University of Illinois Springfield

Stroke Prediction

CSC 532B Introduction to Machine Learning
Final Project
Spring 2021

Thomas L. Champion

Table of Contents

Abstract

The Stroke Prediction Dataset is a publicly available dataset on Kaggle that was initially posted in January 2021. This project aims to predict the occurrence of a stroke based upon the features included in the dataset. Activities involved in this process include data exploration, cleaning and preprocessing as well as model training and evaluation. Models used range from simpler models such as k-Nearest Neighbors to more complex neural networks, with each model undergoing appropriate hyper-parameter tuning and cross-validation.

Problem Definition and Project Goals

According to the World Health Organization, stroke was the second largest cause of death worldwide in 2019[11], accounting for approximately 11% of deaths. In addition to death, survivors of a stroke may experience adverse effects for years after the stroke event, including depression and physical impairments[2]. Aside from these direct impacts, there is also the financial impact of a stroke event, with the costs of post-stroke care estimated to be $4850 per patient month in the USA based upon the 2015 US Dollar[7].

It is therefore advantageous, from both and economic and social perspective, to prevent the occurrence of a stroke or at least minimize the severity of a stroke as much as possible. The ability to do so relies heavily on identifying those individuals at risk of a stroke event.

Using a dataset obtained from Kaggle, this project seeks to develop a machine learning model to identify those individuals at risk of a stroke. The dataset used consisted of 5110 observations listing details about individuals that did and did not experience a stroke. The data set contained the following variables:

Feature Name	Data Type	Description
id	Numeric	unique identification number
gender	Character	gender of individual with three values:: “Male”, “Female”, “Other”
age	Numeric	age of individual
hypertension	Numeric	Indication if the individual was diagnosed with hypertension. 0=No, 1=Yes
heart_disease	Numeric	indication if an individual has heart disease. 0=No, 1=Yes
ever_married	Character	indication if the individual has ever been married. “Yes” or “No”
work_type	Character	Indication of the individual’s category of work. Five values: “children”, ‘Govt_job”, “Never_worked”, “Private” and “Self-employed”
Residence_type	Character	Indication of general area an individual resides. Two values: “Rural” and “Urban”
avg_glucose_level	Numeric	Individual’s average blood glucose level
bmi	Character	Individual’s body mass index
smoking_status	Character	Indication if individual smokes or has ever smoked. Four values: “formerly smoked”, ‘never smoked”, “smokes”, “Unknown”
stroke	Numeric	Binary response variable indicating whether an individual experienced a stroke. 0=No, 1=Yes

Related Work

The dataset used for this project was obtained from Kaggle (http://www.kaggle.com), a website popular amongst those with an interest in data science, machine learning and related fields. The specific data set used, ‘Stroke Prediction Dataset’[9], was initially posted on January 26, 2021 and as of the completion of this project has been viewed ~171K times, downloaded ~26.1K times and shows 337 unique contributors to the project[10] as hosted on Kaggle.

In order to avoid any influence on this project the contributions and discussion on Kaggle were not reviewed until after completion of this project’s analysis and evaluation. Review of the contributions and discussions shows a wide range of activity on this dataset on the Kaggle platform, including extensive work on data visualizations[5], data cleaning and fixing an imbalanced data set.

A large number of projects focused on building predictive models emphasize a high accuracy score. As is discussed as part of the model evaluation process in this project, due to the highly imbalanced nature of the dataset accuracy was not chosen as an evaluation metric for this project. There were other contributors on Kaggle that followed a similar path, choosing to use AUC, Recall, Precision and/or F-1 Score instead of accuracy, such as the project by Sai Charan Randhi[8].

Data Exploration and Preprocessing

Basic exploration and preprocessing

The initial step was to verify that the dataset did not contain any duplicate observations based upon the unique ID included. Once this was verified, the id field was removed from the data set since it is not related to whether an individual had a stroke even or not.

The data set contains two binary features, ever_married and Residence_type that are coded as character fields. Both of these fields were converted to numeric fields as most models will prefer numeric over categorical values. For ever_married, 0='No" and 1="Yes' while for Residence_type 0="Rural" and 1="Urban".

Summary information for gender showed that there were 2897 rows for 'Female', 2011 for 'Male' and 1 for 'Other'. With a single instance of the 'Other' category it will not have any predictive value in the model as was therefore removed. The field was then updated to a numeric variable with 0='Female' and 1='Male'.

Since both work_type and smoking_status have more than two possible values they were each converted to an unordered factor, which will be processed with one-hot encoding as necessary for use in model training.

Exploration of the bmi field showed that in addition to having numerical values stored as characters the field also contains unknown values codes as 'N/A'. These 'N/A' values were converted to NA and then the field converted to a numeric field. It was then found that there were 201 records missing bmi information, which must be resolved prior to building the models.

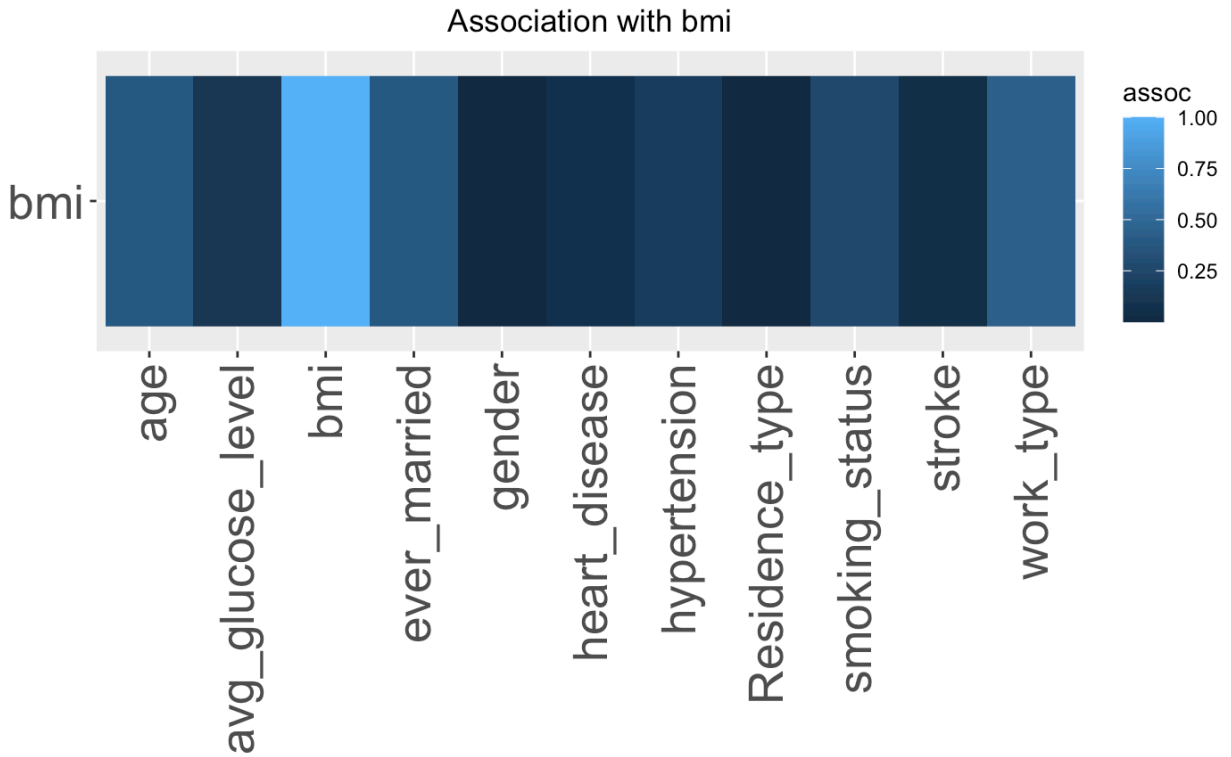
Next, the stroke response variable was explored revealing a highly imbalanced data set with 4861 rows indicating no stroke and 249 indicating stroke. As modeling an imbalanced data set is difficult, it will be necessary to include a process to balance the training data prior to use. It was also decided to update the variable to factor with two levels, "Yes" and "No", to simplify interpretability during the model evaluation phase.

bmi imputation

As previously stated, the bmi field was found to be missing 201 values, which equates to slightly less than 4% of the observations. With a relatively small and highly imbalanced data set it was decided to retain the observations and impute the missing bmi information.

Rather than using the global mean or median for the dataset, an exploration was done for any association between bmi and another predictor variable. Results showed that age had a moderate association with bmi.

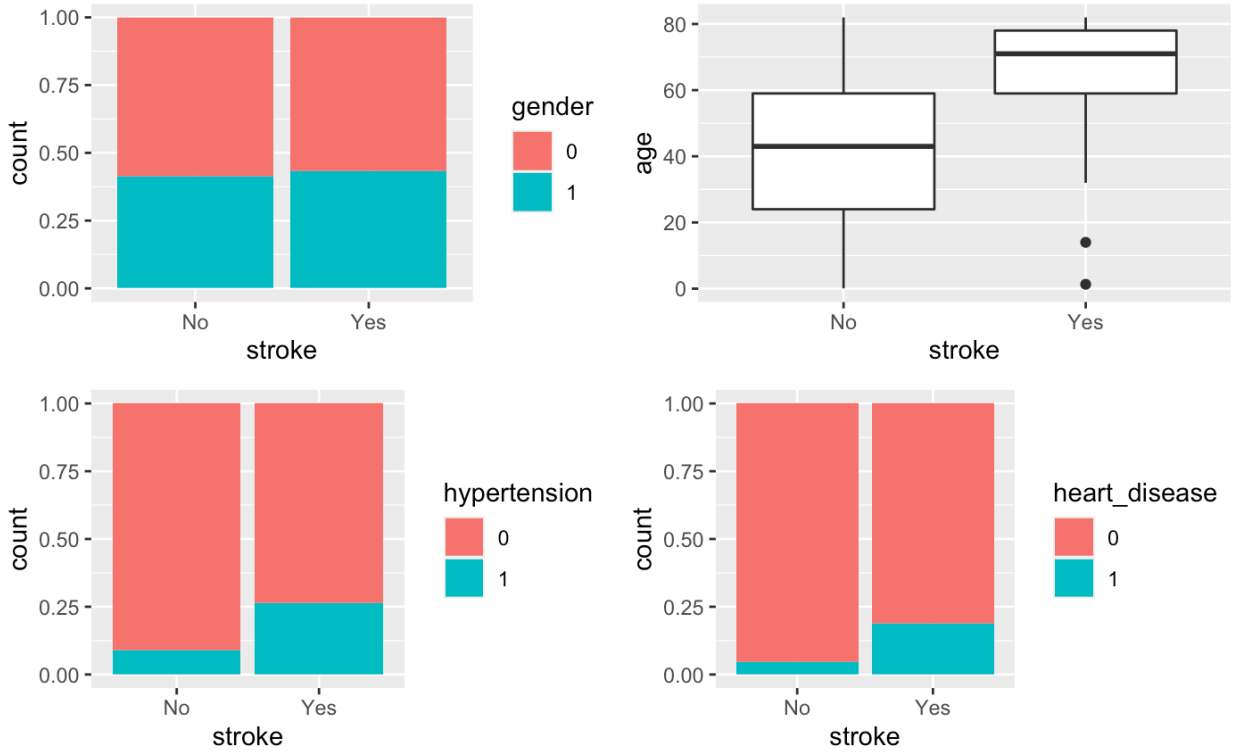
feature	association	type
gender	0.00032415	correlation
age	0.37544096	correlation
hypertension	0.167445342	correlation
heart_disease	0.066092868	correlation
ever_married	0.385616427	correlation
work_type	0.450742125	anova
Residence_type	0.000921627	correlation
avg_glucose_level	0.114633379	correlation
bmi	1	correlation
smoking_status	0.272942631	anova

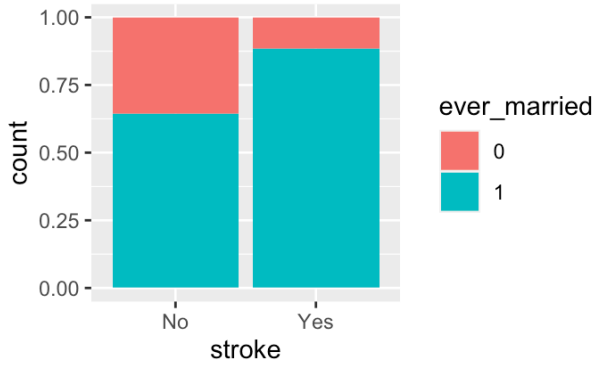


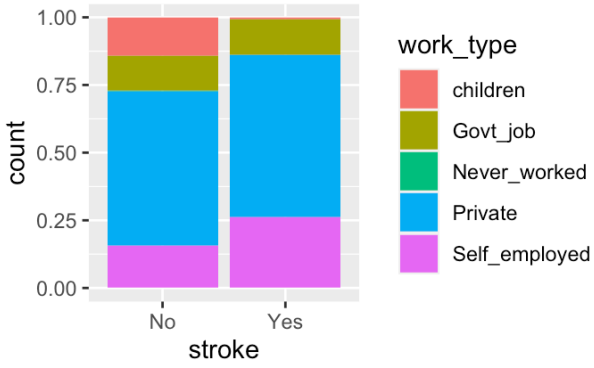
Using age as the criteria, the dataset was partitioned into 10 bins using the clustering method found in the bin function, which is part of the OneR package. The missing bmi values in each bin were then set equal to the median value of its bin.

Association Analysis

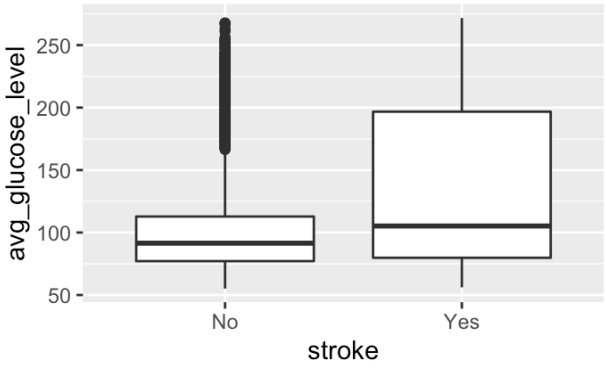
After the initial data cleaning phase significance tests and data visualization made of the relationship between stroke and the predictor variables. Significance tests were conducted using a chi-square test for the categorical variables and a two-sided t-test for the numeric variables. Visualizations were made using bar plots or box plots for categorical and numeric variable respectively. The 5 categorical variables coded as binary numerical values (ever_married, Residence_type, hypertension, heart_disease and gender) were treated as factors for these purposes.

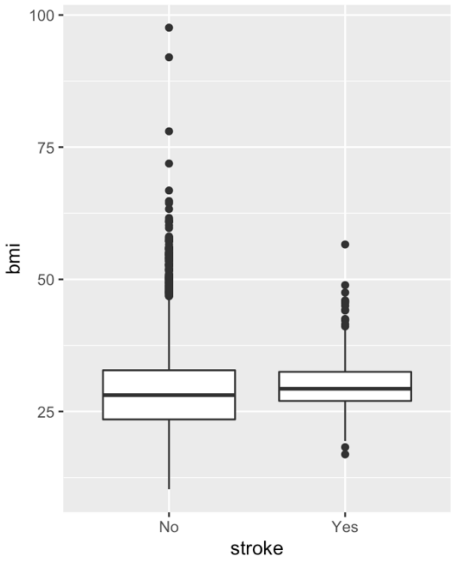


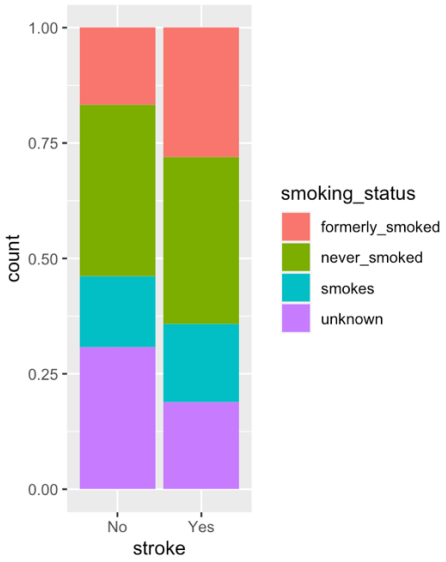

1.00
0.75
0.50
0.25
0.00
count
No
Yes
stroke
ever_married
0
1


1.00
0.75
0.50
0.25
0.00
count
No
Yes
stroke
work_type
children
Govt_job
Never_worked
Private
Self_employed


1.00
0.75
0.50
0.25
0.00
count
No
Yes
stroke
Residence_type
0
1


250
200
150
100
50
avg_glucose_level
No
Yes
stroke


100
75
50
25
bmi
No
Yes
stroke


1.00
0.75
0.50
0.25
0.00
count
No
Yes
stroke
smoking_status
formerly_smoked
never_smoked
smokes
unknown

Chi-Sq Tests

chisq_labels	chisq_values
smoking_status	2.01E-06
gender	0.56
hypertension	1.69E-19
heart_disease	2.12E-21
ever_married	1.69E-14
work_type	5.41E-10
residence_type	0.30

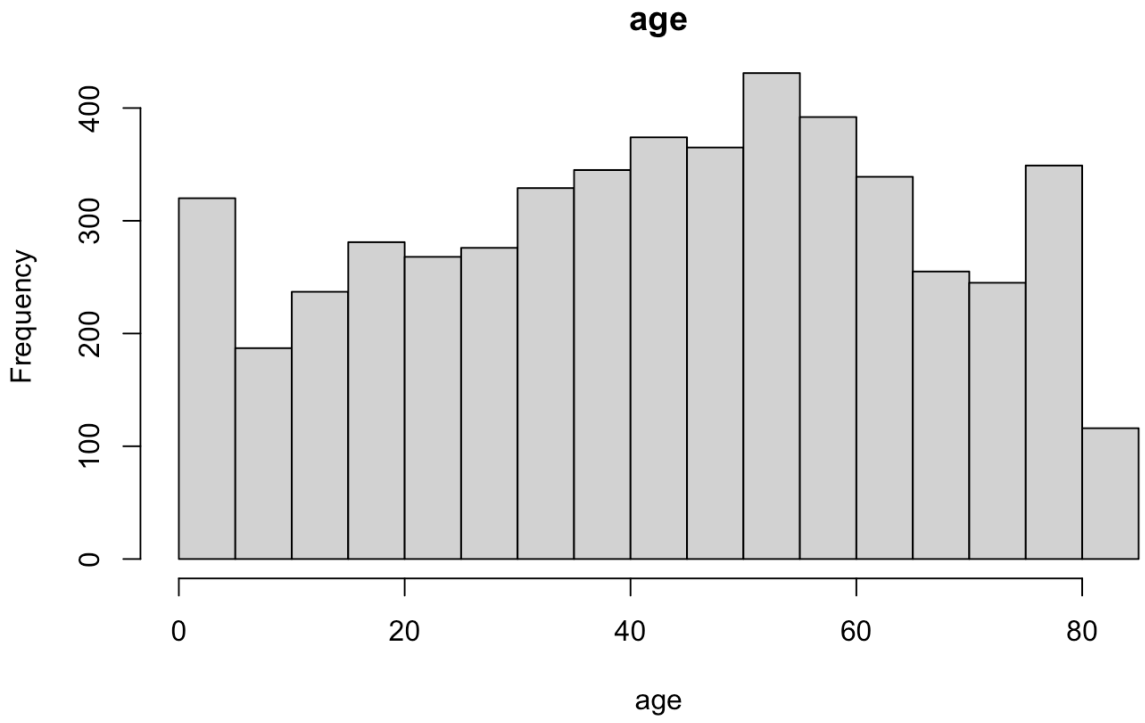
T-Tests

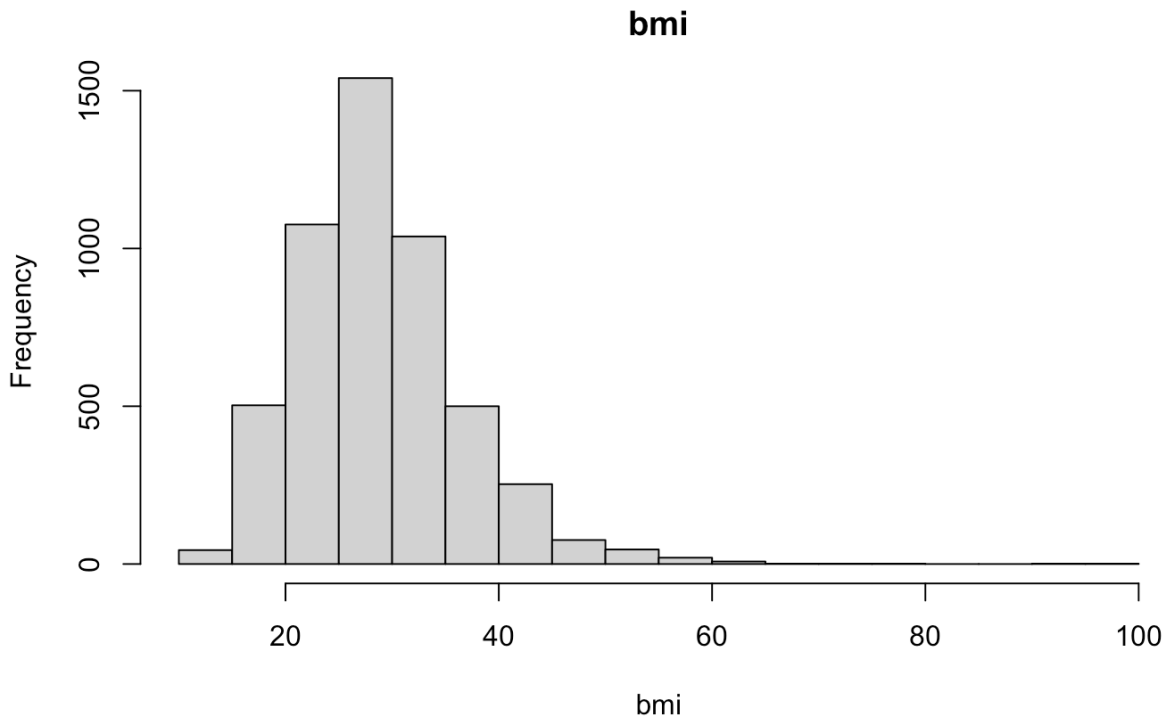
ttest_labels	ttest_values
age	2.18E-95
glucose	2.37E-11
bmi	0.0003064

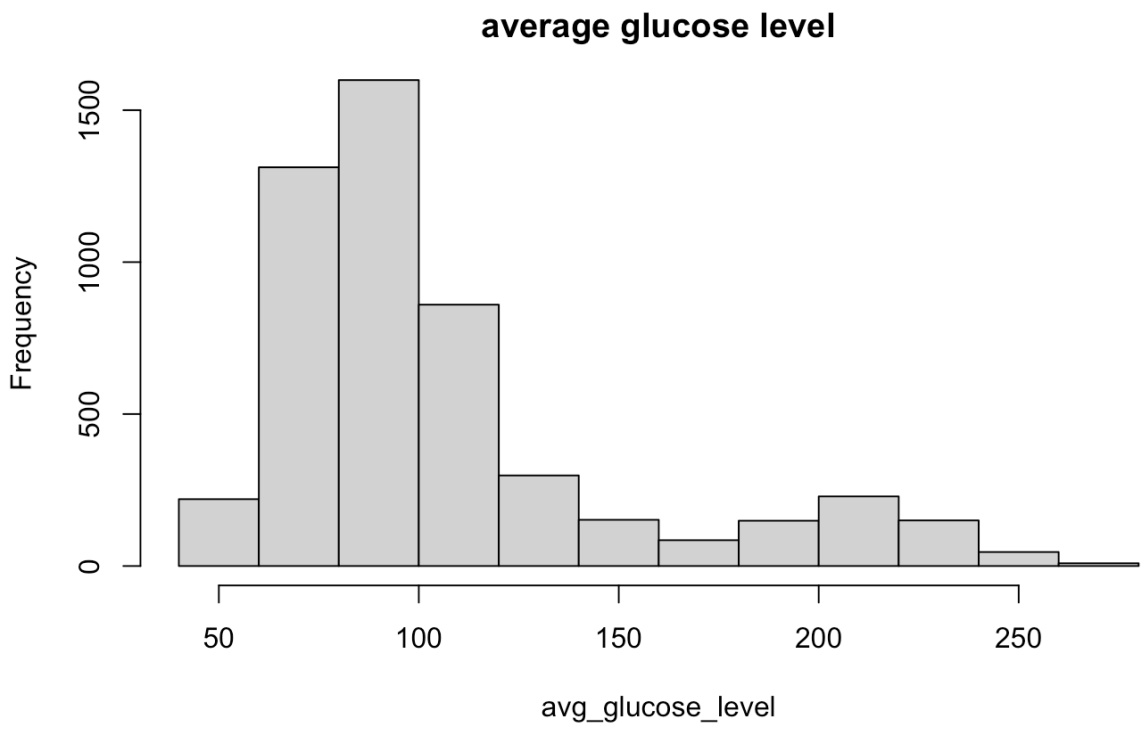
Both visualization and statistical analysis indicate that all predictor variables except for gender and residence_type show an association with stroke. In order to avoid an overly complex model by the inclusion of unnecessary features, both gender and residence_type were removed from the data set.

Distribution of numeric variables

The three numeric variables age, bmi and avg_glucose_level were examined by the use of histograms, which revealed that both bmi and avg_glucose_level are right-skewed.


age
Frequency
0
100
200
300
400
0
20
40
60
80
age


bmi
Frequency
0
500
1000
1500
20
40
60
80
100
bmi



To minimize the impact of the skewed distributions, both avg_glucose_level and bmi were log transformed for use in model training.

Data Preparation

The data was divided into two sets, 80% for training (fullTrain) and 20% for testing (test). The training set was further divided into two subsets, 80% for the final training dataset (train) and 20% for validation (validation).

Further data preparation was done based upon the intended model usage.

Caret Package

In order to deal with the highly imbalanced data the fullTrain dataset was processed using the ROSE package to produce a more balanced dataset.

Before ROSE Processing

No	Yes
3888	200

After ROSE Processing

No	Yes
2016	2072

The fullTrain data set was then centered and scaled using the preProcess function from the caret package. The same preProcess instance was used to center and scale the test dataset.

The fullTrain and test datasets were only used for final model evaluation. The model training process, including cross-validation and hyper-parameter tuning, used the test and validation datasets, neither of which underwent balancing or scaling at this time.

Since many models require numeric inputs rather than categorical, a second copy of each data set (fullTrain, test, train, validation) was made and both the smoking_status and work_type variables were one-hot encoded using the dummyVars function from the caret package.

Keras

In order to prepare data sets for use in a neural network using keras, tensorflow and tfruns a copy of each original data set was made (nnTrain, nnValidation, nnFullTrain and nnTest).

The nnTrain data set was scaled using the scale function. The nnValidation data set was then scaled using the same column means and column standard deviations. Similarly, nnFullTrain and nnTest were scaled using column means and standard deviations from the nnFullTrain dataset.

Both nnTrain and nnFullTrain were then balanced using the ROSE package. Neither nnVal nor nnTest were balanced. After processing both nnTrain and nnFullTrain showed a more balanced distribution.

nnTrain

No	Yes
1642	1629

nnFullTrain

No	Yes
2021	2067

All data sets for use in the neural network were divided into two subsets, one holding the features and one for the response variable. Since neural networks require a numeric response variable, the factor variable was updated to a numeric variable, with 0=No and 1=Yes for these four data sets.

Data Modeling and Results

Models were built to predict stroke incidents using the following methods: K-Nearest Neighbors, Logistic Regression with Elastic Net Regularization, Decision Tree, Random Forest, Support Vector Machine with Linear Kernel, Gradient Boost and Artificial Neural Network.

All models, except for the neural network, were trained using the caret package for cross-validation and hyper-parameter tuning. The neural network was trained using the Keras, Tensorflow and tfruns packages for hyper-parameter tuning and model selection.

Models were built using the train dataset, except for the neural network which was trained with the nnTrain dataset. Details of both were provided in the Data Exploration and Preprocessing section.

Caret Controls

All models trained using the caret package were done so using 10-fold cross validation repeated 5 times. Additional shared parameters were used to provide resampling through ROSE and to use the twoClassSummary function. In summary, trainControl settings were set as indicated below.

parameter	value
method	repeatedcv
number	10
repeats	5
verboseIter	False
classProbs	TRUE
summaryFunction	twoClassSummary
selectionFunction	oneSE
sampling	rose

In addition, all models using caret for tuning and cross-validation used the range preProc option in order to scale and center all numeric values along with the ROC metric. In order to have consistent folds between models for accurate comparisons a common seed was set prior to building each model.

Model specific settings for hyper-parameter tuning will be discussed with each individual model.

Model Evaluation

Evaluation of each model trained with the caret package was done against the validation data set, while the nnVal dataset was used to evaluate the neural network.

While accuracy is a common and easily understood evaluation metric for classification models, it is prone to providing misleading information, especially when evaluating imbalanced data sets.[1] With a 94% vs 6% distribution of negative to positive results, a model could categorize all observations as negative and still achieve a 94% accuracy rating despite it's obvious ineffectiveness in predicting the positive outcome.

Specificity and Sensitivity are common metrics used in medical research.[3]. With stroke detection being highly relevant to the medical field these metrics were chose, along with their related measure ROC-AUC to evaluate model performance. In this way each model was evaluated on its ability to correctly classify a negative result as negative and a positive result as positive.

Model Details and Evaluations

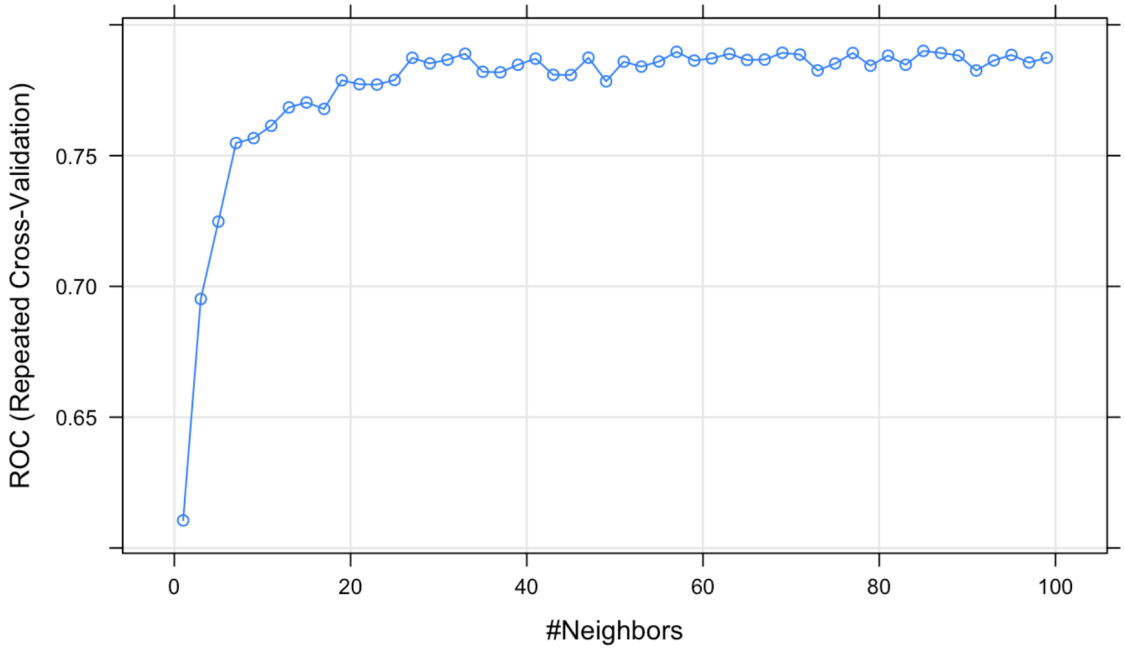
K-Nearest Neighbors

The initial model trained used the knn method in caret and was trained and validated using the one-hot encoded datasets. The tuneGrid settings used were as follows:

Parameter	Value
k	seq(1,100, by=2)

The parameter for k was selected based upon the general practice to begin with k equal to the square root of the number of observations in the dataset[6].

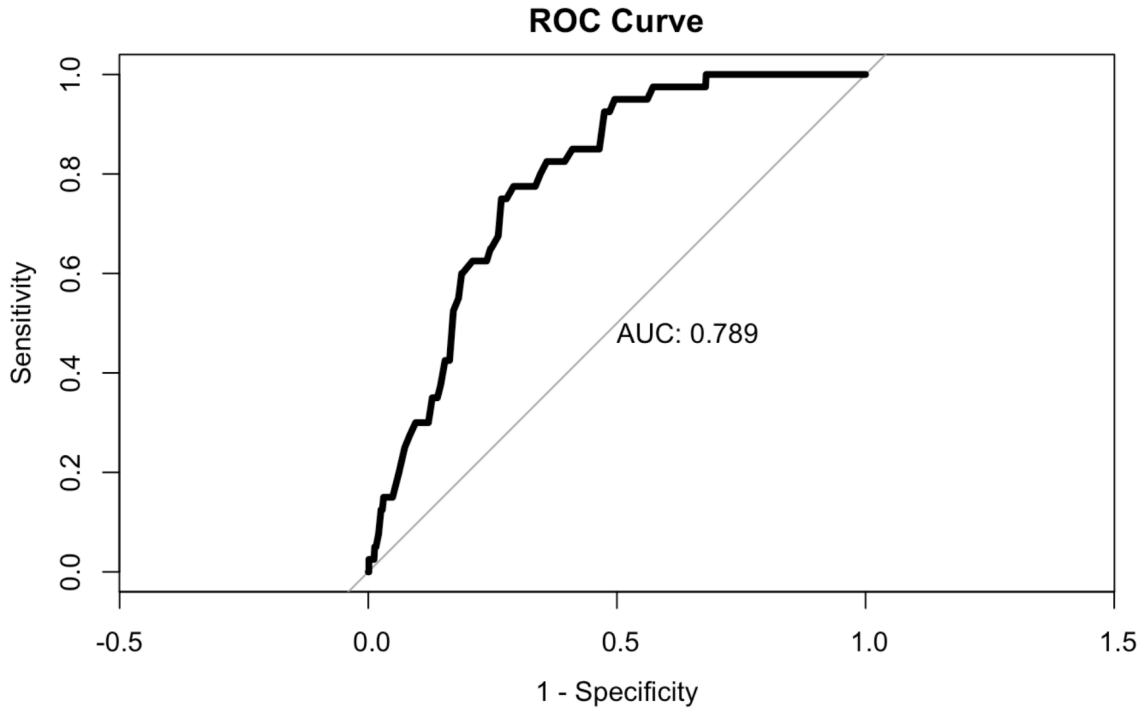
Cross-validation results showed that the optimal value for k was 99.



```
k-Nearest Neighbors

3271 samples
  15 predictor
   2 classes: 'No', 'Yes'

Pre-processing: re-scaling to [0, 1] (15)
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 2943, 2944, 2944, 2944, 2944, 2944, ...
Addtional sampling using ROSE prior to pre-processing

Resampling results across tuning parameters:

  k   ROC        Sens       Spec
   1  0.6105385  0.8048271  0.41625
   3  0.6951932  0.8315020  0.42375
   5  0.7247873  0.8163251  0.46125
   7  0.7548163  0.8279003  0.45875
   9  0.7566825  0.8289298  0.47500
  11  0.7614193  0.8243650  0.49000
  13  0.7684376  0.8212126  0.49750
  15  0.7703217  0.8217953  0.50625
  17  0.7678556  0.8244956  0.47125
  19  0.7788091  0.8132480  0.51250
  21  0.7773042  0.8133047  0.50125
  23  0.7771587  0.8135640  0.53500
  25  0.7789210  0.8165871  0.51375
  27  0.7874020  0.8105468  0.54000
  29  0.7852502  0.8106664  0.52125
  31  0.7866838  0.8080365  0.55000
  33  0.7889458  0.8060413  0.54125
  35  0.7820573  0.8046302  0.55500
  37  0.7818359  0.8024411  0.54125
  39  0.7847358  0.8050791  0.54125
  41  0.7870788  0.7989082  0.58000
  43  0.7808888  0.7982614  0.56250
  45  0.7808209  0.7917104  0.56625
  47  0.7874359  0.7952428  0.57250
  49  0.7784092  0.7935706  0.56750
  51  0.7859306  0.7906159  0.58875
  53  0.7840340  0.7929279  0.58125
  55  0.7859753  0.7889426  0.59125
  57  0.7896676  0.7908739  0.59125
  59  0.7863470  0.7875934  0.59125
  61  0.7871621  0.7915774  0.58625
  63  0.7889519  0.7896496  0.58625
  65  0.7865711  0.7832791  0.60250
  67  0.7867171  0.7789822  0.60750
  69  0.7893115  0.7787804  0.61375
  71  0.7886373  0.7805204  0.61500
  73  0.7825669  0.7715238  0.61000
  75  0.7852127  0.7722908  0.61750
  77  0.7892454  0.7787225  0.61125
  79  0.7843856  0.7742163  0.60875
  81  0.7882281  0.7762794  0.61625
  83  0.7847296  0.7742798  0.61125
  85  0.7900390  0.7643229  0.63500
  87  0.7892112  0.7690075  0.63750
  89  0.7883087  0.7713262  0.60875
  91  0.7825409  0.7645117  0.61250
  93  0.7864030  0.7708818  0.61625
  95  0.7885036  0.7717120  0.61500
  97  0.7855760  0.7701090  0.61875
  99  0.7874074  0.7652774  0.64750

ROC was used to select the optimal model using  the one SE rule.
The final value used for the model was k = 99.
```

Using the selected model to make predictions from the validation dataset provided an AUC of 0.789, Sensitivity of 0.62500 and Specificity of 0.77735.

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  604  15
       Yes 173  25

               Accuracy : 0.7699
                 95% CI : (0.7395, 0.7983)
    No Information Rate : 0.951
    P-Value [Acc > NIR] : 1

                  Kappa : 0.14

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.62500
            Specificity : 0.77735
         Pos Pred Value : 0.12626
         Neg Pred Value : 0.97577
             Prevalence : 0.04896
         Detection Rate : 0.03060
   Detection Prevalence : 0.24235
      Balanced Accuracy : 0.70117

       'Positive' Class : Yes
```



Logistic Regression with Elastic Net Regularization

The second model evaluated was a logistic regression using elastic net regularization. This model was trained and validated using the one-hot encoded datasets with tuning parameters set as follows:

Parameter	Value
alpha	seq(0,1,length=10)
lamda	10*seq(-3,3, length=100)

Cross-validation results showed that for the best model the feature coefficients were:

```
16 x 1 sparse Matrix of class "dgCMatrix"
                                       1
(Intercept)                   -1.166219516
age                            1.480281231
hypertension                   0.333618442
heart_disease                  0.001788369
ever_married                   0.211458168
work_type.children            -0.249526776
work_type.Govt_job             .
work_type.Never_worked         .
work_type.Private              .
work_type.Self_employed        .
avg_glucose_level              0.124124895
bmi                            .
smoking_status.formerly_smoked .
smoking_status.never_smoked    .
smoking_status.smokes          .
smoking_status.unknown         .
```

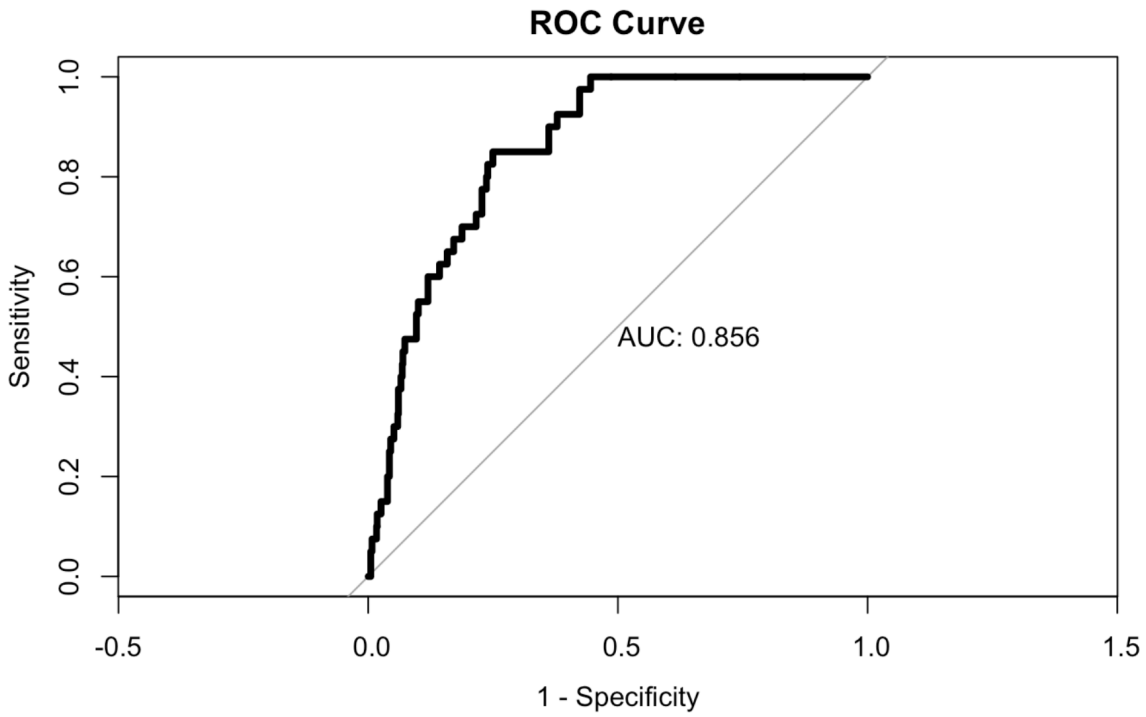
with the scaled variable importance for the included features being:

	Overall <dbl>
age	100.0000000
hypertension	22.5375040
work_type.children	16.8567141
ever_married	14.2849996
avg_glucose_level	8.3852239
heart_disease	0.1208128

(all other features showed a variable importance of zero)

Evaluation metrics showed an AUC of 0.856, Sensitivity of 0.85000 and Specificity of 0.70914.

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  551   6
       Yes 226  34

               Accuracy : 0.716
                 95% CI : (0.6838, 0.7467)
    No Information Rate : 0.951
    P-Value [Acc > NIR] : 1

                  Kappa : 0.155

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.85000
            Specificity : 0.70914
         Pos Pred Value : 0.13077
         Neg Pred Value : 0.98923
             Prevalence : 0.04896
         Detection Rate : 0.04162
   Detection Prevalence : 0.31824
      Balanced Accuracy : 0.77957

       'Positive' Class : Yes
```



Decision Tree

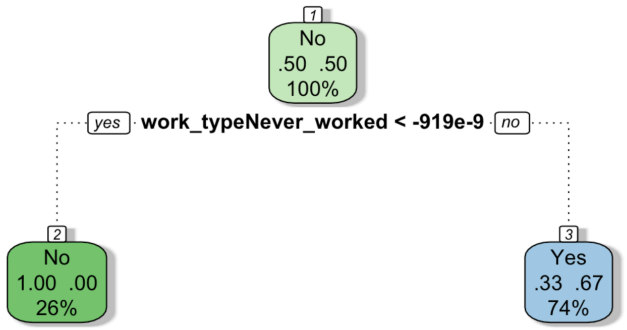
Next a decision tree using the caret method 'rpart' was built and evaluated using the following parameters for tuning:

Parameter	Value
cp	seq(0,1.0, by=0.001)

Initial results showed a tree that simply predicted a 'Yes' value for all observations in the validation dataset, demonstrating no predictive power for stroke.

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No    0   0
       Yes 777  40

               Accuracy : 0.049
                 95% CI : (0.0352, 0.0661)
    No Information Rate : 0.951
    P-Value [Acc > NIR] : 1

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.00000
            Specificity : 0.00000
         Pos Pred Value : 0.04896
         Neg Pred Value :     NaN
             Prevalence : 0.04896
         Detection Rate : 0.04896
   Detection Prevalence : 1.00000
      Balanced Accuracy : 0.50000

       'Positive' Class : Yes
```

The tree itself contained one single split:

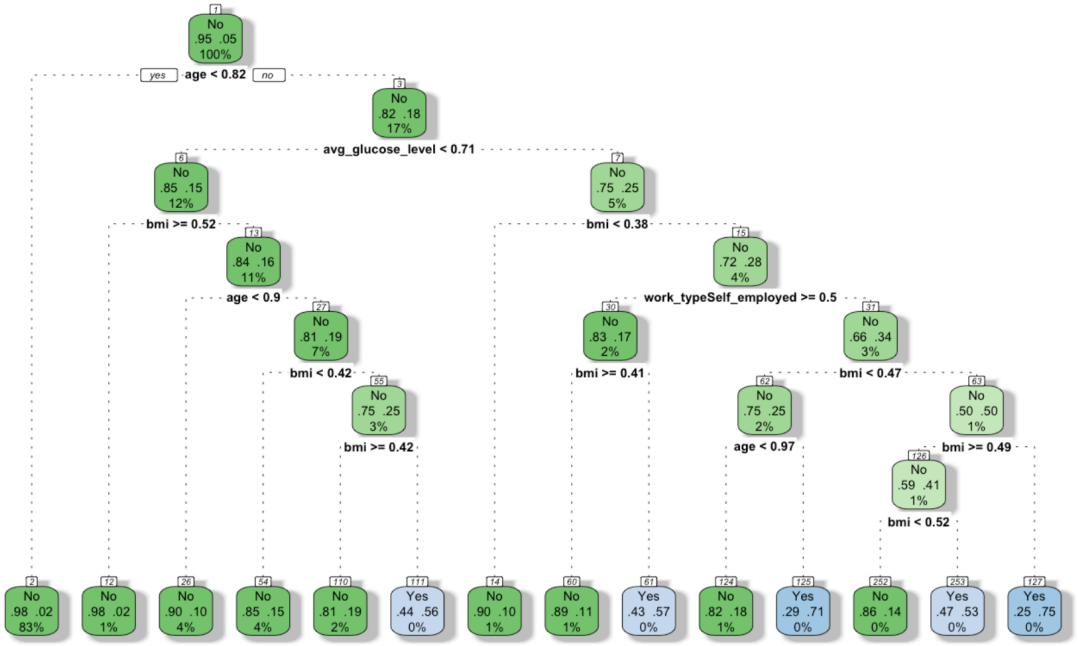


A modified trainControl was then used on a second decision tree model, keeping all parameters the same except for balancing the dataset as part of the cross-validation process:

parameter	value
method	repeatedcv
number	10
repeats	5
verboseIter	False

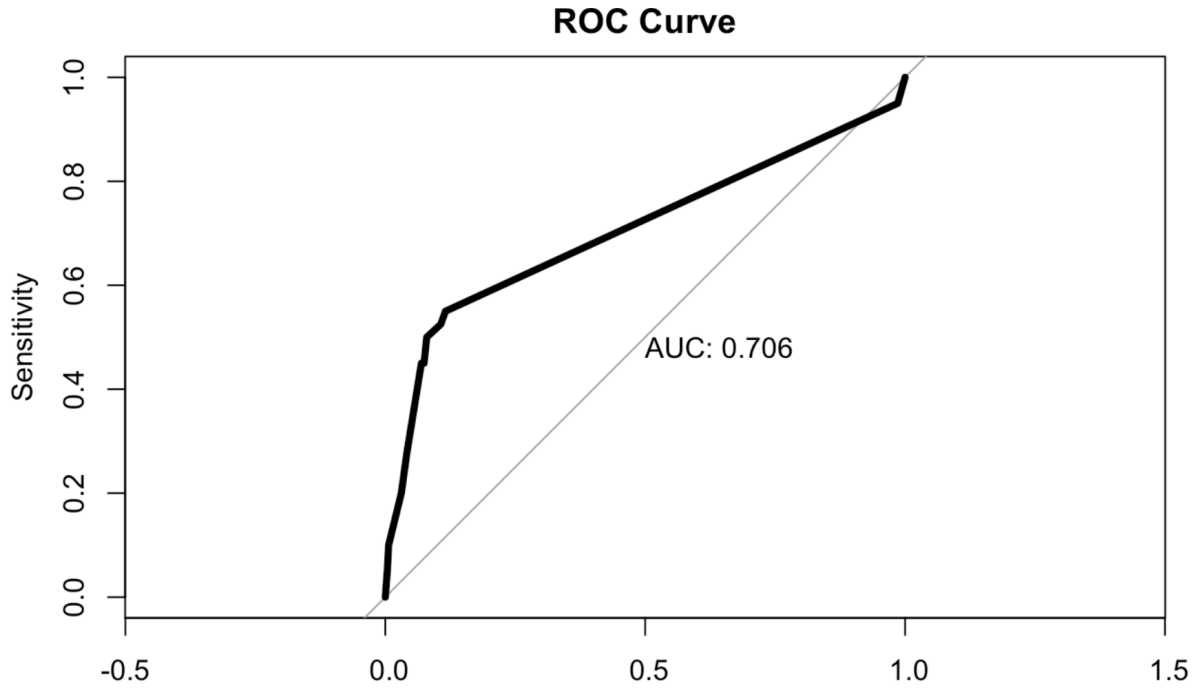
classProbs	TRUE
summaryFunction	twoClassSummary
selectionFunction	oneSE

This second decision tree showed more complexity than the first:



The overall predictive power of the tree, however, remained minimal with an AUC of 0.706, Sensitivity of 0.100 and Specificity of 0.993565. Again, the predictions fell mainly into one class although in this instance they were classified as ‘No’ rather than ‘Yes’ as in the previous decision tree model.

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  772  36
       Yes   5   4

               Accuracy : 0.9498
                 95% CI : (0.9325, 0.9638)
    No Information Rate : 0.951
    P-Value [Acc > NIR] : 0.6049

                  Kappa : 0.1479

 Mcnemar's Test P-Value : 2.797e-06

            Sensitivity : 0.100000
            Specificity : 0.993565
         Pos Pred Value : 0.444444
         Neg Pred Value : 0.955446
             Prevalence : 0.048960
         Detection Rate : 0.004896
   Detection Prevalence : 0.011016
      Balanced Accuracy : 0.546782

       'Positive' Class : Yes
```



Random Forest

A random forest model was then built and evaluated using the 'rf' method in caret. The tuning parameters were set as follows:

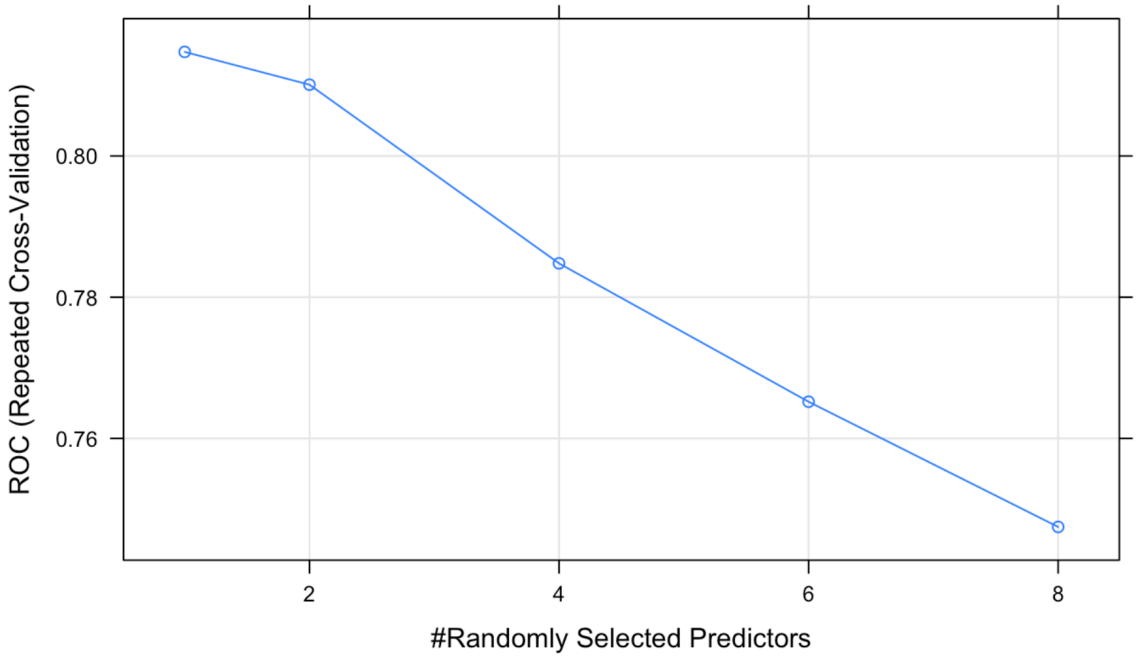
Parameter	Value
ntry	c(1,2,4,6,8)

Evaluation of the selected model shows the following variable importance:

	Overall <dbl>
work_typeNever_worked	100.0000000
age	35.8592231
heart_disease	21.5017294
hypertension	15.9130944
ever_married	11.5458605
avg_glucose_level	9.8683627
work_typeSelf_employed	6.4286074
bmi	4.5351702
work_typeGovt_job	1.7138762
smoking_statussmokes	1.0931829
smoking_statusnever_smoked	0.9678601
work_typePrivate	0.4236776
smoking_statusunknown	0.0000000

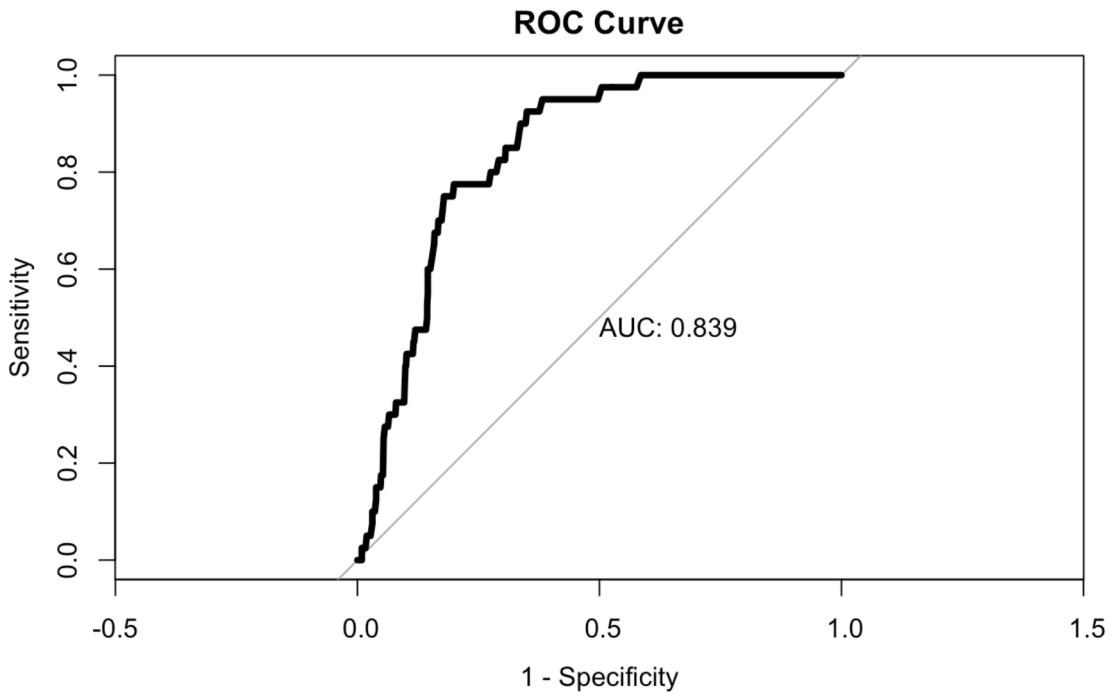
The tuning process indicated that the optimal value for mtry was 1.

```
Random Forest

3271 samples
   8 predictor
   2 classes: 'No', 'Yes'

Pre-processing: re-scaling to [0, 1] (13)
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 2943, 2944, 2944, 2944, 2944, 2944, ...
Addtional sampling using ROSE prior to pre-processing

Resampling results across tuning parameters:

  mtry  ROC        Sens         Spec
  1     0.8147439  0.539185011  0.92125
  2     0.8100990  0.210691112  0.99125
  4     0.7847946  0.004501195  1.00000
  6     0.7651958  0.004501195  1.00000
  8     0.7474645  0.004501195  1.00000

ROC was used to select the optimal model using  the one SE rule.
The final value used for the model was mtry = 1.
```



Performance metrics showed an AUC of 0.839, Sensitivity of 0.95000 and Specificity of 0.53539.

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  416   2
       Yes 361  38

               Accuracy : 0.5557
                 95% CI : (0.5209, 0.5901)
    No Information Rate : 0.951
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0923

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.95000
            Specificity : 0.53539
         Pos Pred Value : 0.09524
         Neg Pred Value : 0.99522
             Prevalence : 0.04896
         Detection Rate : 0.04651
   Detection Prevalence : 0.48837
      Balanced Accuracy : 0.74270

       'Positive' Class : Yes
```



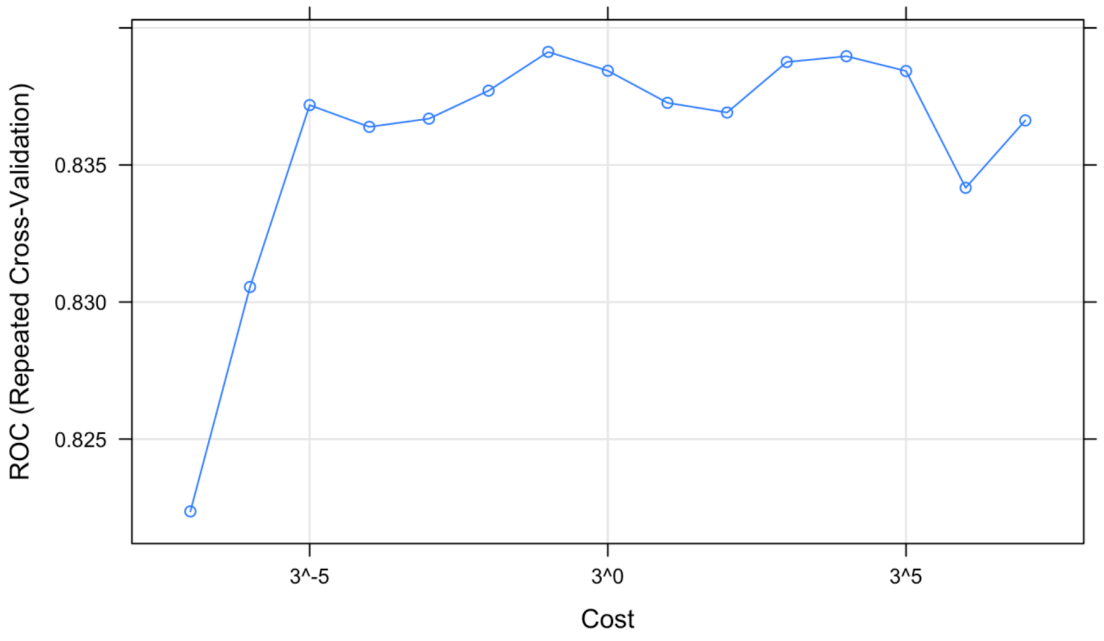
Linear Support Vector Machine

Next, a SVM using a linear kernel was trained and evaluated using the 'svmLinear' method from caret. The model was trained and evaluated using the one-hot encoded versions of the train and validation datasets respectively. Based upon the recommendations by Chih-Wei Hsu, et al.[4], an exponentially growing sequence for C was chosen:

Parameter	Value
C	3**(-7:7)

Cross-validation found the optimal cost value (C) to be 0.004115226.

```
Support Vector Machines with Linear Kernel

3271 samples
  15 predictor
   2 classes: 'No', 'Yes'

Pre-processing: re-scaling to [0, 1] (15)
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 2943, 2944, 2944, 2944, 2944, 2944, ...
Addtional sampling using ROSE prior to pre-processing

Resampling results across tuning parameters:

  C             ROC        Sens       Spec
  4.572474e-04  0.8223611  0.7261314  0.77000
  1.371742e-03  0.8305492  0.7266456  0.79875
  4.115226e-03  0.8371825  0.7276736  0.81625
  1.234568e-02  0.8363893  0.7294760  0.80875
  3.703704e-02  0.8366878  0.7251682  0.81625
  1.111111e-01  0.8377090  0.7283768  0.80875
  3.333333e-01  0.8391237  0.7261312  0.81250
  1.000000e+00  0.8384358  0.7249744  0.81375
  3.000000e+00  0.8372659  0.7269680  0.81125
  9.000000e+00  0.8369120  0.7234337  0.80875
  2.700000e+01  0.8387552  0.7233682  0.81625
  8.100000e+01  0.8389665  0.7263268  0.81625
  2.430000e+02  0.8384265  0.7249740  0.81625
  7.290000e+02  0.8341659  0.7413742  0.78125
  2.187000e+03  0.8366235  0.7315947  0.81000

ROC was used to select the optimal model using  the one SE rule.
The final value used for the model was C = 0.004115226.
```

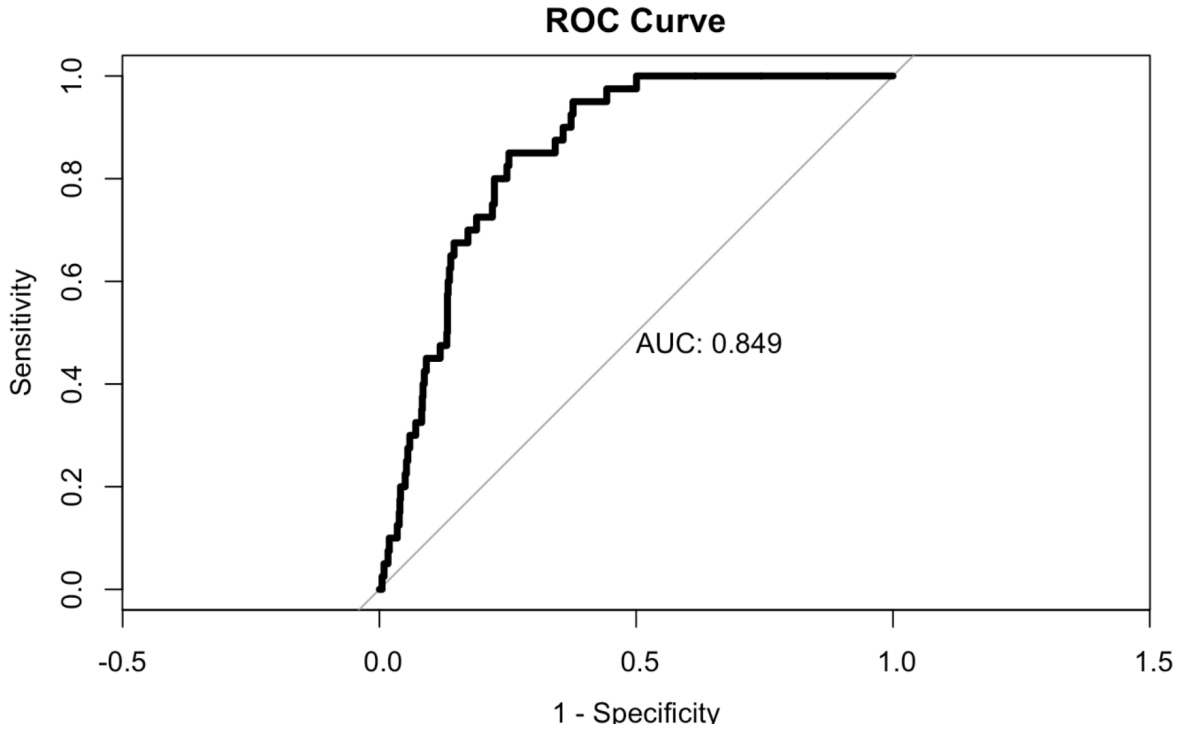


The feature with the largest importance was Age, distantly followed by ever_married and avg_glucose_level.

	Importance <dbl>
age	100.0000000
ever_married	36.7162939
avg_glucose_level	34.6602639
hypertension	27.6005809
bmi	25.3814509
heart_disease	23.5803783
work_type.children	20.4933017
work_type.Self_employed	18.6916228
smoking_status.formerly_smoked	17.7700169
smoking_status.unknown	14.9539652
smoking_status.never_smoked	6.3960662
smoking_status.smokes	3.5800145
work_type.Govt_job	2.4807569
work_type.Never_worked	0.6275409
work_type.Private	0.0000000

Evaluation metrics show a good balance between Sensitivity (0.82500) and Specificity (0.75161) with an AUC of 0.849

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  584   7
       Yes 193  33

               Accuracy : 0.7552
                 95% CI : (0.7242, 0.7843)
    No Information Rate : 0.951
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1799

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.82500
            Specificity : 0.75161
         Pos Pred Value : 0.14602
         Neg Pred Value : 0.98816
             Prevalence : 0.04896
         Detection Rate : 0.04039
   Detection Prevalence : 0.27662
      Balanced Accuracy : 0.78830

       'Positive' Class : Yes
```

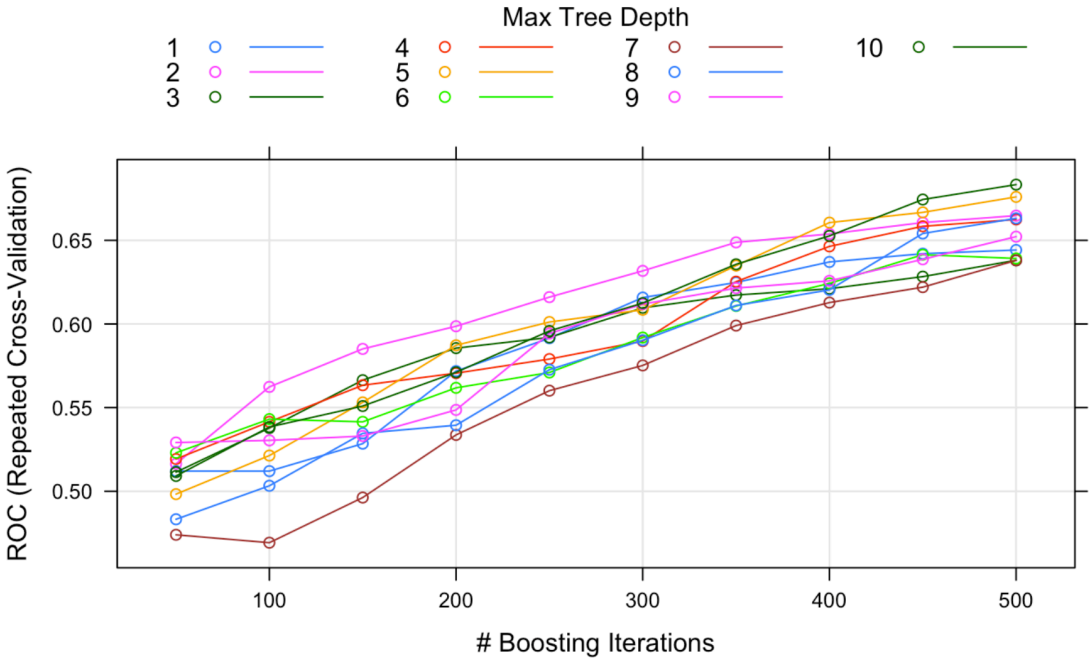


Stochastic Gradient Boost

Next, a stochastic gradient boost model was trained on the one-hot encoded train dataset using the 'gbm' method from caret. The model was set to auto-tune with a tuneLength of 10.

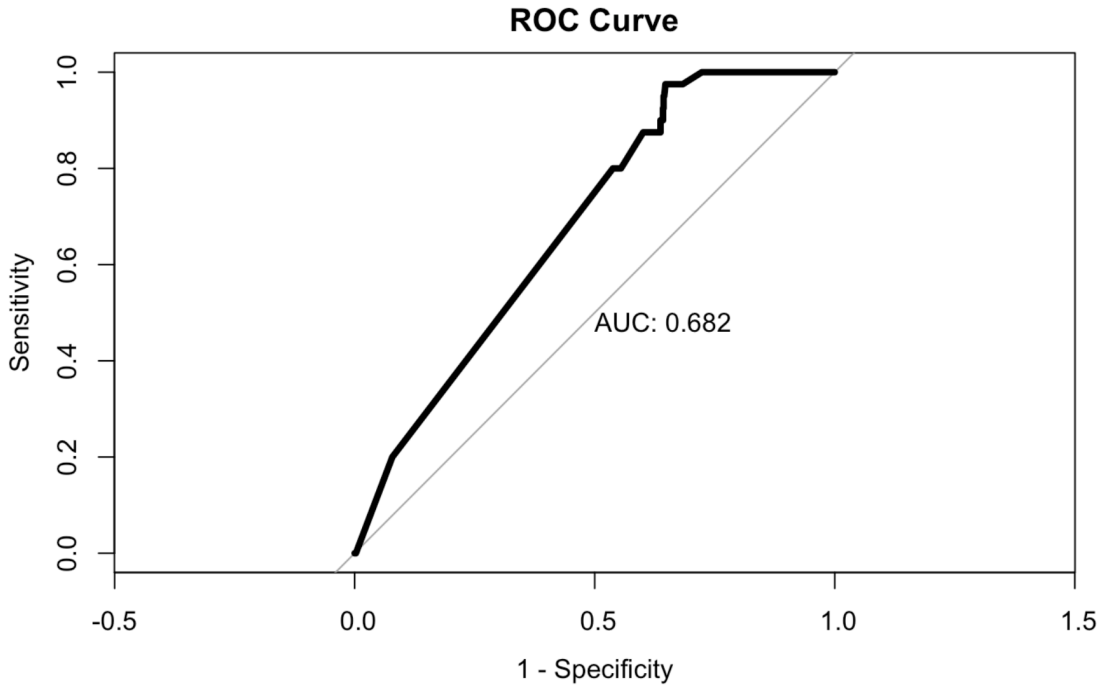
The final values selected during the cross-validation process were:

Parameter	Value
n.trees	450
interaction.depth	10
shrinkage	0.1
n.minobsinmode	10



Evaluation metrics for the gradient boost model showed that most values were simply classified as 'Yes', providing no true predictive power for stroke.

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No    3   0
       Yes 774  40

               Accuracy : 0.0526
                 95% CI : (0.0383, 0.0702)
    No Information Rate : 0.951
    P-Value [Acc > NIR] : 1

                  Kappa : 4e-04

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.000000
            Specificity : 0.003861
         Pos Pred Value : 0.049140
         Neg Pred Value : 1.000000
             Prevalence : 0.048960
         Detection Rate : 0.048960
   Detection Prevalence : 0.996328
      Balanced Accuracy : 0.501931

       'Positive' Class : Yes
```



Neural Network

The final model evaluated was a neural network built using the Keras and tfruns packages. The model built consisted of two hidden layers, each with a dropout layer immediately after. As this is a binary classification problem, the loss was set as 'binary_crossentropy' and the output layer used a sigmoid activation. Metrics were set as 'accuracy'.

```
model <- keras_model_sequential()
model %>%
 layer_dense(units = FLAGS$nodes1, activation = FLAGS$activation, input_shape = 15) %>%
 layer_dropout(FLAGS$drop1) %>%
 layer_dense(units = FLAGS$nodes2, activation = FLAGS$activation) %>%
 layer_dropout(FLAGS$drop2) %>%
 layer_dense(units = 1, activation = 'sigmoid')

model %>% compile(
 optimizer = optimizer_adam(lr=FLAGS$learning_rate),
 loss = "binary_crossentropy",
 metrics=c('accuracy'))

model %>% fit(
 as.matrix(nnTrainFeatures), as.matrix(nnTrainLabels), epochs = FLAGS$epochs, batch_size=
FLAGS$batch_size, validation_data=list(as.matrix(nnValFeatures), as.matrix(nnValLabels)))
```

Tuning parameters were set as follows:

Parameter	Value
nodes1	c(20,50,75)
nodes2	c(20,50,75)
drop1	c(0.2,0.4,0.6)
drop2	c(0.2,0.4,0.6)
learning_rate	c(0.01,0.05,0.001,0.0001)
epochs	c(25,50,100)
batch_size	c(10,25,50)
activation	c('relu', 'sigmoid','tanh')

Two-percent of parameter combinations were sampled to find the optimal model. This restriction was made due to processing time requirements on the author's laptop.

The best model showed a validation accuracy of 0.9510 during training

	run_dir <chr>	metric_val_accuracy <dbl>	metric_loss <dbl>	metric_accuracy <dbl>	metric_val_loss <dbl>
62	runs/2021-04-28T05-06-54Z	0.9510	0.6952	0.5032	0.6749
42	runs/2021-04-28T05-17-39Z	0.8935	0.3309	0.8423	0.3029
17	runs/2021-04-28T05-35-27Z	0.8507	0.3496	0.8435	0.3269
52	runs/2021-04-28T05-11-50Z	0.8433	0.3679	0.8221	0.3554
139	runs/2021-04-28T04-22-13Z	0.8323	0.4309	0.7979	0.3604
167	runs/2021-04-28T04-09-16Z	0.8262	0.3262	0.8539	0.3859
160	runs/2021-04-28T04-12-45Z	0.8225	0.3670	0.8364	0.3611
24	runs/2021-04-28T05-30-59Z	0.8213	0.3084	0.8560	0.3672
89	runs/2021-04-28T04-49-29Z	0.8176	0.4878	0.7625	0.3605
175	runs/2021-04-28T04-04-00Z	0.8164	0.3695	0.8248	0.3828

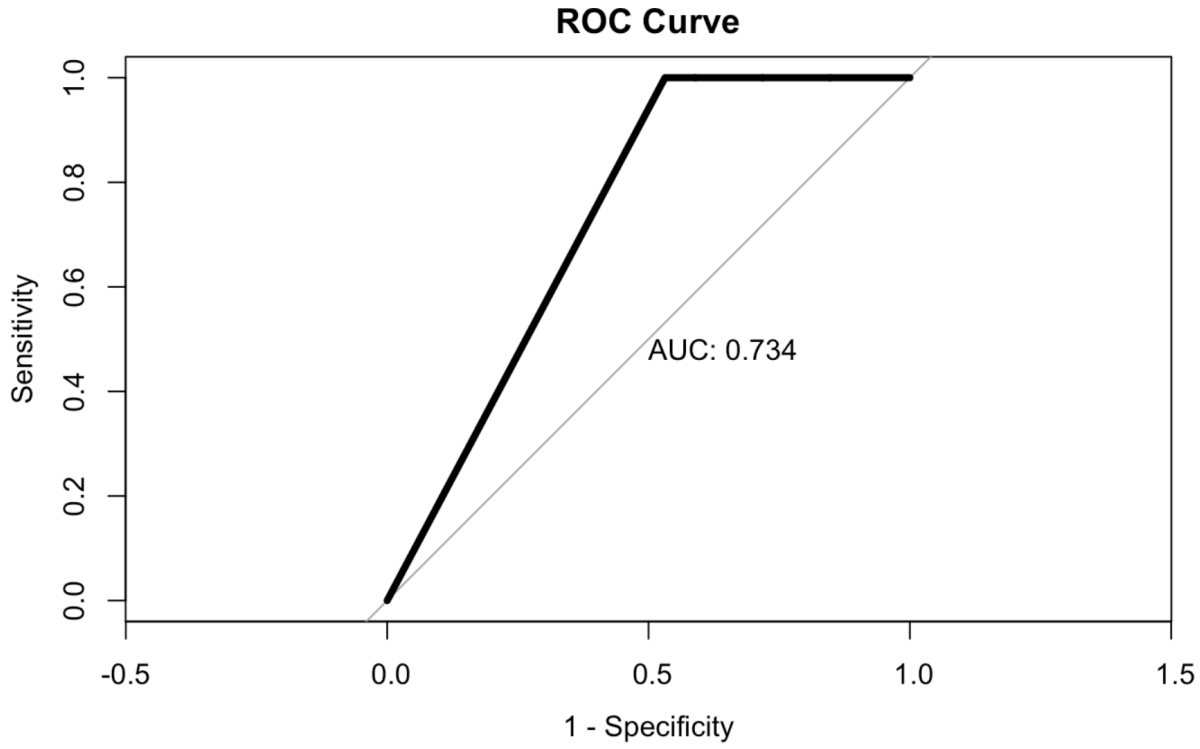
1-10 of 10 rows

The following parameters were used for the best run:

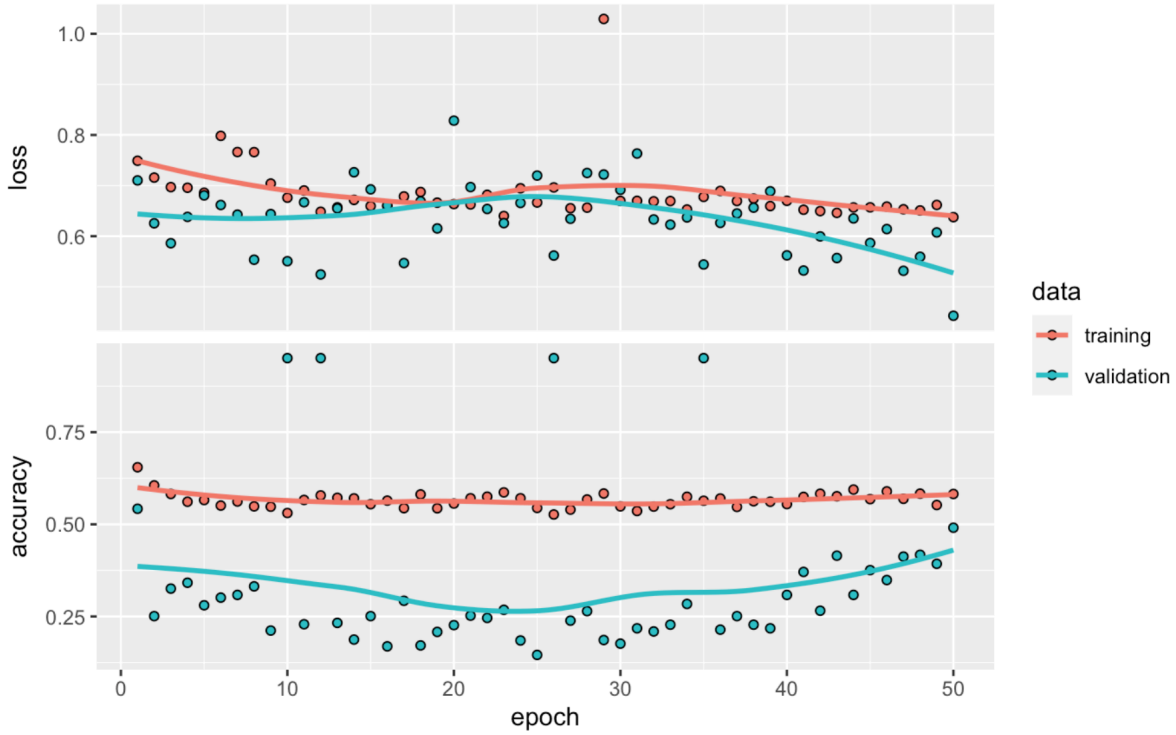
Parameter	Value
nodes1	50
nodes2	50
drop1	.4
drop2	.6
learning_rate	.05
epochs	50
batch_size	10
activation	relu

Building a neural network model using these parameters and then predicting stroke from the validation dataset showed a model that predicted all stroke events correctly, but also classified a large majority of non-stroke observations as stroke.

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  361   0
       Yes 416  40

               Accuracy : 0.4908
                 95% CI : (0.456, 0.5257)
    No Information Rate : 0.951
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0783

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.00000
            Specificity : 0.46461
         Pos Pred Value : 0.08772
         Neg Pred Value : 1.00000
             Prevalence : 0.04896
         Detection Rate : 0.04896
   Detection Prevalence : 0.55814
      Balanced Accuracy : 0.73230

       'Positive' Class : Yes
```



Plotting the progression of the model through each epoch, it can be seen that the model continues to overfit the training data.

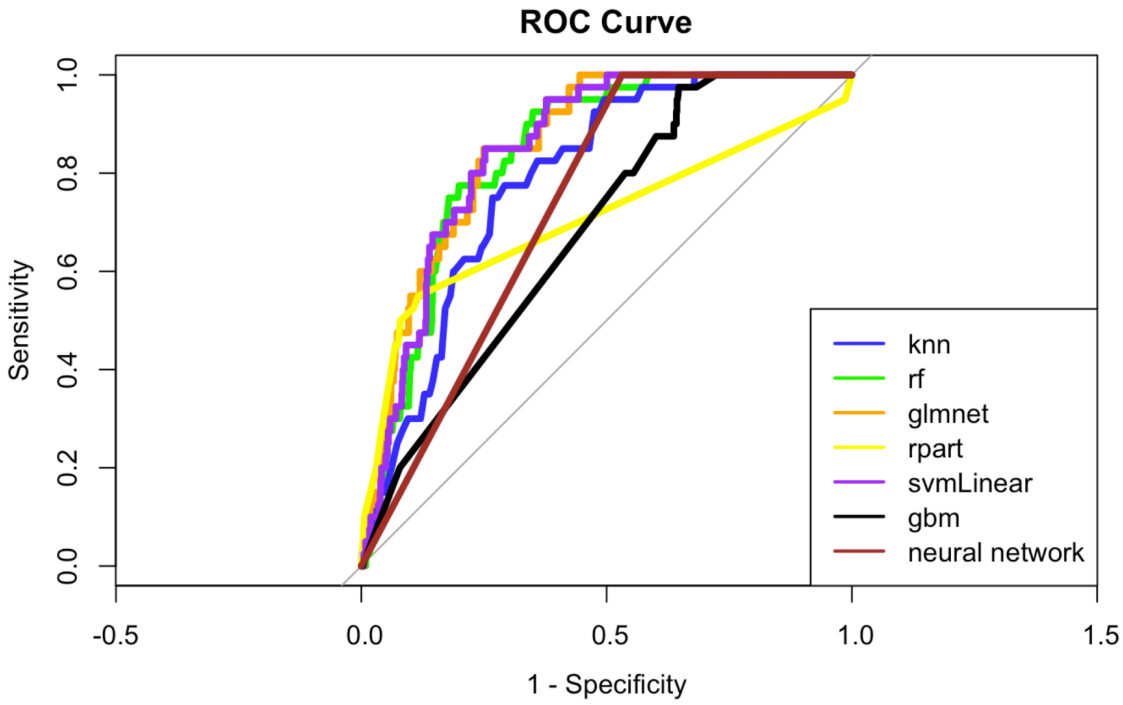


Model Selection

The table below shows a comparison of the metrics for each model, followed by a combined ROC plot.

Model Performace Comparison

	ROC	Spec	Sens
knn	0.7889	0.7773	0.625
glmnet	0.8555	0.7091	0.85
rpart	0.7064	0.9936	0.1
rf	0.8387	0.5354	0.95
svmLinear	0.8488	0.7516	0.825
gbm	0.6796	0.003861	1
neural network	0.7342	0.4646	1



Both the elastic net (glmnet) and linear SVM (linearSVM) models showed an ability to correctly classify both the positive (stroke) and negative (non-stroke) observations, while models such as

the decision tree, gradient boost and neural network only succeeded in classifying all If not most of the observations as one class. The remaining models provided good performance on either sensitivity or specificity, but not on the other.

Since in a situation such as a medical diagnosis or prediction it is important to be able to accurately separate the two classes, rather than just identify one of the two, the only models left under consideration are the elastic net and linear SVM.

With an overall AUC score slightly higher combined with a model that is simpler, both from the number of features involved and from the ease of interpretability between the two models, the elastic net model was selected as the better model.

Final Model Verification

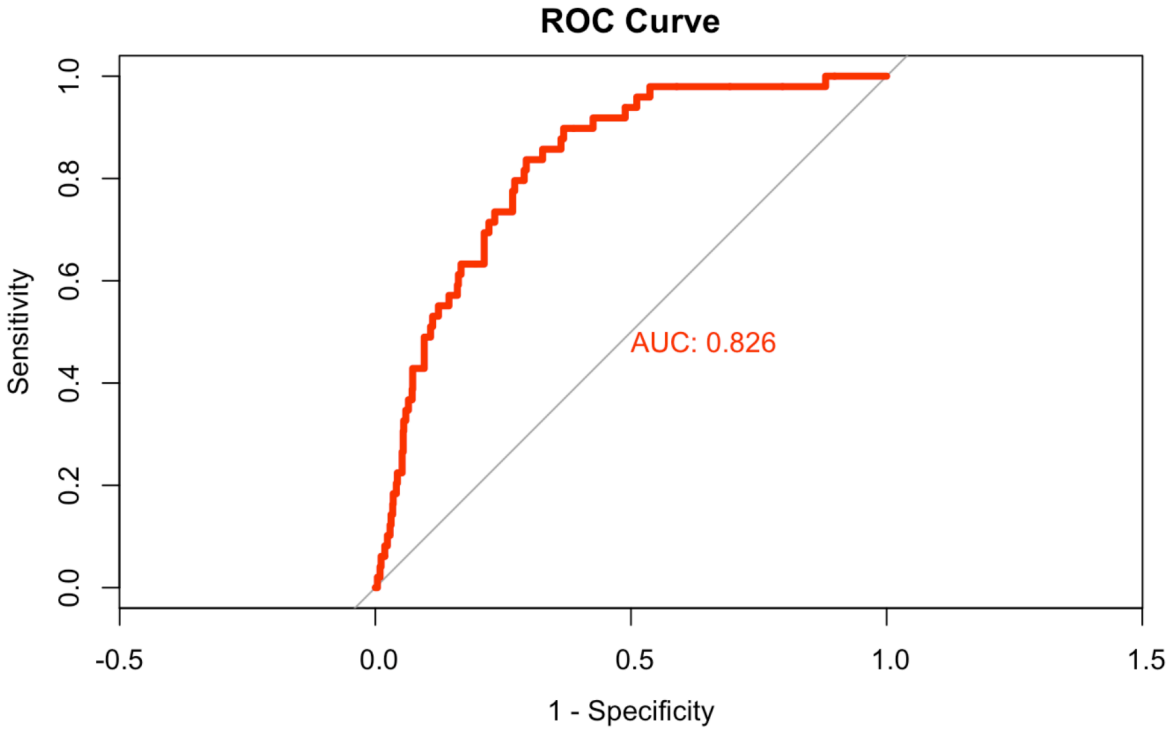
To verify the performance of the elastic net model, a single model was built using the glmnet package. The lambda and alpha values were set based upon the results of the earlier cross-validation.

The model was trained using the original training dataset (80% of full dataset) after scaling the numeric features, encoding all factors using one-hot encoding and rebalancing the response variable using the ROSE package.

The trained model was then fit to the test dataset (20% of full dataset) after first being scaled and one-hot encoded. No rebalancing was done on the test dataset.

Evaluation of the final model verification showed comparable results to that found during the model selection process, with a sensitivity of 0.89796, specificity of 0.62757 and AUC of 0.826.

```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  610   5
       Yes 362  44

               Accuracy : 0.6405
                 95% CI : (0.6103, 0.67)
    No Information Rate : 0.952
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1179

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.89796
            Specificity : 0.62757
         Pos Pred Value : 0.10837
         Neg Pred Value : 0.99187
             Prevalence : 0.04799
         Detection Rate : 0.04310
   Detection Prevalence : 0.39765
      Balanced Accuracy : 0.76277

       'Positive' Class : Yes
```



The final features used in the model are also in alignment with what was found during model selection.

```
16 x 1 sparse Matrix of class "dgCMatrix"
                                            s0
(Intercept)                          0.03373739
age                                  0.28096432
hypertension                         0.02208788
heart_disease                        0.04157673
ever_married                         0.04646855
work_type.children                  -0.13217010
work_type.Govt_job                   .
work_type.Never_worked               .
work_type.Private                    .
work_type.Self_employed              .
avg_glucose_level                    0.02937247
bmi                                  .
smoking_status.formerly_smoked       .
smoking_status.never_smoked          .
smoking_status.smokes                .
smoking_status.unknown               .
```

Conclusion

In training a model to predict a stroke event, a logistic regression model using elastic net regularization was found to provide the best performance compared to a k-Nearest Neighbors, Decision Tree, Random Forest, Linear Support Vector Machine, Gradient Boost Machine and a Neural Network.

The Gradient Boost Machine, Neural Network and Random Forest models all had perfect or near-perfect classification for the stroke event, however this predictive power was at the cost of being able to correctly classify the non-stroke observations. While the high sensitivity is desired, it is also important to maintain a higher level of specificity when dealing with medical related diagnostics.

In training the neural network model, only a small percentage of possible hyper-parameter combinations were explored as part of the model evaluation process. While it is likely that a better performing neural network could be found by evaluating additional combinations, any increase in performance compared to the logistic regression model would need to be balanced with the interpretability of the logistic regression model.

References

1. Brownlee, J. (2021, January 21). *Failure of Classification Accuracy for Imbalanced Class Distributions*. Machine Learning Mastery. https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/
2. Crichton, S. L., Bray, B. D., McKevitt, C., Rudd, A. G., & Wolfe, C. D. A. (2016). Patient outcomes up to 15 years after stroke: survival, disability, quality of life, cognition and mental health. *Journal of Neurology, Neurosurgery & Psychiatry*, *87*(10), 1091–1098. https://doi.org/10.1136/jnnp-2016-313361
3. Evaluation of binary classifiers. (2021, March 24). In *Wikipedia*. https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers
4. Hsu, C., Chang, C., & Lin, C. (2016, May). *A Practical Guide to Support Vector Classification*. https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
5. Kimura, T. (2021, April 4). *Basic Visualization with Matplot & Seaborn*. Kaggle. https://www.kaggle.com/ktakuma/basic-visualization-with-matplot-seaborn
6. Lantz, B. (2019). *Machine Learning with R: Expert techniques for predictive modeling, 3rd Edition* (Third Edition). Packt Publishing.
7. Rajsic, S. (2019). Economic burden of stroke: a systematic review on post-stroke care. *The European Journal of Health Economics : HEPAC : Health Economics in Prevention and Care*, *20*(1), 107–134. https://doi.org/10.1007/s10198-018-0984-0
8. Randhi, S. C. (2021, April 27). *Stroke prediction - EDA and MODEL*. Kaggle. https://www.kaggle.com/saicharan96/stroke-prediction-eda-and-model
9. *Stroke Prediction Dataset*. (2021a, January 26). [Dataset]. https://www.kaggle.com/fedesoriano/stroke-prediction-dataset
10. *Stroke Prediction Dataset*. (2021b, January 26). Kaggle. https://www.kaggle.com/fedesoriano/stroke-prediction-dataset/activity
11. World Health Organization. (2020, December 9). *The top 10 causes of death*. https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death