**Stroke (Heart Attack) Prediction**

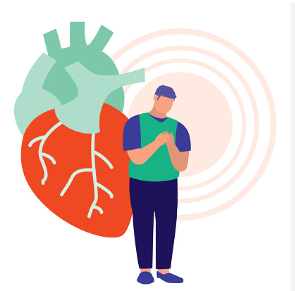
Machine Learning Final Project

Kiran Ram Ganugula

11607355, kganu2

**1 Abstract:**

Every year, heart disease, stroke, and other cardiovascular disorders claim more than 877,500 lives in the United States. The major causes of death in the US are heart disease and stroke, respectively, which rank first and fifth. This dataset is used to assess a patient's risk of suffering a stroke. To predict I used the following K-Nearest Neighbors Algorithm, Logistic Regression, Decision Tree, and Artificial Neural Network.



**2 Problem Definition and Goals:**

Based on input characteristics like gender, age, numerous diseases, and smoking status, this dataset is used to determine whether a patient is likely to get a stroke. The data rows each provide pertinent information about the patient. Our goal is to predict whether the patient is going to get a stroke or not by using the above-mentioned models.

This dataset contains 12 variables and 5000 observations, and the target variable is the **Stroke**, one of the biggest problems in the dataset is data is **unbalanced**, we need to balance the data before performing the models. In Dataset “id” are the patent id number, “Gender”, and “age” of the patent, and “Hypertension” is whether the patient had it or not. The previous history of heart disease “Ever Married” marital status of the patient, work type, Glucose level of the patent, BMI Rate, Smoking status of the patent, and **Stroke** status.

**The Variable is as follows:**

we can download the dataset from the below link.

[Stroke Prediction Dataset | Kaggle](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset)

1. Id
2. Gender
3. Age
4. Hypertension
5. Heart disease
6. Ever married
7. Work type
8. Residence type
9. Avg glucose level
10. Bmi
11. Smoking status
12. Stroke

**3 Related Work:**

We are doing the Stroke prediction on patients and it’s the largest data in the Kaggle([Stroke Prediction Dataset | Kaggle](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset)), I found other data sets on heart failure and related to heart disease but all the other datasets with fewer observations and fewer variables compared to our dataset. We find those sets from this link [Search | Kaggle](https://www.kaggle.com/search?q=heart+prediction+dataset)

Also, I get to know more information about USA heart failure information <https://www.ahajournals.org/doi/10.1161/CIR.0000000000001052> from this I understand more information about heart failure and other heart-related statistics, they mention the information clearly by using graphs and understandable way. So, it helps me understand the best way to represent the graphs and plots.

**4 Data Exploration and Pre-Processing:**

* I did the normalization and one hot encoding for ANN model to deal with my data. Because my data is imbalanced.
* My data was imbalanced data so when doing the models (KNN, Logistic Regression, ANN, etc.) I used the reference as Module 12 to deal with them like (Class weights etc...)
* There are 201 missing values in the variable “bmi” to deal with them I used the Mean and mode to deal with them, it will calculate the column mean and fill l the missing values
* Our data has 6 numeric and 6 categorical variables we converted the categorical data into factorial and the target variable “Stroke” integer into factor after that we replaced the 0,1 with “no” and “yes”
* We removed the first column called “ID” which is not important, which is patent ID.

**Data Exploration:**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* **For Variable Gender**
* We have a chi-squared value of 0.47259. Since the p-Value 0.7895 is not less than the significance level of 0.05, we cannot reject the null hypothesis and conclude that the two variables are in fact dependent.
* **For Variable Age**
* Here the output of t-test shows that the p-value is less than the significance level and we can reject null hypothesis and accept alternate hpothesis which says that difference in age means for two groups with (no stroke)0 and (stroke)1 is not equal to 0.
* **For Variable Hypertension**
* T\_test we obtained a p-value of 1.978e-09, lower than the threshold of 0.05. You can reject the null (H0) hypothesis.true difference in means between group 0 and group 1 is not equal to 0
* **For Variable Heart Disease**
* t-test we obtained a p-value of 4.095e-08, lower than the threshold of 0.05. You can reject the null (H0) hypothesis. true difference in means between group 0 and group 1 is not equal to 0
* **For Variable Ever Married**
* We have a chi-squared value of 58.924. Since we get the p-Value is 1.639e-14 less than the significance level of 0.05, we can reject the null hypothesis and conclude that the two variables are in fact dependent.
* **For Variable Work Type**
* We have a chi-squared value of 49.164. Since we get the p-Value 5.398e-10 is less than the significance level of 0.05, we can reject the null hypothesis and conclude that the two variables are in fact dependent.
* **For Variable Residence Type**
* We have a chi-squared value of 1.0816. Since the p-Value 0.2983 is not less than the significance level of 0.05, we cannot reject the null hypothesis and conclude that the two variables are in fact dependent.
* **For Variable Avg Glucose Level**
* t-test we obtained a p-value of 2.401e-11, lower than the threshold of 0.05. You can reject the null (H0) hypothesis. true difference in means between group 0 and group 1 is not equal to 0
* **For Variable BMI**
* t-test we obtained a p-value of 0.0003591, lower than the threshold of 0.05. You can reject the null (H0) hypothesis. true difference in means between group 0 and group 1 is not equal to 0
* **For Variable Smoking Status**
* We have a chi-squared value of 49.164. Since we get the p-Value 2.085e-06 is less than the significance level of 0.05, we can reject the null hypothesis and conclude that the two variables are in fact dependent.

**5 Data analysis and experimental Results:**

For this Dataset, I’m using the 4 models, for doing these models with my Imbalanced data I referred the module 12&13, in those modules we clearly got an overview of how to deal with Data Imbalance.

* Gradient Boosting Machine (Decision Tree)
* K-Nearest Neighbors Algorithm (KNN)
* Logistic Regression (LASSO, Ridge, Elastic Net)
* Artificial Neural Network (ANN)

**5.1. Gradient Boosting Machine (Decision Tree):**

For doing decision trees I use ensemble trees like GBM, I do these models by using the **class weights that** were thought in module 12, This method of controlling data imbalance is cost-sensitive and involves giving instances belonging to the minority vs. the majority class distinct weights. then we retrain the model with class weights and later we compute Area under the ROC Curve.



Now we predict the confusion matrix for the Gradient Boosting Machine (Decision Tree), with that can examine the model performance.

Text

Description automatically generated

**5.2. K-Nearest Neighbors Machine (KNN):**

The amount of data we are producing has increased the requirement for sophisticated machine learning algorithms. The K Nearest Neighbor algorithm is one such technique. Again, for this model also I’m using I used module 12 to predict the imbalanced data.

Graphical user interface, application

Description automatically generated with medium confidence

Now we predict the confusion matrix for the K-Nearest Neighbors Algorithm (KNN) that can examine the model performance.

Text

Description automatically generated

**5.3. Logistic Regression (LASSO, Ridge, Elastic Net):**

Lasso, Ridge, and Elastic Net can be used to regularize a logistic regression model by penalizing big coefficients, much like linear regression. Caret's train function allows us to perform regularized logistic regression by setting method="glmnet"

In this logistic regression, we have 3 types of models to predict.

* LASSO
* Ridge
* Elastic Net

**5.3.1. LASSO Regression:**

In this, we put the tune lambda (we use a grid with 100 values between -3,3) and set alpha to 1 to perform lasso.

Graphical user interface, text, application, email

Description automatically generated

After implementing the LASSO now, we do the prediction by using the prediction function, with that we can do Confusion Matrix.

With that data we can find the accuracy of model.

Graphical user interface, text

Description automatically generated

**5.3.2. Ridge Regression:**

As in the Ridge linear regression model, to do Ridge regularization, we set alpha=0 and we put the tune lambda (we use a grid with 100 values between -3,3)

Graphical user interface, text, application, Word

Description automatically generated

Same as LASSO, after implementing the ridge now, we do the prediction by using the prediction function, with that we can do Confusion Matrix.

With that data we can find the accuracy of the model.

Graphical user interface, text

Description automatically generated

**5.3.3. Elastic Net:**

As in the Ridge linear regression model, to do Ridge regularization, we set alpha=0 .

Graphical user interface, text, application, email

Description automatically generated

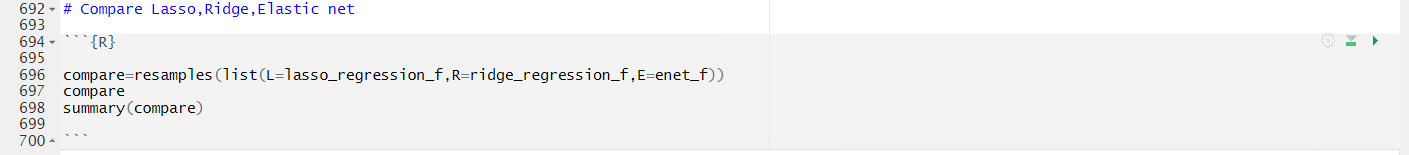
Same as LASSO, after implementing the Elastic Net now, we do the prediction by using the prediction function, with that we can do Confusion Matrix.

With that data we can find the accuracy of the model.

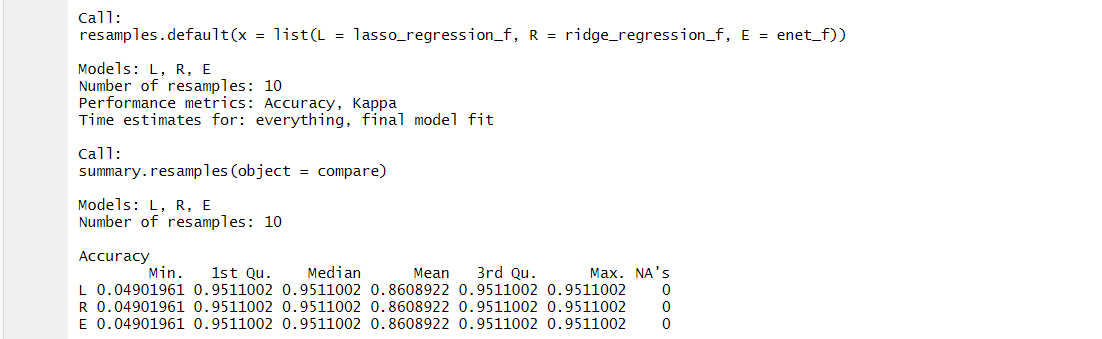
Text

Description automatically generated

**5.3.4. Now we compare the three logistic models LASSO, Ridge, Elastic Net**



By using the resamples function we can compare all the three logistic regression models.



In the above after we are comparing, we got that Min, Median, Mean, Max of all the Lasso, Ridge, and Elastic Net.

**5.4 Artificial Neural Network (ANN):**

**5.4.1. Artificial Neural Network (ANN) without Class Weights:**

To predict strokes, neural network models with and without class weights will be developed. We need to normalize the numerical variables in the stroke dataset before preparing it for neural networks.

Our Data is imbalanced data so for that we are doing it by reference to Module 12 about dealing with imbalanced data to create the ANN Model.

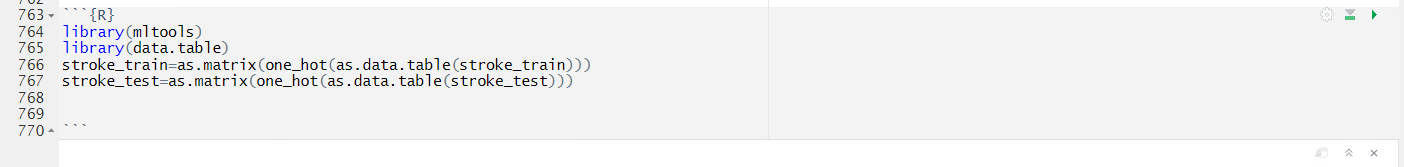
A picture containing chart

Description automatically generated

After we normalize, the data now do need to convert the target variable “Stroke” to 0-1 indices



And now we need we need to one-hot encode the factor variables



Now let’s separate features(x) and labels (y). The labels are the “Stroke” variable, the 11th column in Stroke Train and Stroke Test.



Create a neural network model with two hidden layers and configure it for binary classification.

Graphical user interface, text, application

Description automatically generated

**Graphs:**

We got the graphs as follow below; we got the most accurate way of neural network graphs because. We have imbalanced data,

Timeline

Description automatically generated

Table

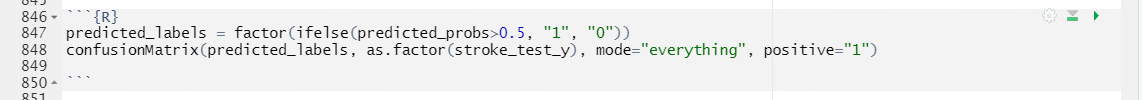
Description automatically generated

Now we do the ANN Model Predictions:

Graphical user interface, text, application

Description automatically generated

Now we do the confusion Matrix to the Predicts:



Text

Description automatically generated

**5.4.2. Artificial Neural Network (ANN) without Class Weights:**

Here we are doing with the class weights, we did the class weights when we did that in the Decision Tree (Gradient Boosting Machine) model, and we are going to use the same now here.

Text

Description automatically generated

To use class\_weights in keras, we can use the parameter “class\_weight” in “fit” function and set it to a list of weights for each class.

Graphical user interface, text, application

Description automatically generated

After that, we got the graph below

Graphical user interface

Description automatically generated

**When comparing with the ANN without class weights graph, the ANN with graph weights got more accurate.**

Tuning R File code

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Obtain the model's predictions on the test data and compute the weighted model's area under the ROC curve.

Graphical user interface, text, application

Description automatically generated

Compute Confusion Matrix:

Text

Description automatically generated

**When comparing with the ANN without class weights graph, the ANN with graph weights got more AUC Value**

**Conclusion:**

Predicting the Stroke with this Imbalance data is more difficult than compared to balanced data, we need to do a lot of work on the data to make it balanced. This project needed a thorough review of the exploratory data and numerous decisions regarding the shrinking of high cardinality predictors and missing values.

In this top 2 most AUC is both the “ANN” and “ANN with class weights” the value of Most AUC is ANN with class weights. (0.9991). we got the least AUC in the Decision Tree which is 0.714. because of the imbalance we got most of the accurate values.

In the Future, I want to do my ML skills to prevent crime in the USA, we can collect data on crimes in various locations and do predictions about the season or certain time periods of crimes and use our predictions to prevent them or stop the crime by using this analysis. If we do like this maybe one day we can find #NoCrime society.

**Comparing All the models**

**While comparing all the models I got the most accuracy "ANN With Class Weights" which is 0.9971**

**the accuracy of GBM Decision tree is 0.714**

**The Accuracy of KNN is 0.7757**

**The Accuracy of LASSO is 0.952**

**The Accuracy of Ridge is 0.932**

**The Accuracy of Elastic Net is 0.923**

**The Accuracy of ANN is 0.996**