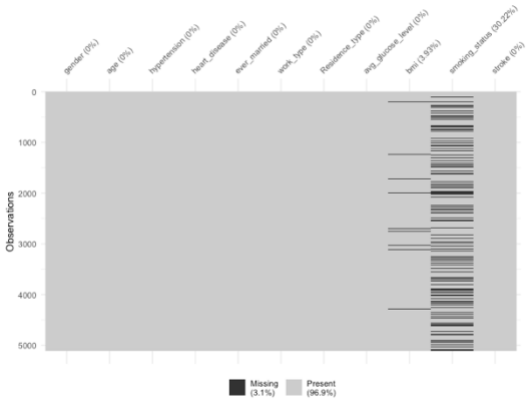
Data Cleaning

Firstly, wrongly coded variables were changed. These were generally binary categorical variables pertaining 0 and 1 which were treated as numeric values, to which we then turn them into categorical variables.

Variable	Type Before	Type After
Stroke	Numeric	Binary Categorical
Hypertension	Numeric	Binary Categorical
Heart Disease	Numeric	Binary Categorical
BMI	Categorical	Numeric

Due to the dataset containing many values, it was appropriate to remove certain points. A single point was removed to its gender being 'Other', this point was the only example of this outcome, so it was removed. For the BMI predictor, 201 values were missing (4%). Of these values, 40 of which have had a stroke. Therefore, the proportion of patients with a missing BMI who suffered a stroke is far higher than the proportion in the rest of the data, roughly 20% versus 6% respectively. Due to this higher proportion, it seems that the BMI variable could be missing in conjunction with the stroke variable. We hypothesise that either, participants withheld the information for their BMI who may have been particularly self-conscious or that the patient could not give consent, with many sufferers of stroke often knocked unconscious or into coma. Given that 201 observations had incomplete BMI data, we felt it was easiest to remove these observations rather than trying to salvage this data.

There was only one other predictor which appeared to have missing values, which was the smoking predictor, with 1684 values were missing. From these values, 69 (4%) have had a stroke, which is the proportion that we would expect from missing values. Because these 1684 values refer to roughly 30% of the dataset, including over 60 instances of strokes occurring in a stroke-limited dataset, it was far harder for us to remove these observations. With smoking status being a common risk factor, it was also difficult for us to just remove this as a predictor as well. Imputation was a method that we tried in this scenario, however, due to the categorical nature of smoking status, it was difficult to do so in a way which was unbiased. Notably, every value that was missing from BMI, was also missing from smoking status. As a result, we simply removed the observations that had missing smoking statuses. This left us with a total of 3426 observations and 10 predictors, not including our dependent variable stroke.



Strokes	0: Hasn't had a stroke	1: Has had a stroke
Observations before cleaning	4861	249
Observations after cleaning	3246	180

3. EDA

3.1 Data Preparation & Transformations

Given that our report is focused on both the understanding and prediction of strokes, it makes intuitive sense to examine the distribution of patients who have had a stroke. This is particularly important in medical studies for example, where the prominence of these types of conditions can be rare, and thus affect the predictive capability of future models. This is shown in the Table below:

Binary Outcome of "Stroke"	0 (No Stroke)	1 (Stroke)
Count	3246	180

As shown above, the frequency of people with "stroke" is small in comparison to its counterpart "no stroke", with approximately 94.75% of patients in our dataset as part of the non-stroke category. This data imbalance often leads to our models tending towards the dummy model, where we always predict the majority class, which in this case, nobody ever has a stroke. This leads to an "accuracy paradox" where a 94.75% classification accuracy sounds great, but only predicting well for one class. Especially in this case, where if we wanted to implement preventative measures for strokes, our models would essentially be useless as our models would never predict the minority (stroke) class. Using the same idea, a 95% accuracy model may not seem like anything to write home about, however the capability of accurately predicting the scenarios where a dummy model would provide a false negative is crucial in medical research. We therefore look towards minimising the false negative response as much as possible.

Using the data in its current form with 180 stroke occurrences and 3246 non-stroke patients, in exploratory models verified what Wu & Fang (2020) showed. Machine Learning on Imbalanced datasets is far from optimal, with accuracies as low as 70% (on an unseen testing data) in our exploratory logistic regression, far lower than if we were to just predict using the dummy model. Therefore, if we wanted to accurately predict or pre-empt stroke occurrences, leaving the dataset in its current form would not allow us to do this. We consider the methods below.

3.2 Balancing the Imbalance

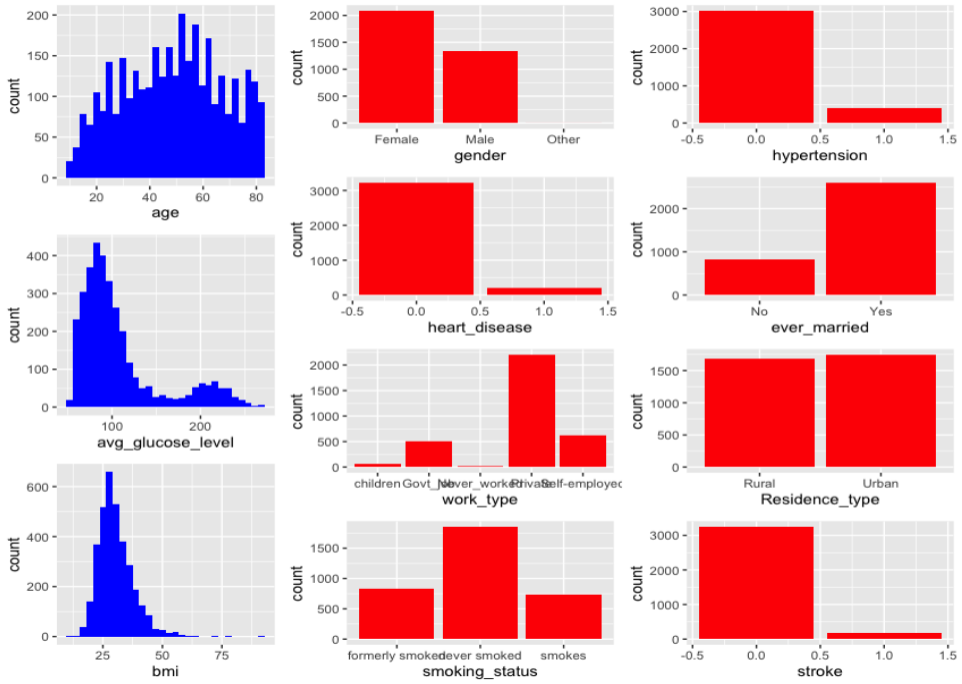
It is crucial we rectify the imbalance in the data, and there are several options for doing so. Two options available are to rework the sampling of the dataset itself.
Oversampling is a method where we "up-sample the minority class", in our case, duplicating stroke cases until we have an even 3246 cases of non-stroke and stroke cases respectively (6492 observations total). Put simply, this is where we supplement the training dataset with duplicate instances of some of the stroke (minority) outcomes. The algorithm we utilise in our code is Majority Weighted Minority Oversampling Technique (MWMOTE) where the "hard-to-learn" minority examples are identified and then given a larger weight. It should be noted that there are several algorithms for oversampling, and we decided on this algorithm based on the feedback from Lin et al (2020), where most of the predictive errors came from edge case (severe) stroke occurrences. This algorithm is therefore particularly useful in our scenario, where we assign a larger weight to edge-case scenarios and thus the machine learning algorithm has the potential to learn more than other algorithms such as SMOTE which was used in the aforementioned study. However, our initial thought

the number of observations is not an issue for us in particular and it should be noted our approach in the data cleaning aspect of this dataset has been pragmatic, where we have simply removed data that is not up to our standard, rather than try to salvage it via imputation. Thus, our initial thought was to remain pragmatic and keep our data as representative as possible, rather than apply MWMOTE, have to change all of our factors to numeric ones and then deal with a more difficult interpretation for most of our models, with training data on categorical variables using decimal points particularly concerning.

Initially, we preferred the other sampling technique, known as Undersampling where we "down-sample the majority class". This happens when we take a random sample of the non-stroke (minority) class to match the number of stroke instances in our dataset, so in this case, we would scale down our non-stroke occurrences to 180 (by taking a random sample), in order to match up with the stroke occurrences, also at 180, leaving us with 360 observations total. The negative side of this is that by reducing our observations overall, our predictive capability becomes impaired as even the non-stroke occurrences that we have decided to remove harboured some useful information about the scenarios where strokes do not occur. Despite this, we thought our results will be more representative of the actual predictive problems posed by real-life patients by underestimating standard errors.

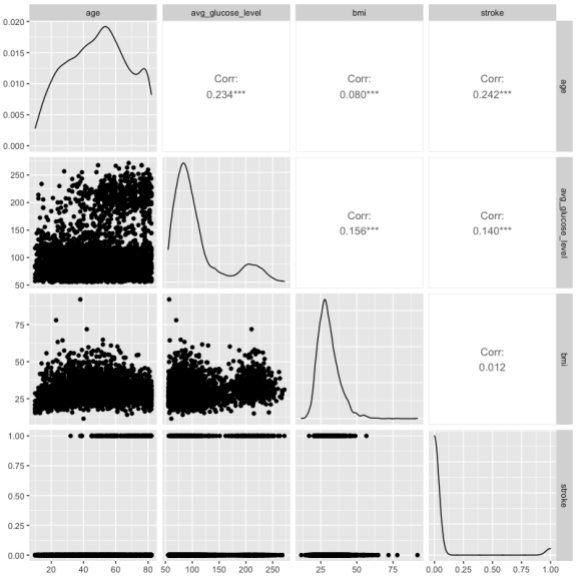
In order to confirm our initial beliefs, we decided to trial both undersampling and oversampling for our more predictively capable models, in this case, random forests and extreme gradient boosting. It became immediately obvious that the performance of these algorithms was severely inhibited by the lack of data, with models using training data that had been undersampled achieving a hit rate of roughly 70% on our testing set. Far below our target of 94.7% that is achieved by a dummy model. When using oversampled training data on the other hand, our resulting accuracy became far more reasonable, with our predictively focused models outperforming the dummy model consistently, and thus providing useful information that could be used for implementation by hospitals, whereas we would have no such capability for implementation using undersampled data. Therefore, we opted to use the oversampled data throughout our methods, noting that the generated data is a disadvantage and may not be entirely representative of the actual population.

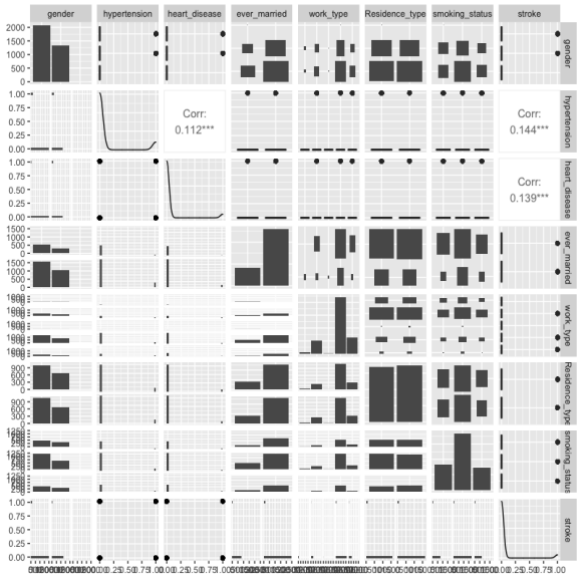
3.3 Univariate Analysis



The Figures above represent the univariate distributions for our continuous predictors (blue) and categorical predictors (red). If we consider the continuous predictors first, the histograms for bmi and avg_glucose_level demonstrate a clear positive skew, although in the case of avg_glucose_level a bimodal aspect to the distribution is also evident. Therefore, we could feasibly consider log transformations on these predictors later in the analysis. The age predictor seems to be uniformly distributed across all ages with more patients tending to be middle-aged. On the other hand, if we consider our categorical predictors, many of them have a single dominant (mis-matched) class. Variables such as stroke (our dependent variable) but also hypertension, heart_disease and work_type notwithstanding are our main examples of having a dominant category.

3.4 Bivariate Analysis





Finally, we decided to investigate the pairwise relationship between each of our variables in the form of a pairwise plotting diagram, as well as specifying the stated pairwise correlations. Of note, many of our independent variables seemed to have a significant relationship to our dependent variable stroke. Examples of these are age, hypertension, avg_glucose_level, and heart_disease with correlations of 0.242, 0.144, 0.140 and 0.139 respectively with our dependent variable. What we can infer from this list of (underwhelming) correlations with the stroke variable is that there is no single “silver bullet” in our dataset which correlates exactly with strokes. Instead, we hypothesise that an extensive combination between our features will allow us to build an accurate classification model. It should also be noted that the lack of large pairwise correlations between the independent variables suggests that multicollinearity, the issue where independent variable can be highly correlated with one another (undermining the significance of these variables), is not going to be an issue. We can verify this with a Variance Inflation Factor (VIF) test for one of our models.

4. Our Methods: 'Interpretable Models'

4.1 Dummy Model

The data set has a high proportion of outcomes that don't have a stroke. For this reason, we could predict every outcome in the raw data to not have a stroke.

	Actual 0	Actual 1
Predicted 0	3246	180
Predicted 1	0	0

This would give 94.7% accuracy. In this case, all of the misclassified observations would be false negatives. False negatives provide little information to us, whereas false positives may be indicative of risk of stroke. We believe this model is particularly important to mention due to the "accuracy paradox" we mentioned earlier in the report. Just because we are using a machine learning technique does not mean it will be good at predicting outcomes no matter how complicated, and therefore we must be aware of the context of our problem, and thus the importance of any of our findings.

It should also be noted that for this dummy model, the accuracy reduces to exactly 50% when we apply both the Undersampling and Oversampling procedures, due to the fact that both datasets will have as many stroke cases as they do non-stroke cases. Despite this, it is unreasonable to assume that when introduced to a completely new set of data (out-distribution generalisation) that they will still be evenly distributed, especially given the context of this machine learning problem, with the imbalance issue a well-documented and common one. Therefore, our threshold for a good predictive model will remain at 94.7% accuracy, where one may be tempted to reduce this to 50% in accordance with our imbalanced data changes.

4.2 Logistic Regression: Baseline Model

Logistic Regressions are a classification algorithm, and it is used to calculate the probability of a binary event occurring based on a set of predictor variables. In our case, logistic regression makes an ideal baseline model, with appropriate interpretational features and predictive capacity.

The sigmoid function maps predicted values to probabilities (between 0 and 1) and this is what gives a logistic regression its famous S shape. After obtaining a probability score between 0 and 1, we select a threshold value or a decision boundary above which we classify values into class 1 and below which we classify values into class 0. In our implementation, we did this by using a pre-set grid of values and testing each value as the parameter, then selecting the best of these models. Cross validation was not usable as false negatives are worse than false positives, thus it's not necessarily just the error we want to minimize and instead a mixture of the accuracy and false negatives.

A logistic regression model is given by the following equation:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_m x_m$$

Where pi indicates the probability of an event, betais are the regression coefficients and xis are the predictor variables.

	Actual 0	Actual 1
Predicted 0	580	47
Predicted 1	355	936

The total misclassification rate was 20%.
The model is as follows:

```
Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -8.2111915  0.4315030 -19.029  < 2e-16 ***
gender          -0.1903558  0.0897963  -2.120  0.03402 *
age              0.0935111  0.0033173  28.189  < 2e-16 ***
hypertension     0.5992638  0.1088342   5.506 3.67e-08 ***
heart_disease    0.7698440  0.1366198   5.635 1.75e-08 ***
ever_married    -0.3236938  0.1485721  -2.179  0.02935 *
work_type        0.1309077  0.0471366   2.777  0.00548 **
avg_glucose_level  0.0066687  0.0007319   9.111  < 2e-16 ***
smoking_status   0.1918944  0.0610377   3.144  0.00167 **
```

This means that the variable with positive coefficients increases the likelihood that an individual has a stroke and those with a negative coefficient decreases the likelihood that an individual has a stroke. Hypertension has the coefficient 0.5992638. this impacts the odds by $e^{0.5992638} = 1.82$, meaning that someone who suffers from hypertension is 182% more likely to suffer from a stroke than an identical individual who doesn't, all things equal. Age has a coefficient of 0.0935111, this impacts the odds by $e^{0.0935111} = 1.098$. This means that for two identical individuals, with one a year older than the other, the expectation that the older individual has a stroke is 1.098 higher, all things even. The other parameters may be interpreted in the same way. Thus, if we interpret these findings, Age, Hypertension, Heart Disease and Average Glucose Level are important risk factors in the prediction of strokes, with all of them highly significant. These reports therefore consolidate the previous studies that have researched risk factors in stroke occurrences. Unusually, smoking status seemed to have a significant effect, but not as highly as we would expect, given that it's one of the first suggested lifestyle changes in order to lessen the probability of facing a stroke. A meta-analysis of stroke studies by Pan et al (2019)[16] shows that the risk of stroke increases by 12% for each increment of 5 cigarettes

[16] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6708836/

a day. Our data for smoking status doesn't capture the extent to which someone has smoked as well as the studies used in this meta-analysis, and we hypothesise that as a result, some people who formerly smoked or still smoke may be diluting the power of this category, with the specific amount these people smoke an unknown factor.

4.3 Lasso Regression: Relatively Interpretable Model

Lasso stands for least absolute shrinkage and selection operator.

Lasso regression performs L1 regularisation. This means that it adds a penalty equal to the absolute value of the magnitude of the coefficients (L1 norm). As a result, the coefficients of less contributive predictors are shrunk to 0. Therefore, lasso models can be used to minimise the effect of highly correlated variables.

Lambda (**λ**), is the tuning parameter and this controls the L1 norm. **λ** has an inverse relationship with the L1 norm. 10 fold cross validation using the misclassification rate was performed to determine the optimal value for **λ**.

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

The model obtained using oversampling included 7 predictors. Namely, gender, age, hypertension, heart_disease, work_type, avg_glucose_level and smoking_status. In a lasso regression, the faster a coefficient is shrunk to 0, the less significant the predictor is.

	s0
(Intercept)	-7.586969947
gender	-0.006107622
age	0.085735590
hypertension	0.507606465
heart_disease	0.593544989
ever_married	.
work_type	0.057024201
Residence_type	.
avg_glucose_level	0.005307559
bmi	.
smoking_status	0.072539170

The coefficients are interpreted similarly to a linear regression. From the model there was a positive correlation among age and stroke. The final model had a misclassification rate of 17.8%.

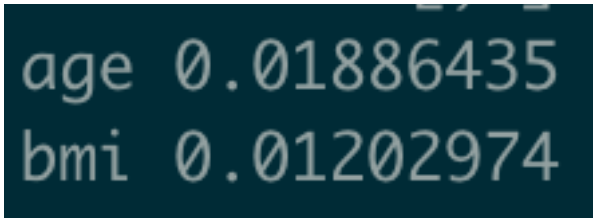
4.4 Gradient Descent

In order to implement the gradient descent, a linear regression was used. To keep things simple, we opted to go with just two significant continuous predictors: Age and BMI. This means that the loss function is the sum of squared residuals. In order to get our estimate, we must get the loss function with respect to the intercepts and slopes and differentiate it for each. From here the algorithm was put into motion with our step size as 0.000001. From here we ran the algorithm till it converged. This was measured by the step size being smaller than 0.000001. In order to increase accuracy, the step size decreases as the values converge to the value.

	Actual 0	Actual 1
Predicted 0	264	3
Predicted 1	671	1009

The classification rate is 65.4.%. This is a model that predicts poorly, however, it has been included to demonstrate the method of gradient descent. It successfully predicts over half the values which suggest it has some predictive power.

Our coefficients are as follows:



The coefficient for age suggests that if we had two identical individuals, with one a year older than the other, the expected probability that the older person has a stroke is 0.01886 higher. The coefficient for BMI suggests that if we had two identical individuals, with one with a BMI one larger than the other, the expected probability that the individual with the highest bmi has a stroke is 0.01203 higher. The coefficients may be used to predict for the training set by using matrix multiplication. When this is done, it predicts with 65.4% accuracy. While this value is low, it is a very simple model that seems to have some predictive power.

5. Our Methods: High Accuracy Models

5.1 Kernel: Higher dimensional model

The Kernel method allows for the lower dimensional data to be treated as higher dimensional. A good support vector classifier could not be found due to the nature of messy, medical data. The Radial Basis Function (RBF) behaves like the weighted nearest neighbours' method, where the outcome of similar observations has a larger effect on the classification than less similar observations (those further away). The influence that an observation has on another is a function of the distance squared, $e^{-\gamma(a-b)^2}$ where a and b are two points. This is the relationship between two points in infinite dimensions. Our parameter, ɣ is the coefficient of the squared distance, meaning that it scales the

influence, the larger ɣ, the less influence that two observations have on each other. The data is not actually transformed, the dot product is calculated which gives the high-dimensional relationship. The RBF is similar to the Polynomial kernel. If you expand out the formula, and expand the term e^{ab} using a taylor expansion, you have an infinite sum where r = 0 and d = 0, 1, 2, …∞.

The parameters used were those indicated to be significant by previous models. Gamma would usually be worked out using cross-validation. In this instance this wouldn't be appropriate as not all errors have the same cost. A false negative is much worse than a false positive, as the false positive may be indicative of an issue. Thus, before performing the kernel method, I produced a grid of values for values of gamma. I implemented each of these and selected the model that appeared the best.

The final model presented has an accuracy of 89.6%. From this, the false negative rate was 5.4%.

	Actual 0	Actual 1
Predicted 0	838	105
Predicted 1	97	907

Due to false negatives being worse than false positives, cross validation was not used. Instead, a grid of values was used, and each value of gamma had a model calculated around it. Then the function 'L(g) = Number of false negatives - Accuracy', was minimized. This function works well as accuracy is between 0 and 1, meaning that it finds the values with the lowest number of false negatives, then chooses the one of those with the highest accuracy.

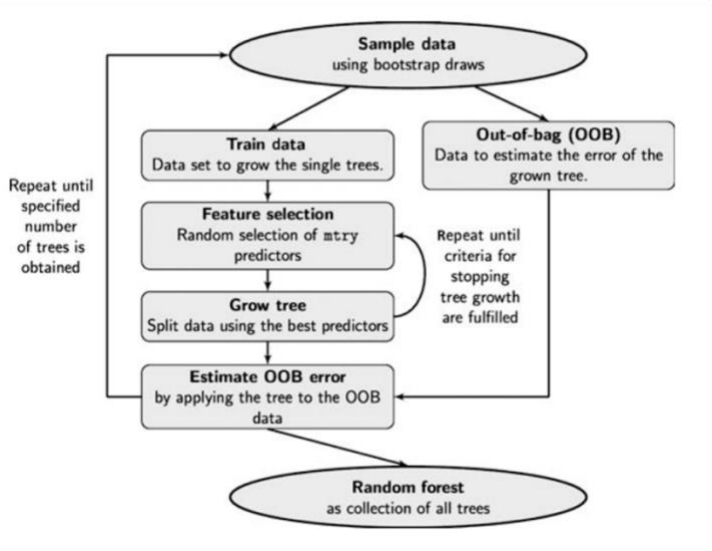
5.2 Random Forests: Predictive Accuracy

The Random Forest algorithm is an extension of the bagging method for improving decision trees. Decision trees often suffer from overfitting and thus high variance (also notably low bias) when there are often too many decision nodes (determined by tree depth) and therefore conditions in order to reach a classification decision. This usually results in poor out of sample (testing) results. We seek to reduce variance by introducing bagging; a method which improves the decision trees by fitting several decision trees using different data samples. We generate these data samples using the bootstrap algorithm, where each observation in our main training data is sampled with replacement, and then fit a separate prediction model to each of them. We average the resulting predictive models in order to obtain a single low-variance model.

The issue with these bagged decision trees is that they can often be quite similar to each other. Consider a scenario where there is one particularly strong predictor in any given dataset alongside other reasonably informative predictors. In the bagged decision trees, the decision tree algorithm will always choose the strong predictor in the top split, leading to all of the bagged trees looking similar, and thus the overall reduction in variance being lower since we understand that averaging many highly correlated quantities does not lead to as large of a reduction in variance as averaging many uncorrelated quantities. We attempt to address this through a slight tweak to the bagged trees algorithm which decorrelates the trees.

This time, when building these decision trees using bootstrapped samples, we only consider a random sample of the predictors (from the full predictor set) each time a split in the decision tree is

considered which is often referred to as feature selection (or feature randomness). The result is that more of the reasonably powerful predictors are used, decorrelating the bagged decision trees, with the resulting averaged decision tree less variant and hence more reliable. The full random forest algorithm can be summarized using the diagram provided in Boulesteix et al. (2012)[17].



Before we start training, we must select the three main hyperparameters for Random forest algorithms. These parameters are node size (min.node.size), the number of trees (B), and the number of features sampled (mtry). Of note, if we increase the number of trees (and thus bootstrapped samples), random forests will not overfit. Therefore, we select a number of trees such that the error rate has settled. In our implementation, we decide that our minimum node size should be 5 so that our resulting decision trees are not underfit, and that we use several values for our number of features sampled and fit the algorithm using each. We then select the value for mtry such that the accuracy is highest from a vector of common values. Other things to note is that our splitting rule was defined as the gini impurity (standard for classification problems). This selection of hyperparameters should give us a competitive solution for a random forest's algorithm, with close to the maximum accuracy this method can achieve, shown below:

```
> confusionMatrix(randomforest.os_predictio
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 944  46
         1  22 936

               Accuracy : 0.9651
                 95% CI : (0.956, 0.9728)
    No Information Rate : 0.5041
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9302
```

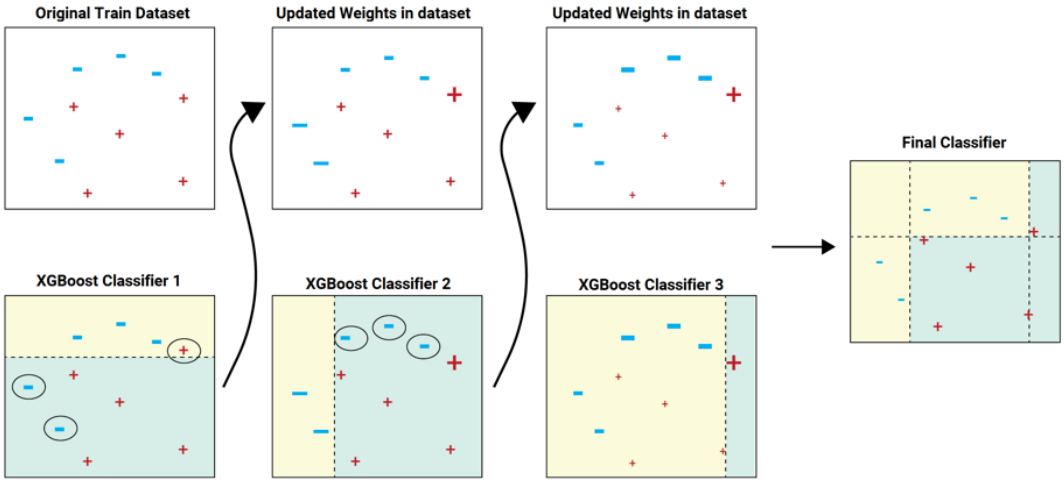
	Actual 0	Actual 1
Predicted 0	944	46
Predicted 1	22	936

At 96.51% accuracy, random forests represent our highest accuracy so far, misclassifying 68 cases in total. Despite this high accuracy, random forests represent the most complex method that we have presented so far, making it highly difficult to ascertain why any given prediction has been given. This is further intensified by the oversampling method we decided to use, making each individual decision tree problematic to follow.

[17] https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1072

5.3 Extreme Gradient Boosting: (Extreme) Predictive Accuracy

Developed in 2016 by University of Washington researchers[18], Extreme Gradient Boosting (otherwise known as XGBoost) is based on the gradient tree boosting algorithm in ensemble learning, as it enlists many classification models, in the form of decision trees, and classifies any given observation by taking a linear combination of these trees in order to generate a cumulative model to make predictions together. These types of boosting algorithms in ensemble learning work by generating an initial structure of weak models and combining them together to make a powerful predictive framework. Gradient boosting specifically uses the gradient of the loss function (minimized by gradient descent) in order to decide which models, have the highest predictive performance and thus how to build an optimally powerful model. XGBoost's key advantage and indeed what it is famous for is its predictive accuracy, with several competition winning models[19] built using this algorithm. Its extra predictive accuracy comes from using second-order approximation of the scoring function, allowing it to calculate optimal split conditions for a decision tree. In our context, XGBoost offers our research another key advantage over others in the field of stroke prediction, where the algorithm remains scarcely used despite its ability to predict well in similar scenarios[20].



We train our XGBoost algorithm based on seven given parameters in the train function we employ in R. We set nrounds (number of boosting iterations) at 3500, max_depth (max tree depth) at 7, eta (shrinkage) at 0.01, gamma (minimum loss reduction) at 0.01, colsample_bytree (subsample ratio of columns) at 0.75, min_child_weight (minimum sum of instance weight) to 0 and subsample (subsample percentage) to 0.5. Another useful feature of XGBoost is that it has a useful cross validation function, which at each boosting iteration, returns the optimum number of trees. We set this to 5-fold cross-validation. In the interests of saving time, as well as not having learned the fundamental intricacies of how each of these parameters affect our final predictive model, we decide to opt for "standard values" for these hyperparameters.

[18] https://arxiv.org/pdf/1603.02754.pdf

[19] https://github.com/dmlc/xgboost/blob/master/demo/README.md#machine-learning-challenge-winning-solutions

[20] https://www.ahajournals.org/doi/full/10.1161/STROKEAHA.117.019440

Despite the lack of hyper-parameter tuning, the XGBoost model delivers our highest predictive accuracy out of all our models, with an accuracy of 96.71% as shown below, correctly classifying 6 more of our observations than the Random Forest algorithm, further solidifying its status as the most predictively accurate algorithm.

```
> confusionMatrix(xbg.os_pred, factor(testir
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 942  40
         1  24 942

               Accuracy : 0.9671
                 95% CI : (0.9582, 0.9746)
    No Information Rate : 0.5041
    P-Value [Acc > NIR] : < 2e-16
```

	Actual 0	Actual 1
Predicted 0	942	40
Predicted 1	24	942

It is not without disadvantage, however. In our context, we have treated it as a Blackbox algorithm with several layers in the form of hyperparameters that we have decided not to interfere with, and which make it far more difficult to interpret than even random forests. This as well as not tuning hyperparameters (leaving some performance on the table) mean that this particular algorithm is not optimized. But even as an exploratory analysis for further research in the future, it is clear to see that in our case stroke prediction using an XGBoost algorithm provides us with the most accurate classifier, and therefore the best mechanism which can be applied in real life scenarios to save the most lives.

6. Conclusion

6.1 Findings

	Using Undersampling		Using Oversampling	
	Accuracy	False negative	Accuracy	False negative
Dummy	50%	100%	50.0%	100%
Logistic	71.3%	9.7%	80.6%	4.8%
Linear model using gradient descent	58.3%	0.0%	65.4%	0.3%
Lasso	76.8%	23.5%	81.4%	14.9%
Kernel	74.0%	23.5%	89.6%	10.4%
Trees	75.9%	21.6	96.5%	4.7%
Gradient Boosting	77.7%	19.6%	96.7%	4.1%

We have made a selection of models that have predicted with varying degrees of accuracy. Unsurprisingly, the baseline predicted with the lowest accuracy. There appears to be a trade-off

between accuracy and false negative rate. However, this relationship was non-existent when it came to trees and gradient boosting. For our more interpretable models, we have found some results that cross-validate what was found in previous research in the area. For our more predictively focused models, we have achieved an accuracy with Oversampling where we would feel comfortable implementing our models in real-world environments in order to pre-empt stroke occurrences. It should be noted that the relative differences between Undersampling and Oversampling in terms of accuracy is large, with Oversampling the superior option for predictive problems.

6.2 Limitations

Our analysis has several key limitations. Firstly, oversampling was used. This technique alters the data to give a heavier weighting to minority classes by making new data points to represent them. This allowed us to use the full data set. The alternative was Undersampling which means we would lose a lot of the data points, which means some information would possibly be missing. The limitation of oversampling stems from the fact that much of the data is not real, instead it has been automatically generated.

Another limitation of our analysis is the source of the data. Despite being highly rated on Kaggle, the source of the data is listed as 'Classified'. This means that we don't know where the data was collected geographically, or why the data was collected. This could be particularly significant as it could be that the subset of the population the data is sampled from does not represent the population. Thus, using our models to predict stokes within individuals not in the data set would potentially be highly inaccurate.

6.3 Improvements

Our analysis could have been improved in a number of ways. While it was out of our control, a data set with a more balanced selection of values would have strengthened our analysis. It would have meant that we would not need to use Undersampling or Oversampling. Yet more, knowing the source of the data would allow for more confidence in the results.

The gradient descent model was included purely to demonstrate understanding, however, the model it produced was poor. If we had more time, we could have used gradient descent to determine the predictor for another model type such as ridge. Another interesting addition could have been the use of Deep Neural Networks or some forms of Bayesian Multilevel Models (implemented with Markov Chain Monte Carlo sampling) both of which have been shown in previous studies to have good predictive accuracy, but are not within the scope of this report.