STROKE PREDICTION USING LOGISTIC REGRESSION

Aditya Karad (18115010)
Akash Deep (18115012)
Arkesh Mishra (18115020)

MOTIVATION & OBJECTIVE

According to the World Health Organisation (WHO), stroke is the second most frequent cause of death and the third most frequent cause of disability globally. Stroke alone accounts for 11% of total deaths in the world. Stroke is the sudden death of some brain cells due to lack of oxygen supply in brain cells when the blood flow to the brain is lost by blockage or rupture of an artery to the brain. A blockage obstructs oxygen supply and leads to ischemic strokes whereas rupture in blood vessels in the brain leads to hemorrhagic strokes. It is also a leading cause of dementia and depression.

Nearly 800,000 people in the United States suffer from a stroke each year, with about three in four being first-time strokes. 80% of the time these strokes can be prevented, so putting in place proper education on the signs of stroke is very important.

Despite the increasing prevalence of strokes among middle-aged adults, there is limited knowledge regarding factors that may affect survivors' motivation to engage in rehabilitation.

There is a requirement of a clear demarcation of how much a certain variable contributes towards the occurrence of stroke in a person and we aim to quantify these variables and hence highlight the underlying increased susceptibility of a person towards strokes.

There are certain comorbidities which heavily increase susceptibility and chance of a stroke. The motive lies in how our computation can reveal a pattern in terms of people’s habits and pre-stroke conditions.It is astounding how beautifully one regression can predict the susceptibility and make people aware of what not to pursue in order to prevent strokes and save millions of lives lost every year.

The aim of this study was to examine the relationship among factors that predict the rehabilitation motivation of middle-aged survivors after a stroke through structural

equation modeling. Our main goal is to construct a prediction model for predicting stroke and to assess the accuracy of the model.

We aim at establishing a concrete extent of contribution of each variable towards a stroke which would thereby be helpful in highlighting the importance of letting go of certain lifestyle choices and adopt healthier lifestyle practices helpful in developing a world much more educated and less susceptible to strokes.

DATA SOURCE

The dataset attached contains data for a population of 5110, out of which X are females and Y are males. The dataset for this study is extracted from Kaggle data repositories (https://www.kaggle.com/datasets) to predict whether a patient is likely to get stroke based on the following attribute information:

1) id: unique identifier
2) gender: "Male", "Female" or "Other"
3) age: age of the patient
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6) ever_married: "No" or "Yes"
7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8) Residence_type: "Rural" or "Urban"
9) avg_glucose_level: average glucose level in blood
10) bmi: body mass index
11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12) stroke: 1 if the patient had a stroke or 0 if not

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

METHODOLOGY

In this project, we have examined the relationship among factors that predict the rehabilitation motivation of middle-aged survivors after a stroke through structural equation modeling. We have used linear regression techniques to model the various parameters that tend to affect the cause of a stroke and made a model to assess the relationship between them quantitatively.

We have used the Stroke Prediction Dataset from Kaggle**,** which has the ‘stroke’ as the dependent variable and other variables such as age, BMI, gender, and all others as the independent variables. We used Python for our entire work and took the help of its libraries such as Pandas, NumPy, matplotlib, and scikit-learn, imported at the beginning of the code. The dataset is loaded at runtime, and the CSV file is read by the Python script.

Logistic regression is a statistical model that, in its basic form, uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable, this generalizes the odds ratio. In a binary logistic regression model, the dependent variable has two levels (categorical). Outputs with more than two values are modeled by multinomial logistic regression and, if the multiple categories are ordered, by ordinal logistic regression (for example the proportional odds ordinal logistic model. The logistic regression model itself simply models the probability of output in terms of input and does not perform statistical classification, though it can be used to make a

classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other; this is a common way to make a binary classifier.

In the original dataset, we realized that some of the columns were empty because of missing data fields, so we had to first 'clean' the data, that is we had to use a suitable approximation to fill the missing data fields. We chose the mean as the value to fill the missing terms. Most of the missing data were in the BMI field and had to be cleaned.

Now the data is pre-processed in which the dataset was split into dependent and independent variables. In our case, the dependent variable is 'stroke', because we are trying to predict whether an individual will have a stroke given some data. All other variables like 'age', 'BMI', 'age' are independent variables. This preprocessing is done because we clearly want to demarcate the independent and the dependent variables.

Categorical data was then encoded. Encoding categorical data is basically converting string responses into numbers, so it is easier to analyze. We used the function OneHotEncoder() for encoding 'gender','work_type', and 'smoking_status' since there were more than 2 types of responses to these variables. For encoding 'ever_married' and 'residence_type', we use Label Encoding, since there were binary responses to these variables. This encoding is done because some of the data fields have string data that cannot be used by our model so we have to convert them to decimal numbers for usage.

Dataset is split into a training-set and test-set to avoid over-fitting. This is done so that the model is not judged upon predicting outcomes of individuals it has already been trained on. We then apply the process of feature scaling which is an integral part of machine learning algorithms like linear regression, logistic regression, neural network, etc. that use gradient descent as an optimization technique as having features on a similar scale can help the gradient descent converge more quickly towards the minima. Feature-scaling is the same as normalization of data. Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling. We use the python library scikit-learn to perform the scaling.

There is a problem with our model that there are too few examples of the minority class for a model to effectively learn the decision boundary. One way to solve this problem is

to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model. An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective. Perhaps the most widely used approach to synthesizing new examples is called the Synthetic Minority Oversampling TEchnique, or SMOTE for short. This technique was described by Nitesh Chawla, et al. in their 2002 paper named for the technique titled “SMOTE: Synthetic Minority Over-sampling Technique”. Therefore we then apply to our model the concept of SMOTE which is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling.

Then logistic regression is applied to the model. The corresponding part of the code is shown below.

```
classifier = LogisticRegression(random_state = 0)
classifier.fit(x_train_res, y_train_res)
```

We then predict the output for the testing dataset

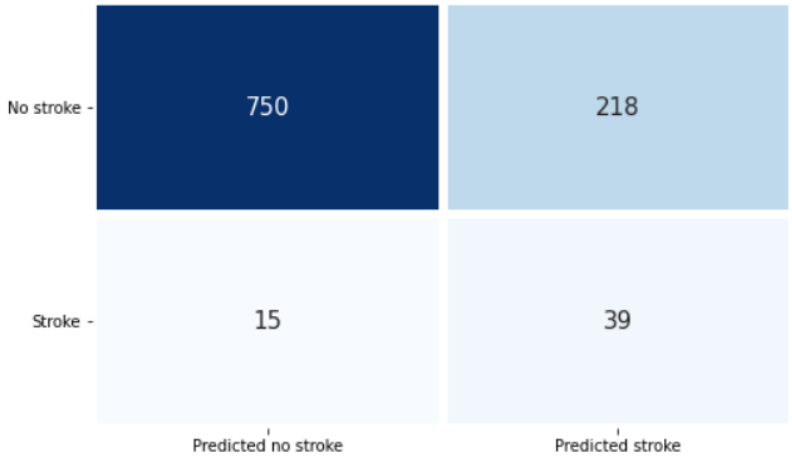
```
y_pred = classifier.predict(x_test)
```

We then calculate the different metrics to determine the efficiency and the quality of the model.

ANALYSIS & RESULTS

In the last part, we discussed the methodology for predicting stroke. Also, we trained the machine learning model and tested it on the testing set. We shall now see how well our model can predict a stroke.

```
Logistic Regression metrics:

Confusion Matrix:
 [[750 218]
 [ 15  39]]

Accuracy Score:  0.7720156555772995

K-Fold Validation Mean Accuracy: 79.45 %

Standard Deviation: 1.39 %

ROC AUC Score: 0.75

Precision: 0.15

Recall: 0.72

F1: 0.25
-----------------------------------
```



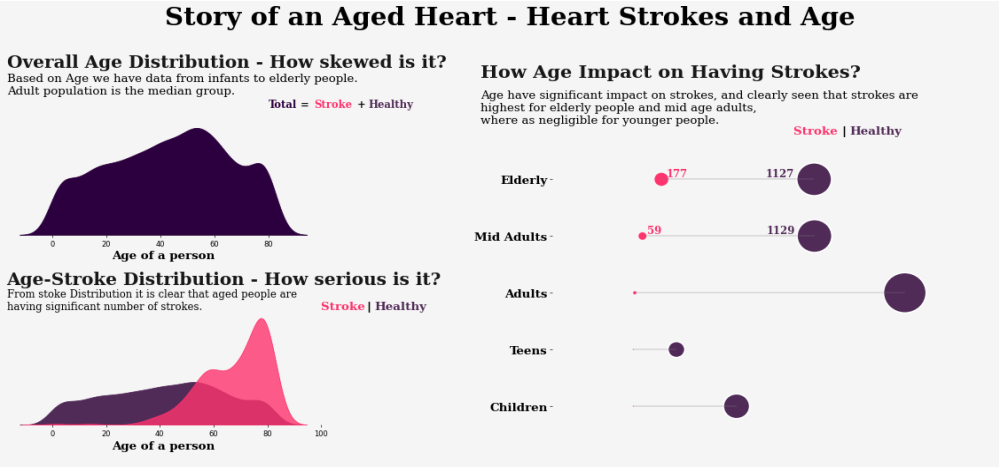
Let's understand these metrics: The testing set consisted of 1022 individuals. Out of them, 54 had a stroke, while 968 others did not.

1. True positive: 39 cases where our model predicted that an individual had a stroke, and it was correct.
2. True negative: 750 cases where our model predicted that an individual did not have a stroke, and it was correct.
3. False positive: 218 cases where our model predicted that an individual had a stroke, but it was incorrect.
4. False negative: 15 cases where our model predicted that an individual did not have a stroke, but it was incorrect.

Our machine learning model using Logistic Regression yielded an **accuracy of 77.20%**, with an **F1 score of 0.25**.

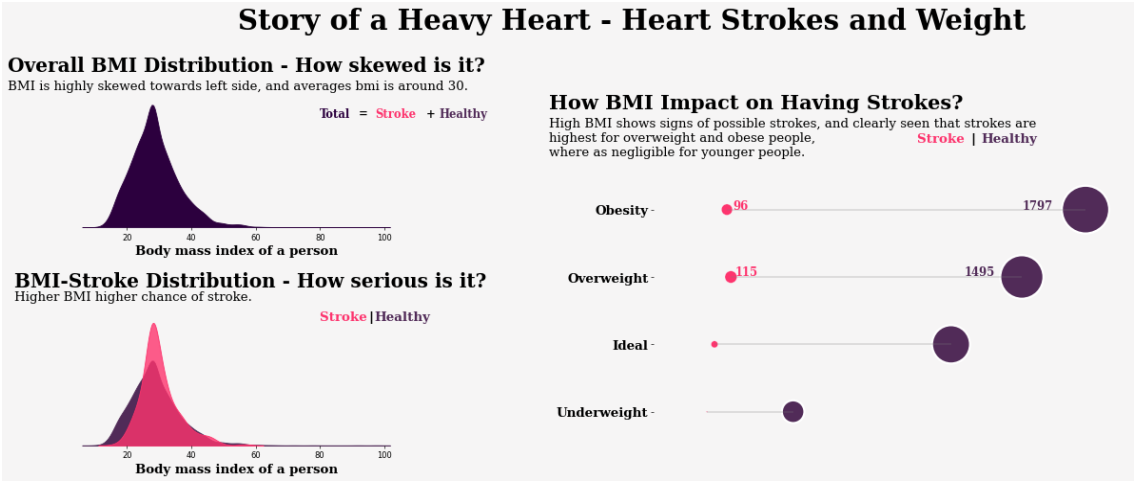
INTERESTING INSIGHTS

Age vs Stroke



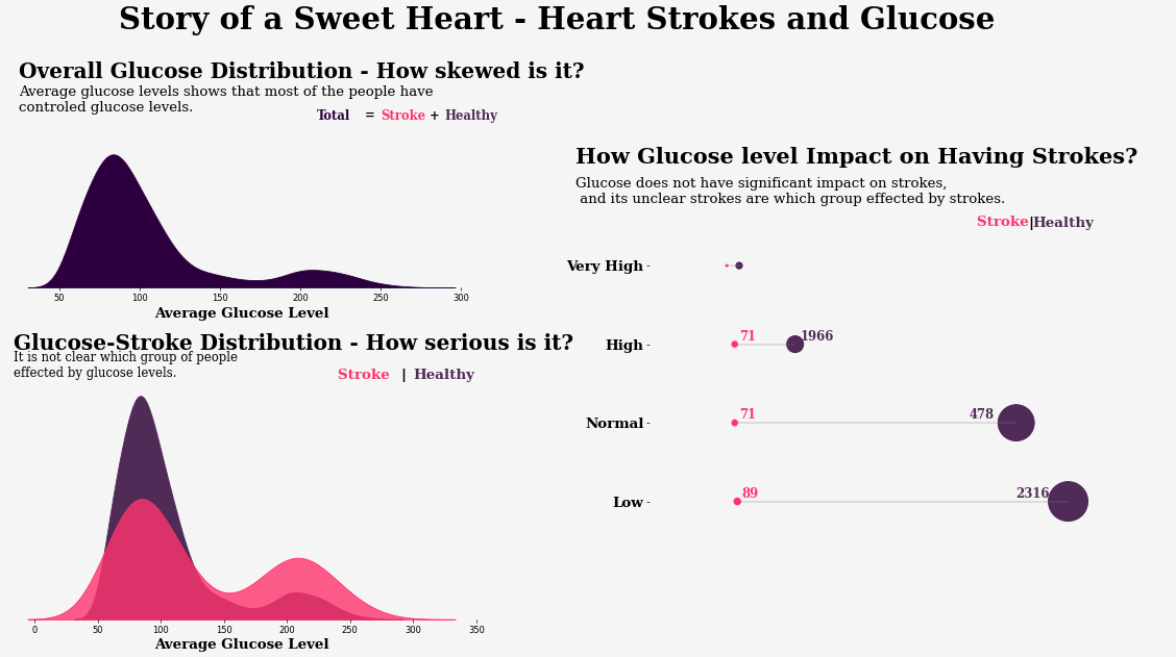
We observe clearly that adults above age 60 experience a much larger fraction of strokes as compared to those in the 40-60 bracket. It is quite evident that those with age groups in the small vicinity to 80 experience the highest number of strokes.

BMI vs Stroke



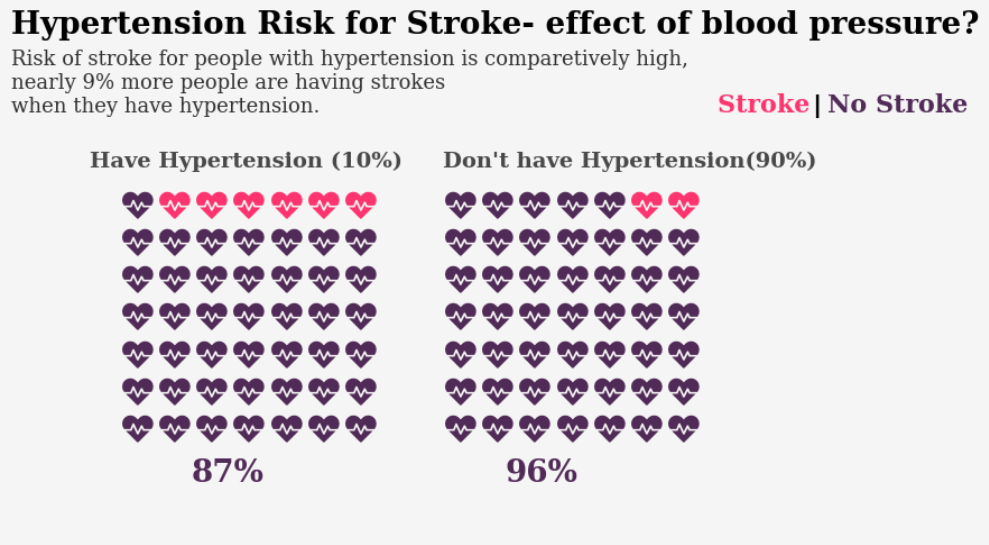
Clearly ideal and underweight people developing a stroke is a very rare event whereas those in the overweight category are most susceptible to strokes.

Avg Glucose level vs Stroke



There is a clear relationship and we get a positive close to 1 correlation coefficient. As glucose level increases the percentage of people affected by strokes in that category increases.

Hypertension vs Stroke



Subjects that were previously diagnosed with hypertension have a high risk of having a stroke.