Brain stroke Prediction with Logistic Regression

Statistical Methods-III

Rudrashis Bardhan (BS2118)
Purushottam Saha (BS2119)

Indian Statistical Institute

November 22,2022

Introduction

In this project, we have fitted a Logistic Regression model on Brain stroke data set based on covariates as age, average glucose level, heart disease, hyper-tension, BMI and a few predictors, some of which are multi-class; namely work type and smoking status. We have predicted the stroke (a binary response) on the basis of our covariates.
However our dataset had a problem of imbalance (only 5 % of our data had stroke patients). In order to tackle that, we have applied some imputations, sampling and threshold selection techniques to be discussed in due course.

Data Description

The following are the twelve covariates in our data.

- id: unique identifier
- gender: "Male", "Female" or "Other"
- age: age of the patient
- hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
- heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
- ever_married: "No" or "Yes"
- work_type: "children", "Govt_job", "Never_worked", "Private" or "Self-employed"
- Residence_type: "Rural" or "Urban"
- avg_glucose_level: average glucose level in blood (in mg/dl)
- bmi: Body Mass Index
- smoking _status: "Formerly smoked", "Never smoked", "Smokes" or "Unknown"

Exploratory Data Analysis

Exploratory data analysis is always the first step to any kind of statistical observation which will help us to understand and correlate the findings. We perform a basic exploratory data analysis on our data, to observe some features of the data set and the distributions. It is to be noted that 0 holds for female and 1 holds for male. (The only "Other" data point was removed in the data cleaning process)
Figure-1 shows that the aged people are more likely to have stroke in our dataset which is expected from intuitive sense.

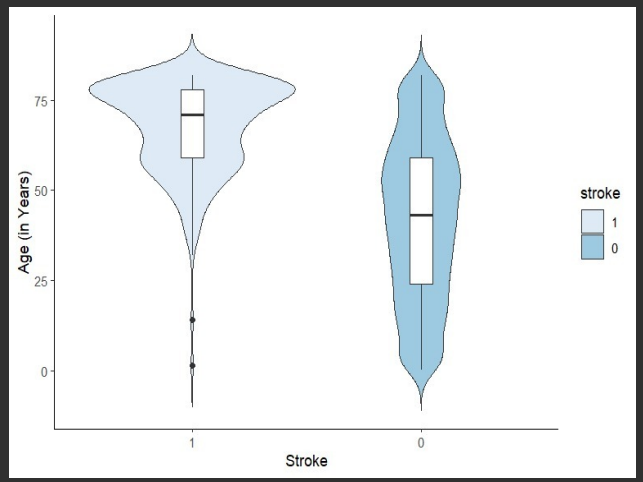


Figure: Age vs stroke

Exploratory Data Analysis

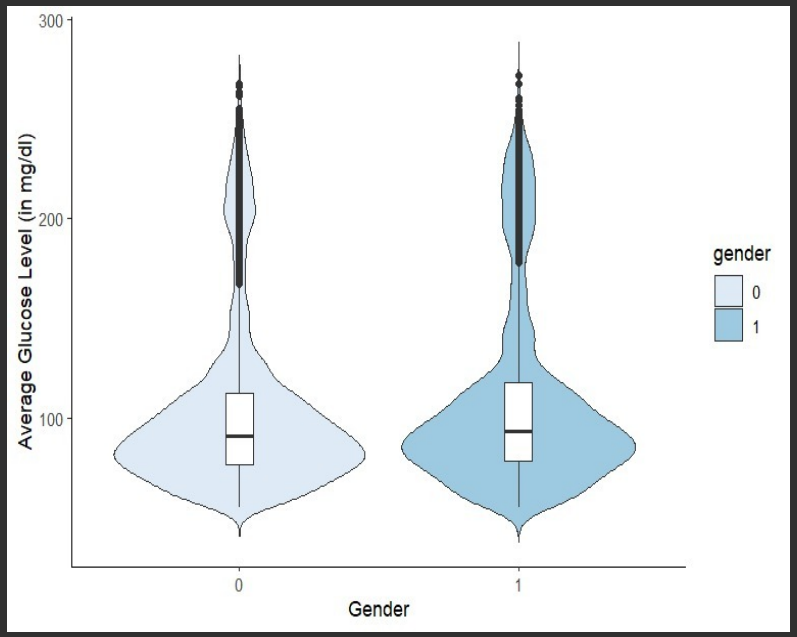
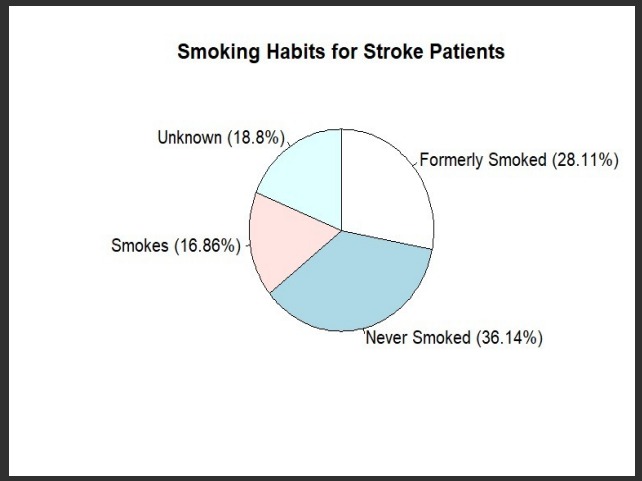


Figure: Average glucose level vs gender

Figure-2 provides insights about the fact that most of the Male patients' average glucose levels fall in nearly the same range as that of the females with the distributional average glucose levels for males being slightly higher than that of the females.

Exploratory Data Analysis



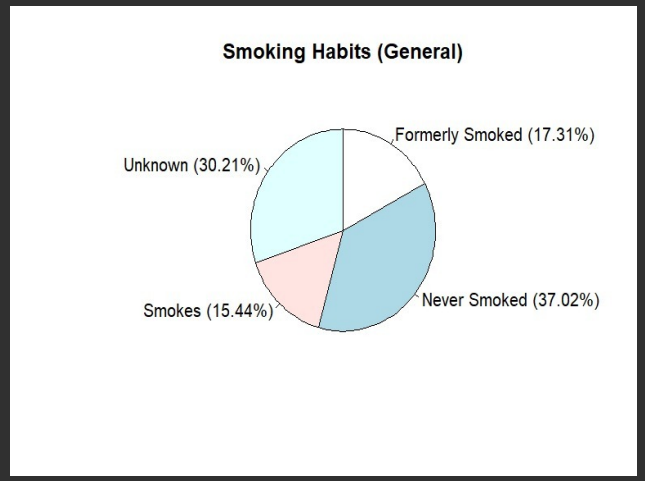


Figure: Smoking habits for stroke patients vs general people

Figure-3 shows that people who smoke or used to smoke are percentage wise more in the category of stroke patients than on an average the general population, implying Smoking habits do influence Stroke possibilities.

Exploratory Data Analysis

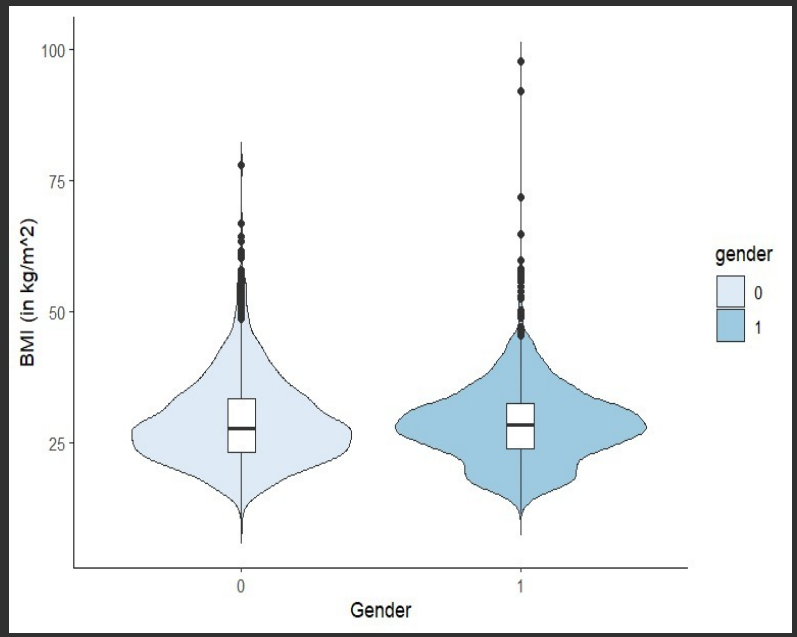


Figure: BMI vs gender

Figure-4 shows that people from both genders have nearly equal distribution of BMI, with a little difference in the extreme cases and less BMI patients.

Data Cleaning

We had only one data point (patient), whose gender was mentioned as "Other", which we removed.

For some patients, the BMI was as high as 75, which was pretty ridiculous (not impossible, of course, but not a good data point for a linear model like Logistic Regression, which we are going to fit). So we removed 13 data points for which, BMI was greater than 60. (The last class for BMI is ≥ 45)

After this, we had many more missing values in BMI and Smoking habits (as Unknowns), but removing those data points would be a huge loss. For example, out of 201 missing values in BMI, 40 patients had a stroke. So we cannot just omit those data points.

Data Cleaning

First, we observed that, smoking habits were unknown for even some children (mentioned as work type) also. With the basic assumption that children generally do not smoke, we assigned "never smoked" to those missing values.

For the rest of those missing values, we used "mice" library from R to impute them. For BMI, Predictive Mean Matching Model (pmm) was used, and for Smoking Habits, Polyreg model (polr) was used.

With this steps followed, we had a clean data set with no missing values to proceed with fitting a Logistic Regression model.

Problem with the Imbalance of Data

In several Binary Classification problems, the two classes are not equally represented in the dataset. When one class is under-represented in a dataset, the data is said to be unbalanced. In such problems, typically, the minority class is the class of interest. Having few instances of one class means that the model is often unable to generalize the behavior of the minority class well, hence it performs poorly in terms of predictive accuracy - making it difficult to both train and test a model on a unbalanced data.

Recovering Imbalance of Data

A common strategy for dealing with unbalanced classification tasks is to under-sample the majority class or over-sample the minority class in the training set before learning a classifier. By removing majority class instances, the training set is artificially re-balanced. Similarly, over-sampling the minority class balances the training set artificially.

However both have their advantages and disadvantages. Under-sampling causes a loss of information while over-sampling causes over-fitting of data.

Instead of random over-sampling or under-sampling, we use a technique called SMOTE (Synthetic Minority Oversampling Technique), where unlike exactly over-sampling the minority class, we add some random noise to the over-sampled point, to make it a not exact copy of any of previous data points, resolving the issue of duplicates.

Applying SMOTE to our data

To start with, we separate the training and the testing data in a 60:40 ratio. (Reason for such a big testing data is so that it has a few positive stroke data points to strongly validate the model.) Here, we also take care about the even-ness of the distribution of the positive stroke cases in training and testing data sets, so we separately sampled from positive and negative stroke cases, and then combined to get the full training and testing data sets.

Note : From now, where-ever "in avg" is mentioned, it means that, we have set our seed to 151, and then got 100 training-test splits and then averaged the quantity mentioned, otherwise the inference is done on a random instance of split.

After this initial procedure, we had only 4.88 % positive stroke data (in avg) in our Training data.

Fitting a model with this unbalanced data will result in neglecting most of the positive stroke cases, failing our target : Classifying the Stroke cases! So, here we applied SMOTE algorithm to balance the dataset. After applying the algorithm, the positive stroke cases went up to 49.37 % (in avg) of our training data, which is obviously balanced to fit our model.

Model : Logistic Regression

Introducing Dummy Variables

We still are not fully ready to train our well-known, Logistic Regression model, as we have a few covariates that are multi-class categorical variables. To use them, we need to introduce **Dummy variables** for them, which are basically indicator variables, for each class of the categorical covariate (one less for the fact that at least one class is chosen in the variable). For example, we have 3 categories in Smoking status, namely : Previously smoked, Never smoked and Smokes. So we need 2 indicator variables to encode this categorical covariate by indicating if a person previously smoked, or (the another indicator) smokes or not. Both of them cannot be 1, but both of them can be 0 if the person never smoked. Similar approach was taken for Work type, to make indicator for children, self employed, govt job and private jobs : 0 in each of the four indicators imply the "Never worked" class.

Logistic Regression : Model 1

Now we fit a logistic regression model to our data with all the covariates, 13 in total (including dummy variables). The results (one instance, due to randomness of sample) are shown below.

```
Call:
glm(formula = stroke ~ gender + age + hypertension + heart_disease +
    ever_married + work.private + work.self_emp + work.govt_job +
    Residence_type + avg_glucose_level + bmi + form_smokes +
    smokes, family = binomial(link = "logit"), data = train_new)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5834  -0.6224  -0.1419   0.7062   2.6968

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -5.1639689  0.3644623 -14.169  < 2e-16 ***
gender            0.1562894  0.0780181   2.003  0.04515 *
age               0.0880727  0.0029231  30.130  < 2e-16 ***
hypertension      0.5531360  0.1088763   5.080 3.77e-07 ***
heart_disease     0.6227846  0.1309086   4.757 1.96e-06 ***
ever_married     -0.2255674  0.1219954  -1.849  0.06446 .
work.private     -0.6665265  0.3814035  -1.748  0.08054 .
work.self_emp    -1.1801202  0.3977438  -2.967  0.00301 **
work.govt_job    -1.0451553  0.3926994  -2.661  0.00778 **
Residence_type    0.1480505  0.0767507   1.929  0.05373 .
avg_glucose_level 0.0032957  0.0006825   4.829 1.37e-06 ***
bmi               0.0042158  0.0060879   0.692  0.48863
form_smokes       0.4118977  0.0927143   4.443 8.89e-06 ***
smokes            0.5150438  0.1041597   4.945 7.62e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7998.1  on 5769  degrees of freedom
Residual deviance: 5185.7  on 5756  degrees of freedom
AIC: 5213.7

Number of Fisher Scoring iterations: 6
```

Figure: Output of first logistic regression model

Model 1 : Inferences

The model has some interesting observations to point out, some of them are listed below.

'Age' is the most significant covariate, as expected, because from general intuition it follows that old aged people are more prone to strokes.

Smoking indicators have the highest positive coefficients and also very significant covariates, indicating smoking indeed strongly increases the chance for brain stroke.

Another interesting result was observed as the work type indicators all had negative coefficients, so if all the indicators were 0, i.e. the subject never worked, he/she has the most probability to have a brain stroke, fixing other covariates.

Model 1 : Inferences (Continued)

Hyper-tension and heart disease indicators were also quite significant covariates with positive coefficients, implying they indicate towards more chance of brain stroke.

Average glucose level is also a very significant covariate, with a very close to '0' positive coefficient, expected as the average glucose value has mean around 106 mg/dl.

BMI has the highest p-value (0.488), which says it has no significance in this model, which in turn signifies the need for a better model, atleast based on p-values.

Gender and ever-married also has lesser p-values, so a better model withdrawing these 3 covariates can be considered. (One work type indicator can not just be removed, as it may affect the interpretation of the other indicator covariates from work type.)

Insights from the Model : What does the Metrics tell us

The confusion matrix and other metrics are given by :

	Actual 0	Actual 1
Predicted 0	TN	FN
Predicted 1	FP	TP

Accuracy is defined as $\frac{TP+TN}{TP+TN+FP+FN}$

Sensitivity is defined as $\frac{TP}{TP+FN}$

Specificity is defined as $\frac{TN}{TN+FP}$

Insights from the Model 1 : What does the Metrics tell us

The confusion matrix and other metrics are given by (threshold = 0.5)

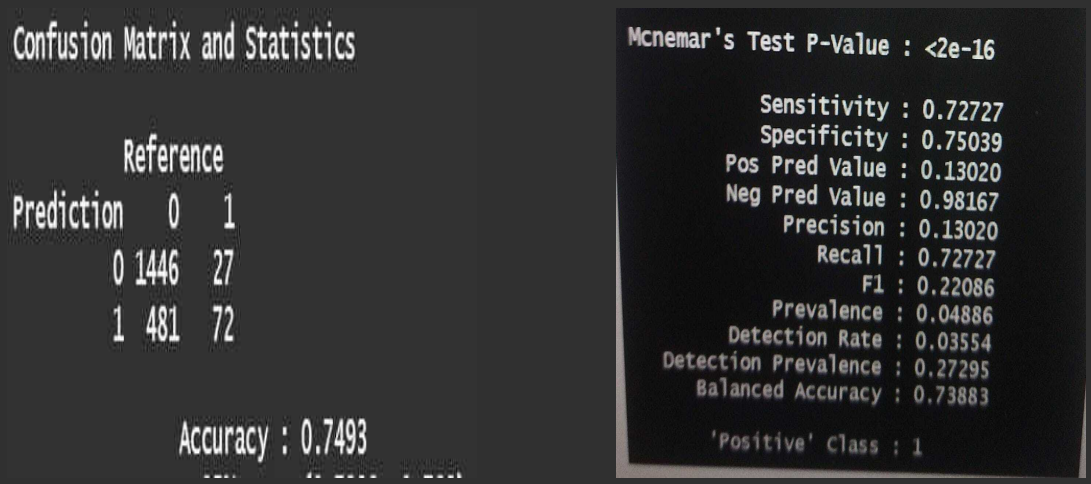


Figure: Confusion matrix and related metrics of first logistic regression model

The accuracy comes out to 74.93 % and the sensitivity and specificity are 72.727 % and 75.039 % respectively are okay to start with. The average accuracy turns out to be 75.12%, while average sensitivity turns out to be 77.77%.

Logistic Regression : Model 2

From the first model, we observed the "not-so-significant" covariates as gender, bmi and ever-married, for our model. So we removed those covariates, in wish to get a better model. The summary of the model is given below. (one instance)

```
Call:
glm(formula = stroke ~ age + hypertension + heart_disease + work.private +
    work.self_emp + work.govt_job + Residence_type + avg_glucose_level +
    form_smokes + smokes, family = binomial(link = "logit"),
    data = train_new)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6288  -0.6242  -0.1451   0.7102   2.6879

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -4.9984657  0.3425514 -14.592  < 2e-16 ***
age               0.0866486  0.0028126  30.807  < 2e-16 ***
hypertension      0.5576426  0.1077969   5.173 2.30e-07 ***
heart_disease     0.6686289  0.1298553   5.149 2.62e-07 ***
work.private     -0.7671159  0.3668693  -2.091 0.036530 *
work.self_emp    -1.2868036  0.3841258  -3.350 0.000808 ***
work.govt_job    -1.1590013  0.3793744  -3.055 0.002250 **
Residence_type    0.1517480  0.0766226   1.980 0.047652 *
avg_glucose_level 0.0034600  0.0006453   5.362 8.24e-08 ***
form_smokes       0.4122729  0.0910839   4.526 6.00e-06 ***
smokes            0.5069546  0.1037867   4.885 1.04e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7998.1  on 5769  degrees of freedom
Residual deviance: 5193.2  on 5759  degrees of freedom
AIC: 5215.2

Number of Fisher Scoring iterations: 6
```

Figure: Output for second logistic regression model

Model 2 : Inferences

As we observe the summary, we see that nothing much has changed, than a few subtle changes like :

Now every covariate is quite significant.

Coefficients for heart disease has gone a bit up, proning towards more chance to have a stroke, if you have heart disease.

Coefficients for work type indicators has decreased more, giving stronger supports to our explanation in model 1.

Model 2 : Confusion matrix and metrics

The confusion matrix and other metrics are given by

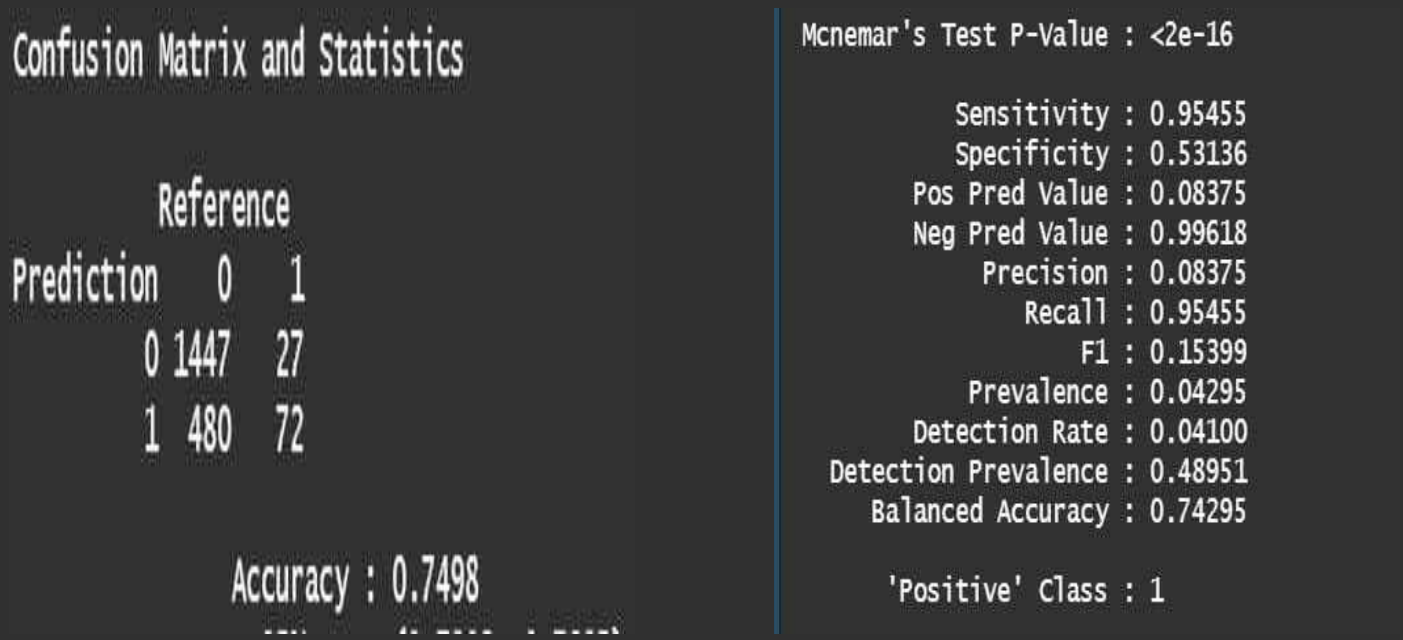


Figure: Confusion matrix and related metrics of second logistic regression model

Again, only a slight increase in accuracy in this instance, but avg accuracy is 75.10%, quite similar to our first model, and avg sensitivity of 78.58%, slightly better than our first model. However our model remains more or less the same.

Is the second model any better then ?

For the most part, they are same.

Still if we want a winner, we can do so by checking AIC, less AIC implies better model.

AIC value for (same sample) model 1 was 5213.7, where for the second model it is 5215.2, even higher AIC after reducing covariates.

So the first model works better, so further works are done on the basis of model 1.

Threshold Value: is there a better alternative than 0.5?

Till now, every computation of confusion matrix and the metrics like accuracy, specificity etc, are done choosing the threshold value as 0.5, which is the default value. But the confusion matrix and hence those metrics varies, if we vary the threshold value.

So the question arises that, which threshold value should we choose? What metric do we want to maximise?

Can we trust Accuracy ? What do we want ?

As our data is very imbalanced (only 5% data points from the minority class), if we just predict that every data point belongs to majority class, the accuracy will still be very high, even close to 95%. But that model will be of no use, as we want to predict those minority points. So we see that accuracy is not the useful metric here.

We want our model to predict all (or atleast most) positive stroke cases as positive. As from a medical test point of view, predicting a possible stroke patient as safe is far worse than predicting a less possible stroke patient as serious. So we want to maximise our sensitivity (TP/TP+FN).

Maximising Sensitivity

To maximise the sensitivity, we follow this idea :

For all the positive stroke cases in the test data, we compute the probabilities (rather predict the probabilities) based on our model, and take their minimum as our threshold. In this way, every positive stroke case is predicted truly. Though, we can allow a 5% error margin, in the sense, we can allow 5% of the positive stroke cases in test data, to be predicted as negative, so as to increase the other metrics like accuracy and specificity. So we choose the n * 0.05 th minimum value of those predicted probabilities, if n is the number of positive stroke cases in test data.(FP+TP)

Now the cutoff varies as n varies, hence we allow the process 100 times and take the mean threshold, which is 0.188, for which we get our average accuracy of 52.89%. Though the average accuracy is not good enough, the cutoff atleast satisfies the first priority : 95% sensitivity which as said before shall be our more important criterion to diagnose brain strokes.

Conclusion

So to summarise as a whole, smoking really increases the risk of brain stroke as a whole and also the presence of hyper tension and heart diseases are red flags for brain stroke. The average glucose level can also be considered a significant reason for increased risks of brain stroke. An interesting observation is that had all the work type indicators been zero (that is subject never worked) has the maximum probability of brain stroke.

There is also a trade off between accuracy and sensitivity as we have observed previously by deciding the optimal cutoffs. However, since we prioritise on the sensitivity more than accuracy, hence decrease in accuracy does not impact our model that much.

Questions

Thank You

Appendix : SMOTE

SMOTE algorithm does not over-sample randomly the minority class points, however it synthesises data points close to original, but not exact. First a total no. of oversampling observations, N is set up. Then it starts iterating by first selecting a positive class (minority class) instance (or data point) at random. Next, the KNNs (K nearest neighbours) for that instance is obtained. At last, N of these K instances is chosen to interpolate new synthetic instances. To do that, using any distance metric the difference in distance between the feature vector (or the tuple of covariate values) and its neighbors is calculated. Now, this difference is multiplied by any random value in (0,1) and is added to the previous feature vector as a synthetic data point. Hence, it is used to obtain a synthetically class-balanced or nearly class-balanced training set, which is then used to train the model. SMOTE can be expressed as:
s = x + u(x^R - x) where $0 < u < 1$ and x^R is selected among the chosen K nearest neighbours of x and x is a minority class sample.

Appendix : Null Deviance and Residual Deviance

Null deviance is defined as
2(LL(Saturated Model) - LL(Null Model)) on $df = df_{sat}$ - df_{null}.

Residual Deviance is defined as
2(LL(Saturated Model)- LL(Proposed Model)) on $df = df_{sat}$ - $df_{proposed}$.

Saturated model is a model that assumes each data point has its own parameters(so there are n parameters to estimate).

Null Model is a model that assumes only one parameter for all its data points.

The proposed model explains our model with p parameters + intercept term so (p+1)parameters in all.

So, if Null deviance is small, it explains the null model well, and if we wish to compare our null model with proposed model, we can look at (Null deviance - Residual deviance) which follows χ^2 distribution with $df_{proposed}$ - df_{null} = p degrees of freedom.

Appendix : AIC

AIC (Akaike Information Criterion): Suppose that we have a statistical model of some data. Let k be the number of estimated parameters in the model. Let L be the maximized value of the likelihood function for the model. Then the AIC value of the model is the following:

$$AIC = 2k - 2ln(L)$$

Thus, AIC rewards goodness of fit (as assessed by the likelihood function), but it also includes a penalty that is an increasing function of the number of estimated parameters.The penalty discourages overfitting, which is desired because increasing the number of parameters in the model almost always improves the goodness of the fit.

Appendix : Plots : Threshold vs Metrics

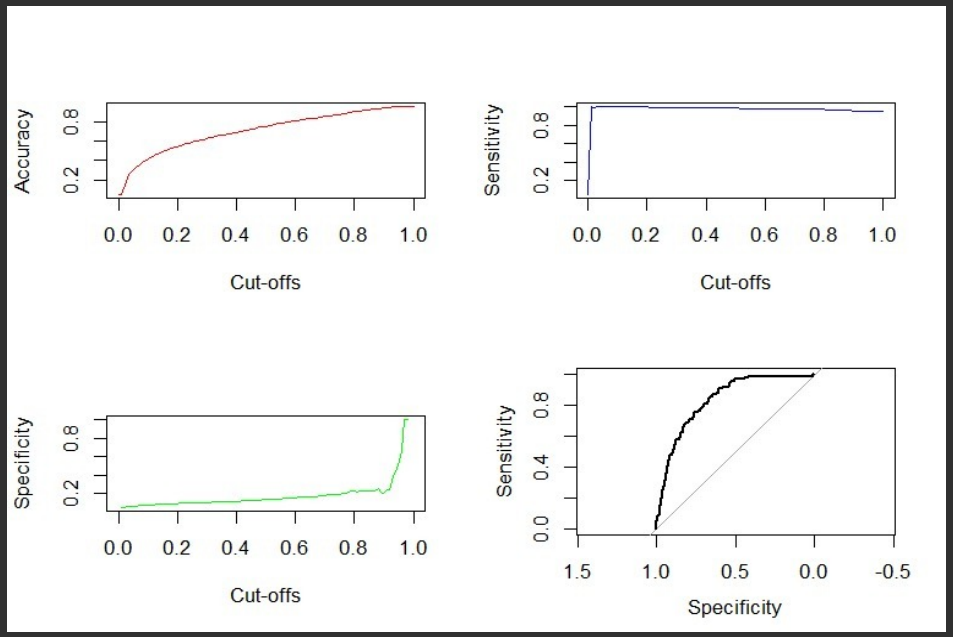


Figure: Related metrics and their relation with cutoffs

For a particular instance of training and testing data, the fitted model showed these relations in between metrics and threshold values. The last plot is special, and is called the ROC plot, is a measure of the classification algorithm, or more specifically the area under the curve (AUC) is. For this setup, the value was 0.845.

Appendix : Density plot for thresholds based on 95 % sensitivity

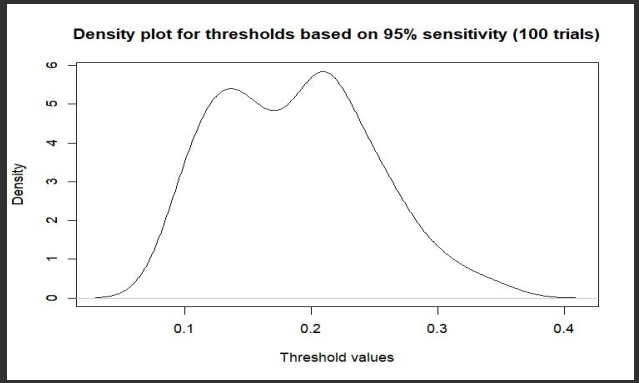


Figure: Density plot for thresholds based on 95 % sensitivity

By setting the seed to 151, we tried to have 100 random training and testing sample pairs, trained our model with them and based on that 95% sensitivity criteria, we got 100 values for thresholds, this is the density for that variable, estimated by kernel estimation method. The 2 modes give a interesting flavour to the density. After all, the mean which is 0.188, is chosen as the final threshold for our purpose.