Brain Stroke Prediction with Logistic Regression

Purushottam Saha (BS2119)
Rudrashis Bardhan (BS2118)

Statistical Methods III : End-semester Project

Abstract

In this project, we fitted a Logistic Regression model on Brain stroke data set based on covariates like - age, average glucose level, hyper-tension (0 or 1), smoking status and many more. The data was very unbalanced, with only 5% positive stroke patients in the dataset, hence re-sampling techniques like SMOTE and approximating null values by different methods were used to make the data usable. The optimal threshold value for this model is also discussed, based on Sensitivity and related metrics. ROC and AUC are also introduced as a goodness criteria for the classification.

1 Data Overview

Repository location The dataset has been collected from Kaggle and the associated link is https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

Brief Description about the Data 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"
- stroke: 1 if the patient had a stroke or 0 if not

Note: "Unknown" in smoking_status means that the information is unavailable for that patient

We will be predicting the response Stroke, which is a binary response (0 or 1) on the basis of the rest of the covariates.

2 Exploratory Data Analysis

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.**

Figure-1 shows that the aged people are more likely to have stroke which is expected from intuitive sense.

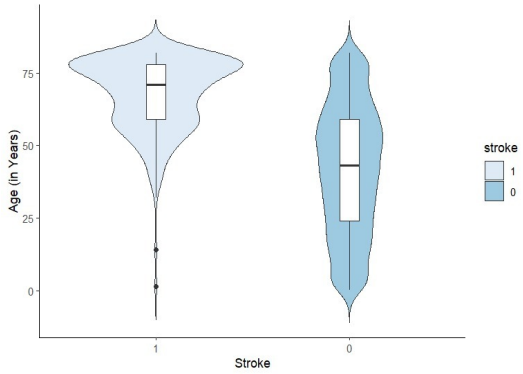


Figure 1: Age vs Stroke

Figure-2 provides insights about the fact that most of the Male patients' average glucose levels fall in between 80 to 100 and most of the Female patients' average glucose levels fall in between 70 to 100.

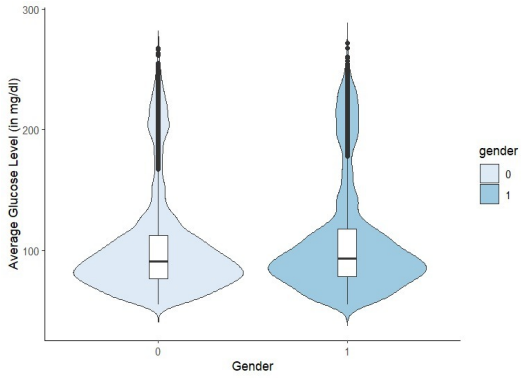
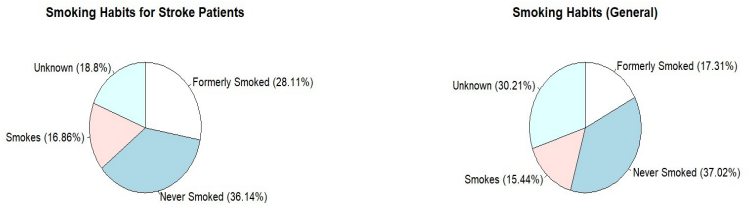


Figure 2: Average glucose level vs Gender

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 a Stroke.



(a) Smoking for stroke patients (b) Smoking for general people

Figure 3: Smoking habits for stroke patients vs general people

. **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.

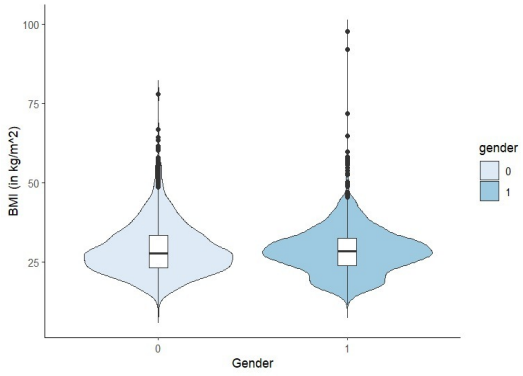


Figure 4: BMI vs gender

3 Data Cleaning

The data had some anomalies and some missing data, for which Data Cleaning and Estimating missing data was performed, as below. In the very beginning, the id column was removed, as it had no special value as a covariate, atleast as per our method of analysis. Only one "Other" gender type was in the data, and that data point was removed for simplifying the data, just a bit. There were some BMI values greater than 75, which is exceptionally big values and can not be considered in a linear model which we are going to implement, hence data point with BMI values greater than 60 ($\geq$ 45 is the last group for BMI) are removed. There were some "Unknown" data points (Approximately 30.21%) in Smoking Habits, and some "N/A" data points in BMI (Approximately 4%). These groups could not be removed, as they contained a major pool of people having Brain stroke. A significant part of the "Unknown" group in the Smoking habits had Children as 'Work Type', so by valid assumption (that generally children do not smoke), they were set to "Never smoked". For the rest part, imputation within the data was done such that the distribution and mean do not change for those covariates. For this procedure, module "mice" was used for two type of imputations : 1. Proportional odds model (polr) for smoking_status and, 2. Predictive mean matching (pmm) for BMI. After this procedure, the data was complete in the sense that it had no more na values.

4 Problem of Imbalance

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

learning algorithm is often unable to generalize the behavior of the minority class well, hence the algorithm performs poorly in terms of predictive accuracy . 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. The assumption behind the under-sampling strategy is that in the majority class there are many redundant observations and randomly removing some of them does not change the estimation of the within-class distribution. If we make the assumption that training and testing sets come from the same distribution, then when the training is unbalanced, the testing set has a skewed distribution as well. 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. There are several algorithms for both, however we will be using an algorithm namely SMOTE.(**Synthetic Minority Oversampling Technique**)

SMOTE Algorithm: This algorithm helps to overcome the over-fitting problem posed by random over-sampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together.
At first the total no. of oversampling observations, N is set up. Our goal is to ensure that the minority class and majority class get equal representation .Then the iteration starts by first selecting a positive class instance at random. Next, the KNN's 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 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.

Applying SMOTE algorithm on our data Our data is heavily imbalanced, minority class consist of only (5.4%) of the whole data set. Without handling this issue, we would have get almost no minority class in test set. For this reason, we had to deal with this issue in, first a brute forced way, to separate the stroke cases and then randomising splitting into Training data and Testing data in 60:40 ratio. (If we took even less test data points, it would have even less positive stroke data in the Test Data set.

After this procedure of separating, we had only 5.28% positive stroke data 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 48.68% of our training data, which is a good starting point to fit the model.

5 The Model

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-nary 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.

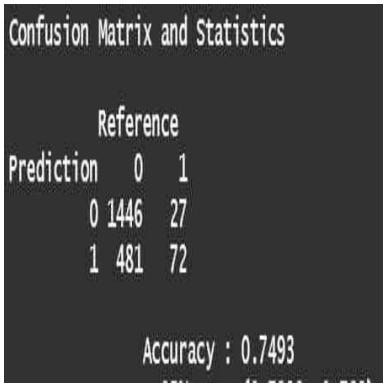
Logistic Regression Now we are in a stage to fit the Logistic Regression to our training data. We choose all the covariates (a total of 13) in the data (except the Id) as predictors, and we get the result:

Here we observe that our predictors like hypertension, heart-disease, age, references of former smoking, average_glucose level as well as employment types are our important predictors according to the model for stroke prediction, where gender, marital status and BMI are not that good predictors for this model.

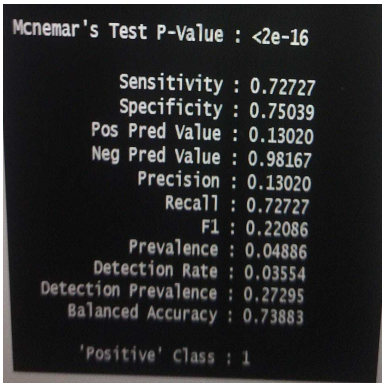
The **confusion matrix** and associated information of accuracy,specificity,sensitivity for our model is represented below.

```
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 5: Output of our 1st logit model



(a) Confusion matrix of our predicted 1st model



(b) Other related metrics

Figure 6: Confusion matrix and related metrics of 1st predicted logit model

As we can see, we have an accuracy of 74.93%, which is not that great, but it is predicting 72/99 True positive case true, which is a good start. (The cut-off or threshold value is set to be 0.5)

5.1 Logistic Regression Model 2

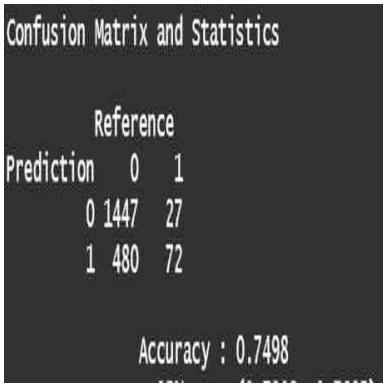
Variable Selection using p-value significance level

We saw earlier that marital status, BMI, gender are not that good predictors for the model, so we removed them, to get a better model. The summary of the new model is attached below.

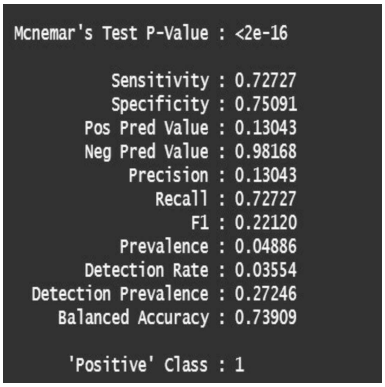
The Confusion Matrix and associated information of accuracy,specificity,sensitivity for our second Logistic Regression model is shown below.

```
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 7: Summary of our predicted Logistic model after removing some predictors



(a) Confusion matrix of our second Logistic model



(b) Other related metrics

Figure 8: Confusion matrix and related metrics of our Logistic model

6 Prediction : The Optimal Cut-off

Confusion Matrix In any machine learning model, we usually focus on accuracy. But if we are dealing with a classification problem, we also need to worry about the percentage of correct classification and misclassification. So we need a mechanism that not only provides accuracy but also helps in estimating correct classification and misclassification. The Confusion Matrix serves this purpose. It is an 2x2 matrix (for binary classifications) which helps to evaluate the performance of machine learning model for Classification problems.

The metrics are explained below.

- TP:True Positives.
- FP:False Positives.
- TN:True Negatives.
- FN:False Negatives.

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

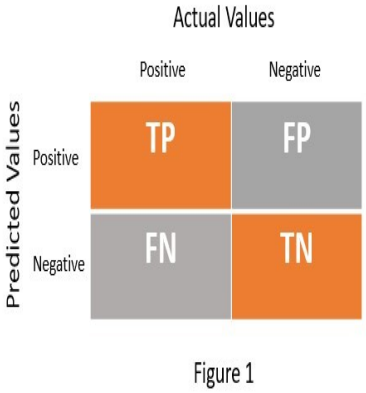


Figure 9: Confusion matrix overview

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

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

High Accuracy but no Specificity ? As our data is very unbalanced, and the test data is of the same distribution, so we can get pretty high accuracy if we predict most of our predictions are negative, which is definitely not a good model, but still has better accuracy. As if we choose the cutoff to be 0.94, we get a accuracy of 95.44%, but we predict 84 out of 88 stroke positive cases in the test data, wrongly. So we understand that maximising accuracy is not of first priority here. More over, for a testing purpose over a disease, False positives are prioritized over False negatives, i.e. predicting a person has a stroke when he doesn't, is still fair but, predicting a person doesn't have a stroke when he does, is not. So, in our situation for timely diagnosis of brain stroke, we should try to predict atleast most of the actually positives as positives.

Predicting all the Actual Positives as Positives To approach that, we predict the test data point of positive stroke case which has least predicted probability to be positive as per the model. If we set that probability as our cutoff or threshold value, we will predict all the True positives as positives. It turns out that probability is 0.1266, which is very small as a threshold value, i.e. we will predict many True negatives as positives, which will reduce the accuracy very much. For our case, we predict 1155 False positives and our accuracy reduces to 43.63%.

But we don't want to do that, as we can leave a 5% chance (standard), that we can predict False negatives. So we can set our cutoff or threshold as n*0.05th smallest probability of actually positives, as per the model when n is the number of actually positives in test data (in our case : 88), which (the threshold) for our case becomes approximately 0.22.

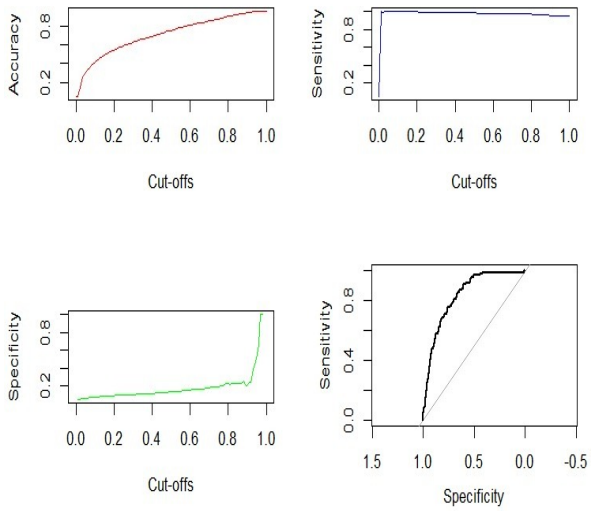
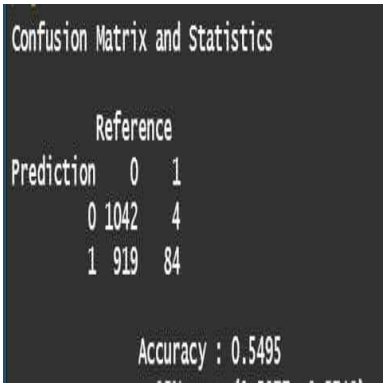
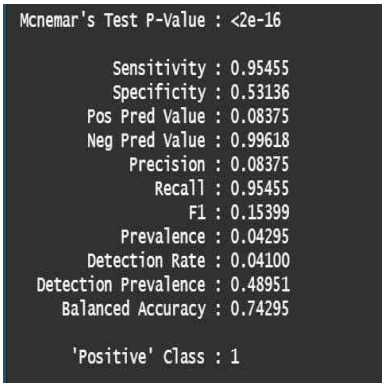


Figure 11: Cutoffs vs other metrics graph (1st model),
Last one : ROC curve



(a) Confusion matrix at new threshold



(b) Related metrics at the new threshold

Figure 10: Confusion metrics and other associated metrics

We observe our new Confusion matrix by deciding the threshold to be 0.22. The accuracy has decreased obviously but if we observe carefully, we see our sensitivity has increased from previously fitted models to 95.4 percent which is more important in our scenario of proper diagnosis.

Cut-off vs Accuracy, Specificity and Sensitivity Here we plot the Cut-off vs Accuracy, Cut-off vs Specificity and Cut-off vs Sensitivity plots. This help us to visualize how these metrics differ with cutoff value or the threshold value. We see here, that the accuracy increases with cutoff in a super-linear fashion, where Specificity increases sharply at the very end. These explains how unbalanced our test data set is, and the difficulty of choosing the most effective cutoff value with respect to many metrics, as they increase very differently. Though another test for the model can be introduced here, which is ROC curve and AUC (Area under the (ROC) curve).

ROC and AUC explained : An ROC curve (Receiver Operating Characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two metrics: Specificity vs Sensitivity. A balance-off between them is desired as one varies inversely proportional to the other.

To compute the points in an ROC curve, we can evaluate a logistic regression model many times with different classification thresholds, but this would be inefficient. Fortunately, there's an efficient, sorting-based algorithm that can provide this information for us, called AUC (Area under (ROC) curve).
A random classifier has a AUC value of 0.5, (as the curve then follows Specificity=1-Sensitivity, note that the specificity starts at 1 where sensitivity starts at 0). AUC value of ≤ 0.5 indicates that the model predicts worse than a random classifier, and higher value than 0.5 of AUC indicates the model is a good classification model. Our first model gives a AUC value of 0.845, which is really good value for the same, implying our model is a good Classification model.

Note : The last part for second model of Logistic Regression is not shown, as it shows really similar statistics.

The R code for this project is available here : https://github.com/iamnoob-dot/Brain-Stroke-Prediction

7 References

- https://www.ibm.com/in-en/topics/logistic-regression
- https://mlu-explain.github.io/roc-auc/
- https://www.analyticsvidhya.com/blog/2021/04/smote-and-best-subset-selection-for-linear-regression-in-r/