STROKE PREDICTION

Thomas Champion

UIS CSC 532B

Spring 2021

Project Definition and Goals

- Strokes account for approximately 11% of deaths worldwide
- Survivors may experience lasting effects after a stroke
  - Depression
  - Physical impairments
- Aside from physical impact, there are financial impacts associated with a stroke

PROJECT DEFINITION AND GOALS

- GOAL IS TO FIND A MODEL THAT WILL PREDICT A STROKE BASED UPON DATASET CONTAINING INFORMATION OF INDIVIDUALS THAT HAVE AND HAVE NOT EXPERIENCED A STROKE

Stroke Prediction Dataset

- Dataset was obtained from kaggle.com
- Originally posted in January 2021
- ~177K views
- ~26.8K downloads
- 346 unique contributors
  - Data Visualization
  - Data Cleaning and Balancing
  - Stroke Prediction

DATA EXPLORATION

- 5110 OBSERVATION
- 12 VARIABLES
  - 1 BINARY RESPONSE VARIABLE
  - 11 PREDICTIVE 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

Data Exploration and Preprocessing

Type Conversions

- Converted ever_married to numeric
  - 0 = No, 1 = Yes
- Converted Residence_type to numeric
  - 0 = Rural, 1 = Urban
- Removed 1 observation with gender = 'Other'
- Converted gender to Numeric
  - 0 = Female, 1 = Male
- Converted work_type to factor
- Converted smoking_status to factor

Data Exploration and Preprocessing

Stroke Response Variable

- Highly imbalanced data
  - 4861 observations = No Stroke
  - 249 observations = Stroke
- Will be necessary to include balancing as part of the modeling process

Data Exploration and Preprocessing

BMI

- BMI is character field storing numeric values
  - Contains 'N/A' as values
- Convert 'n/a' to NA
- Convert field to numeric
- 201 missing values

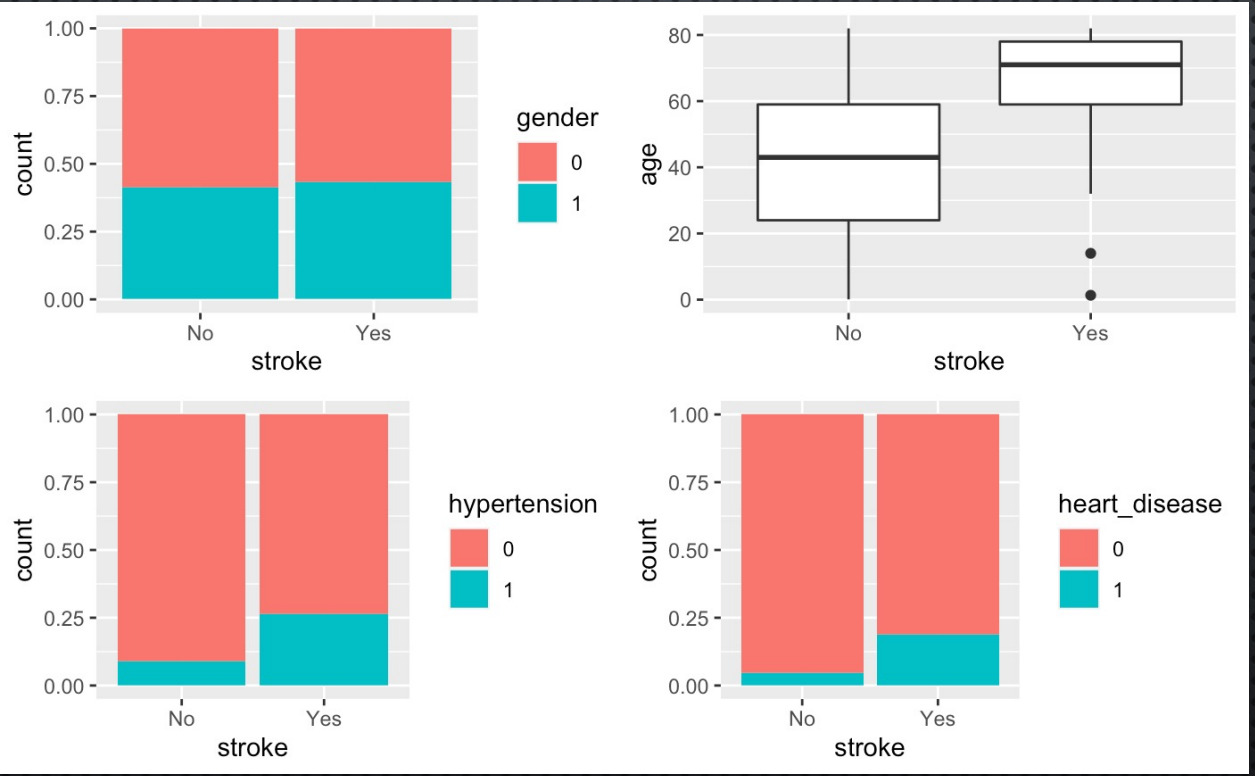
Data Exploration and Preprocessing

BMI

- Impute missing values
  - Create 10 bins using clustering method on age variable
  - Set missing bmi values equal to the median age value of the bin they belong to

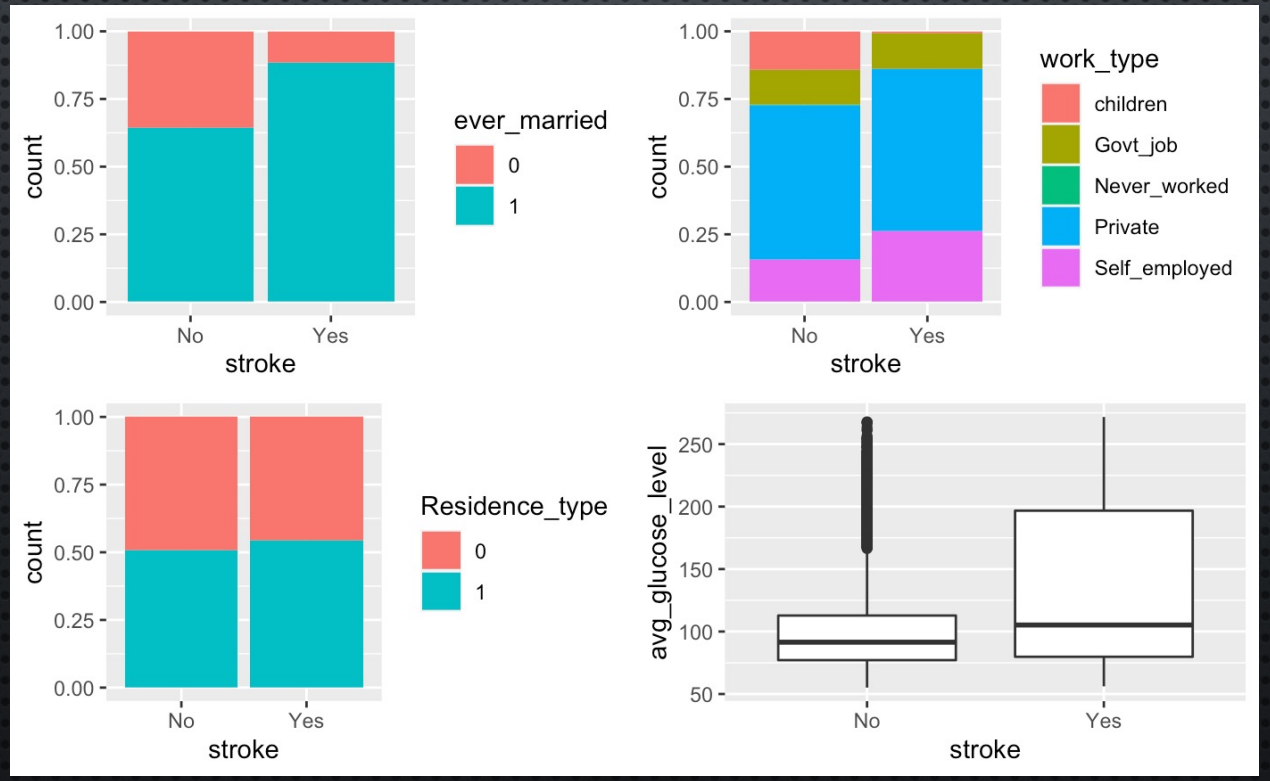
DATA EXPLORATION AND PREPROCESSING

ASSOCIATION ANALYSIS



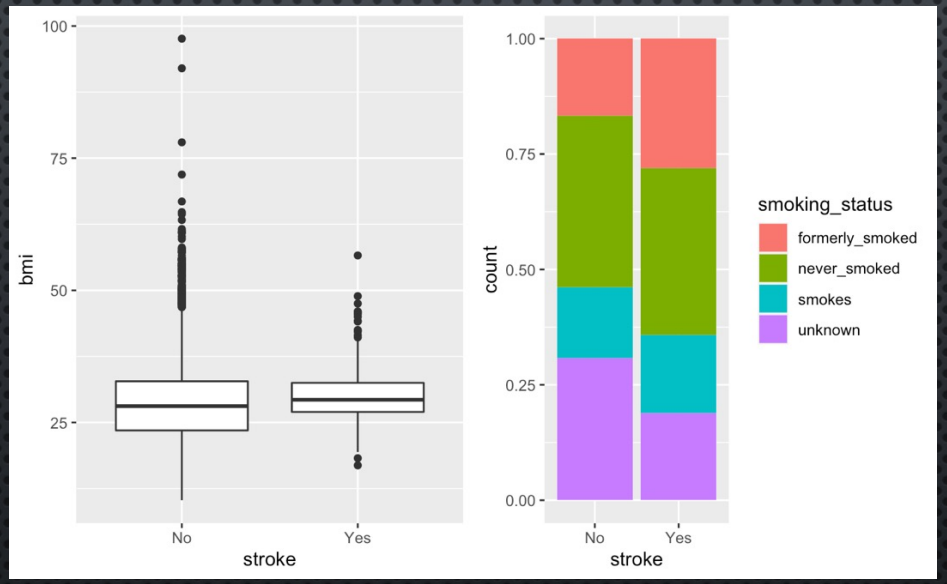
DATA EXPLORATION AND PREPROCESSING

ASSOCIATION ANALYSIS



DATA EXPLORATION AND PREPROCESSING

ASSOCIATION ANALYSIS



DATA EXPLORATION AND PREPROCESSING

ASSOCIATION ANALYSIS

Chi-Square Tests

chisq_labels	chisq_values
smoking_status	2.008e-06
gender	0.5598
hypertension	1.689e-19
heart_disease	2.121e-21
ever_married	1.686e-14
work_type	5.409e-10
residence_type	0.2998

T-Tests

ttest_labels	ttest_values
age	2.176e-95
glucose	2.373e-11
bmi	0.0003064

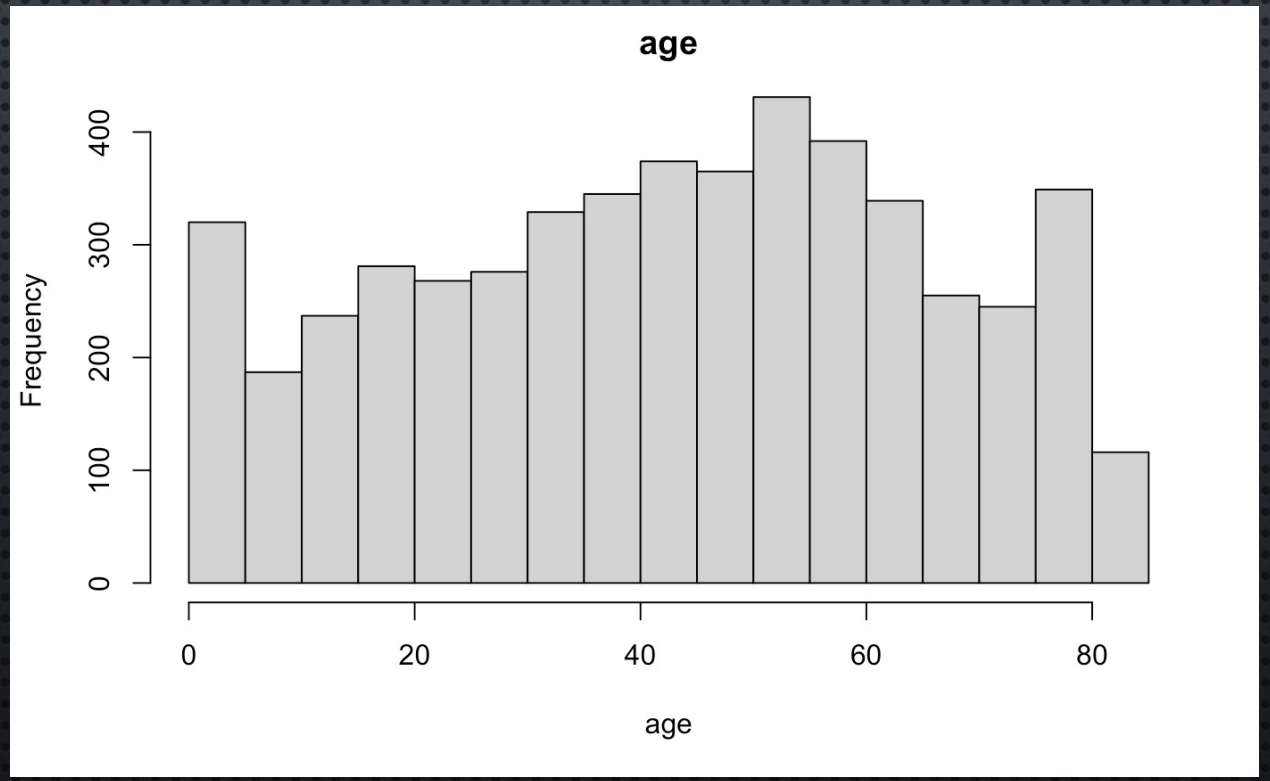
Data Exploration and Preprocessing

Association Analysis

- All variables showed a statistically significant relation with stroke except for residence_type and gender
  - Both variables removed

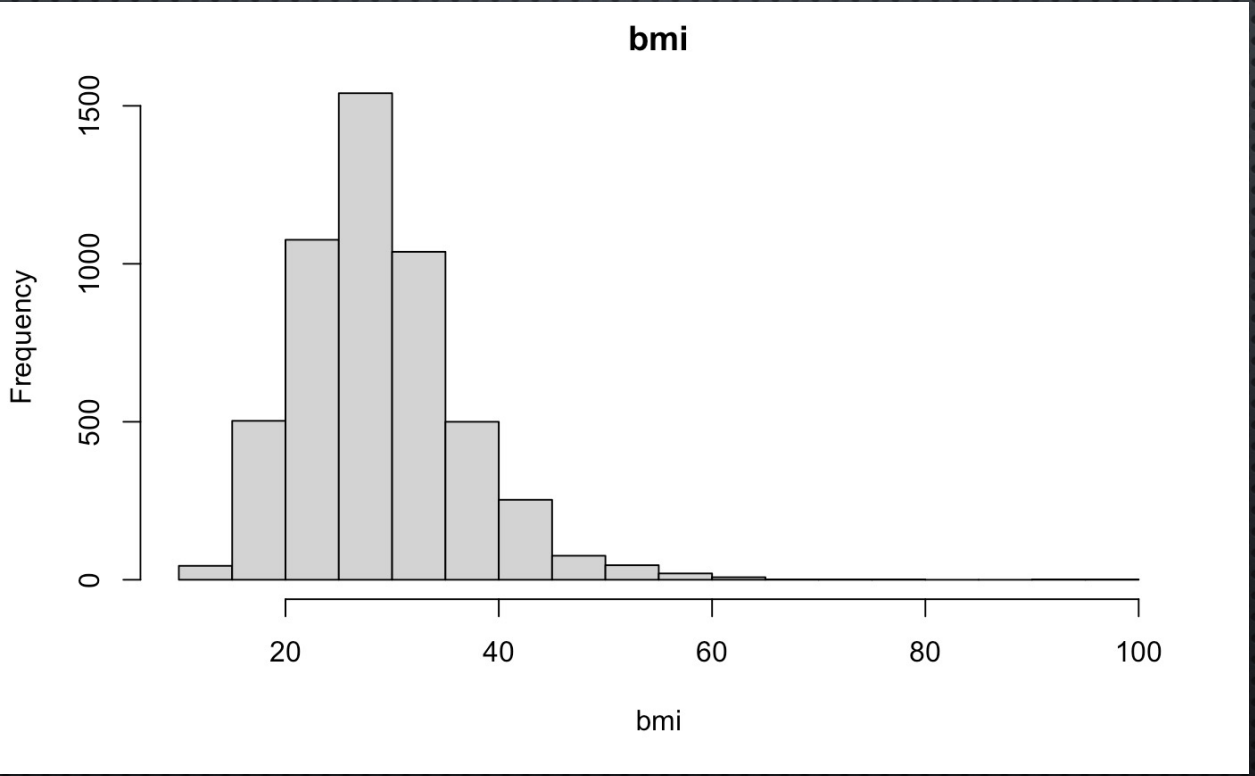
DATA EXPLORATION AND PREPROCESSING

DISTRIBUTION OF NUMERIC VARIABLES



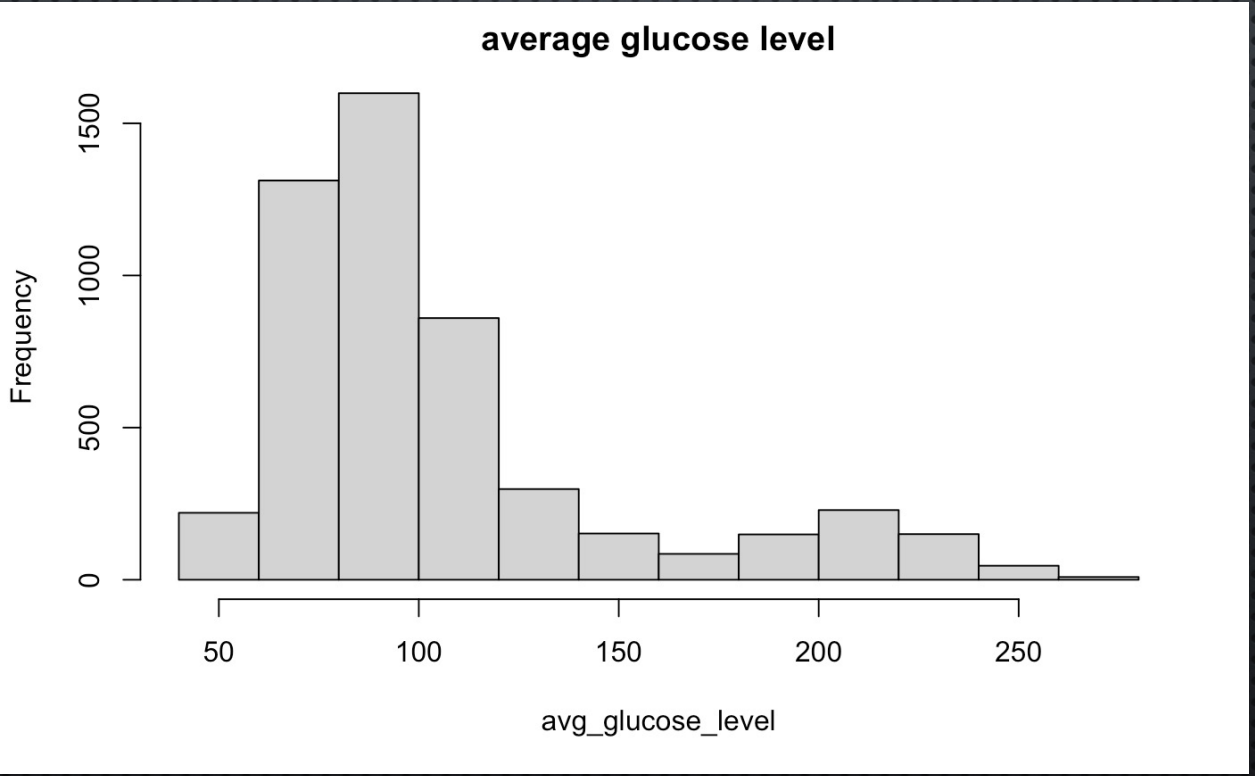
DATA EXPLORATION AND PREPROCESSING

DISTRIBUTION OF NUMERIC VARIABLES



DATA EXPLORATION AND PREPROCESSING

DISTRIBUTION OF NUMERIC VARIABLES



DATA EXPLORATION AND PREPROCESSING

DISTRIBUTION OF NUMERIC VARIABLES

- AVERAGE_GLUCOSE_LEVEL AND BMI ARE RIGHT-SKEWED
- BOTH VARIABLES WERE LOG TRANSFORMED

Data Exploration and Preprocessing

Dataset Preparation

- Initial split of dataset into fullTrain and Test datasets
  - 80% fullTrain, 20% Test
- fullTrain split into test and train datasets
  - 80% traing, 20% test
- Second copy of each new dataset made
  - One set for training with caret
  - One set for training with keras/tfruns

Data Exploration and Preprocessing

Dataset Preparation: caret

- fullTrain dataset balanced using ROSE package
- fullTrain dataset centered and scaled
  - Same instance used to center and scale test dataset
- No centering, scaling or balancing for train or validation datasets
- Copies of all four datasets made and one-hot encoded using dummyVars function

Data Exploration and Preprocessing

Dataset Preparation: keras/tfruns

- nnTrain dataset was scaled using scale function
  - nnValidation dataset scaled using same columns means and standard deviations
- nnFullTrain dataset was scaled using scale function
  - nnTest dataset scaled using same columns means and standard deviations
- Copies of all four datasets made and one-hot encoded using dummyVars function

Model Training and Evaluation

- Multiple models trained and evaluated
  - K-nearest neighbors
  - Logistic regression with elastic net regularization
  - Decision tree
  - Random forest
  - Linear support vector machine
  - Stochastic gradient boost
  - Neural network
- Models were trained using the caret package for all models aside from the neural network

Model Training and Evaluation
caret package

All models trained using the caret package were trained as follows :

- 10-fold cross-validation repeated 5 times
- classProbs were set to be calculated
- twoClassSummary as the summaryFunction
- oneSE as the selectionFunction
- Sampling was done using the ROSE method
- Data was centered and scaled using the preProc option
- Metric for evaluation was ROC

MODEL TRAINING AND EVALUATION

CARET PACKAGE

MODEL TUNING PARAMETERS

Model	Method	Tuning Parameters
K-Nearest Neighbors	knn	k = seq(1,100,by=2)
Logistic Regression with Elastic Net Regularization	glmnet	alpha: seq(0,1,length=10) lambda: 10*seq(-3,3,length=100)
Decision Tree	rpart	cp = seq(0,1,by=0.001)
Random Forest	rf	ntry = c(1,2,4,6,8)
Linear Support Vector Machine	svmLinear	C = 3**(-7:7)
Stochastic Gradient Boost	gbm	Auto-tuned with tuneLength = 10

Model Training and Evaluation Keras/tfruns

- Neural Network was tuned using the tfruns package
- Model was designed with two hidden layers, each followed by a dropout layer
- Loss function was binary crossentropy
- Metric was accuracy
- Final layer used a sigmoid activation

MODEL TRAINING AND EVALUATION KERAS/TFRUNS

```
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)))
```

MODEL TRAINING AND EVALUATION

KERAS/TFRUNS TUNING PARAMETERS

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')

Model Training and Evaluation

- After each model was trained and tuned it was fit to the appropriate validation dataset
- Models were evaluated on AUC, Specificity and Sensitivity

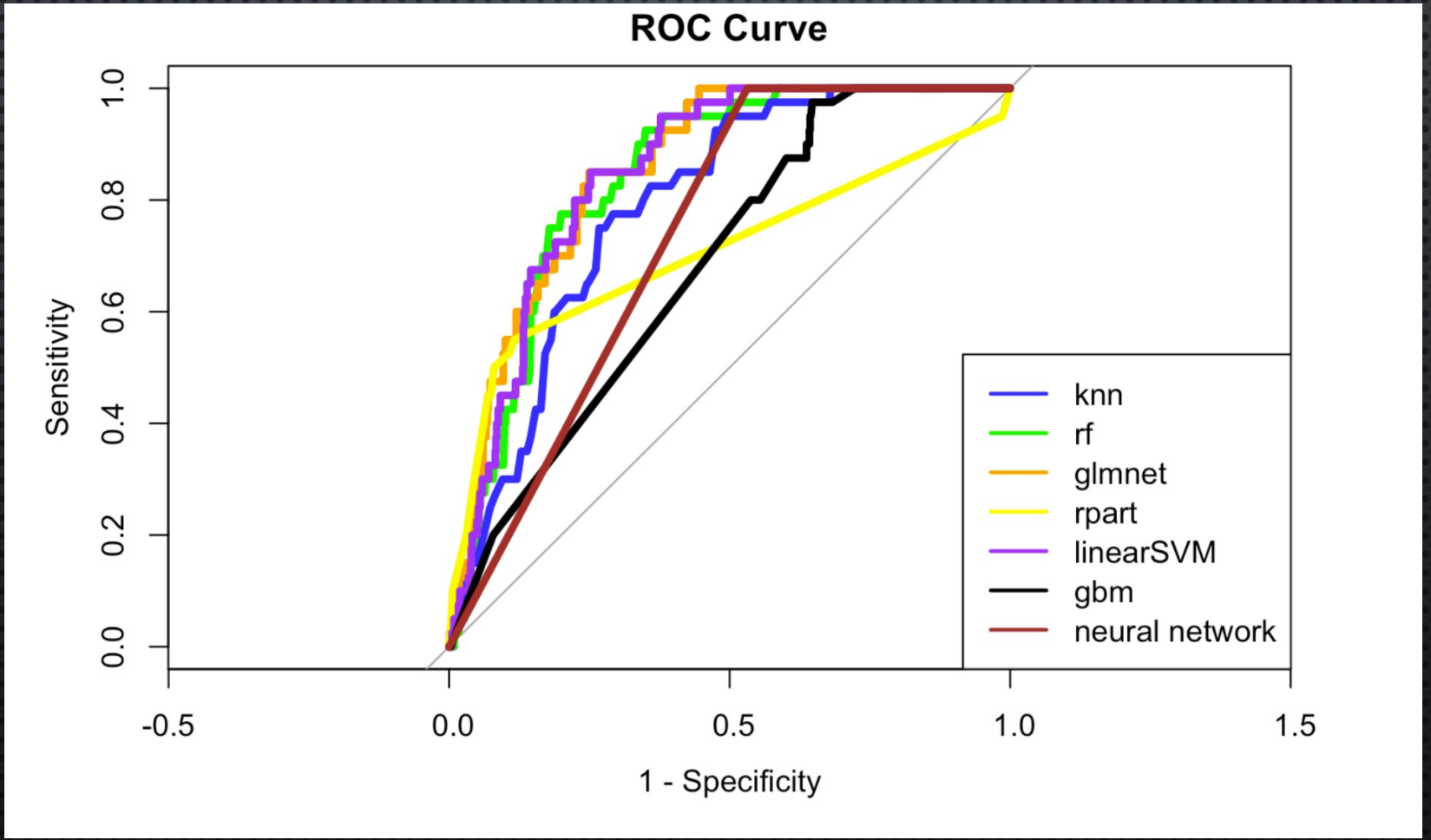
MODEL TRAINING AND EVALUATION

EVALUATION METRICS

	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
linearSVM	0.8488	0.7516	0.825
gbm	0.6796	0.003861	1
neural network	0.7342	0.4646	1

MODEL TRAINING AND EVALUATION

EVALUATION METRICS



Model Selection

Based upon the evaluation metrics selected, the Logistic Regression with Elastic Net Regularization was determined to be the better performing model

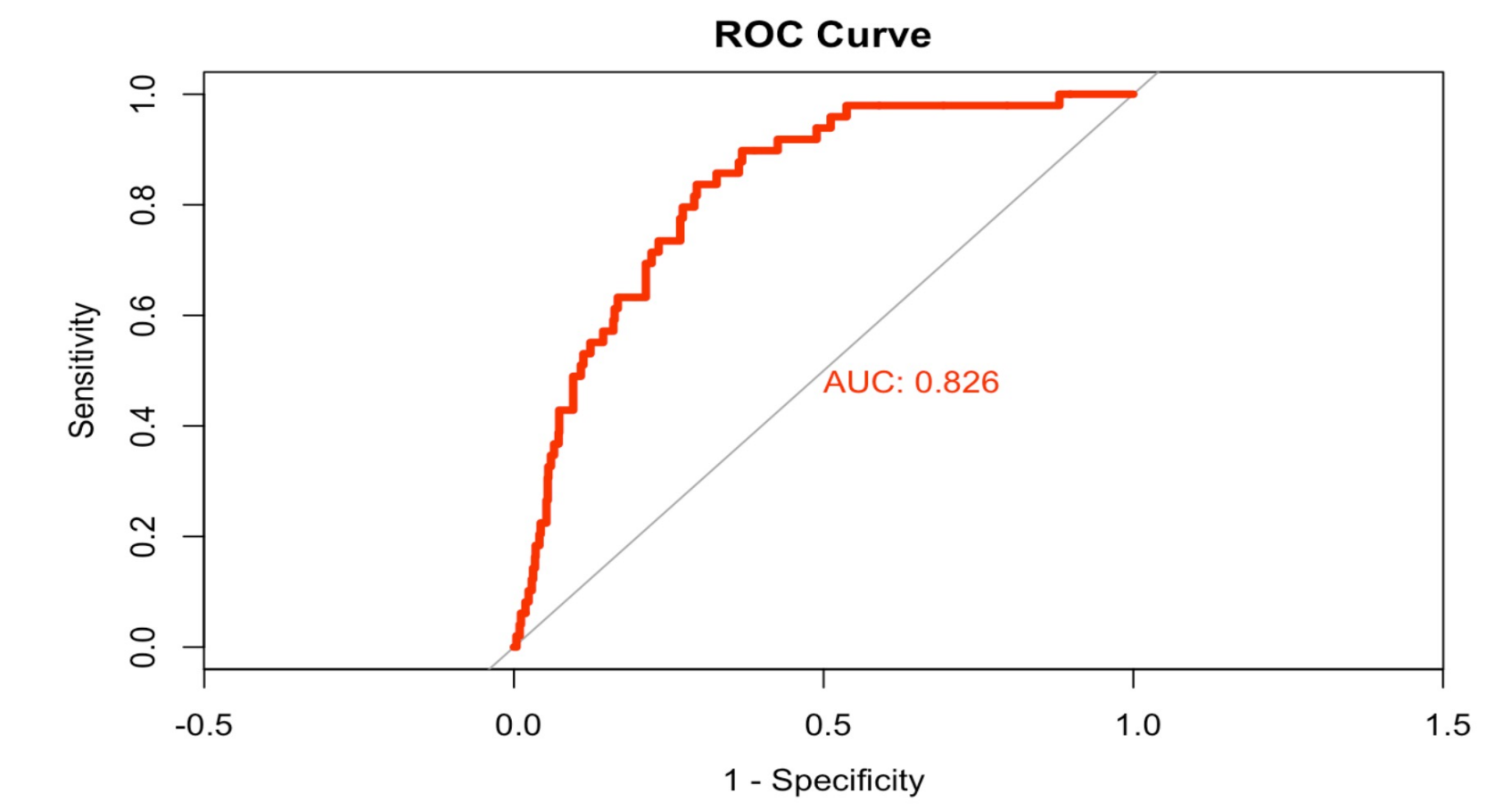
Final Model Evaluation

- Final model was built using the glmnet package
  - Alpha and lambda were set based upon the results of the cross-validation and tuning process with the caret package
  - Model was built using the fullTrain dataset
  - Model was evaluated against the previously unused test dataset

FINAL MODEL EVALUATION

```
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
```

FINAL MODEL EVALUATION



FINAL MODEL EVALUATION

VARIABLE SELECTION SUMMARY

```
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 and evaluating a variety of machine learning models for the prediction of stroke, this project found that a logistic regression model using elastic net regularization provided the best balance between sensitivity and specificity as well as providing a more easily interpretable solution.