Stroke Prediction Modeling

Systems Analysis and Design

INFT-22001

Professor: Laith Adi

Faculty of Engineering and Technology

Elise Zamora, Mohsin Mohammed, Nick Maitland

Introduction

According to Centres for Disease Control and Prevention (2022), a stroke is caused by the lack of blood supply to the brain either due to a blood clot resulting from a blocked artery or a ruptured artery within the brain. A stroke can cause irreversible brain damage which can result in disability or cause death.

The Dataset

The dataset consists of 11 attributes which are a combination of both numerical and categorical which help make predictions about the likelihood of a person getting a stroke or not.

Dataset Attributes

- **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_jov", "Never_worked", "Private" or "Self-employed"
- **Residence_type** : "Rural" or "Urban"
- **avg_glucose_level** : average glucose level in blood
- **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

Exploratory Data Analysis

Missing values

Understanding the data set requires analysing it using different strategies which will help build successful models. A quick look at the dataset reveals that our data set has missing values represented as NaN, short for not a number. Handling missing values is an important step in data preprocessing as it can have a significant impact on the machine learning algorithms. For example:

- Missing data can lead to bias
- The performance of the machine learning model can be hampered because a model may not be able to train with missing values, as a result will fail to generalize new data
- Missing values skew the interpretability of a data analysis

How to handle missing values?

If our attributes are numerical and follow a normal distribution, we can impute the missing values by replacing them with the attribute's mean. Likewise, if our attributes are categorical in nature, we can replace the missing values with the most occurring value, also known as mode. However, if our missing values tend be act like outliers, we can also consider deleting them as it will help the model generalize well.

How is this relevant?

The ‘bmi’ attribute of our dataset had 201 missing values and were replaced using the attributes mean.

Dropping insignificant attributes/rows

id

The dataset had an attribute named id, a unique identifier but attributes like these have no significance on the performance of machine learning models and can be deleted.

Gender

A countplot on the gender attribute plotted the count of each gender type in our dataset and helped us understand that there are three genders listed. A quick count of the unique values using the value_counts function revealed that we had one value named ‘Other’ in the gender attribute. Since deleting 1 value of gender type ‘other’ will have no significant impact on the overall performance of the algorithm, the corresponding row was dropped from the model.

Statistical Analysis

The dataset’s distribution reveals that 75% of the population are aged 61 or lower with the highest age being 82 while 25 % of the population is aged 25 or lower. It also gives us the mean and standard deviation of each attribute in our dataset. It also provides a total count of our attributes and provides a quick and easy way to understand if there are any missing values because the count of ‘bmi’ attribute is different from the count of all other attributes.

Data Visualizations

Data visualizations can help us understand meaningful insights from our dataset. For example, By implementing a scatter plot between ‘age’ and hypertension, we were able to determine that there was a weak positive correlation between the two attributes which meant that a person’s chances of getting hypertension increased with age. There was a similar relation established between heart disease and age. This was further tested by calculating the correlation coefficient between the attributes which confirmed our understanding of the scatterplots. A strong positive correlation is indicated with 1 and a value of -1 means no correlation at all.

A bar plot was also done on the ‘bmi’ attribute to understand its distribution. The plot revealed that it followed a normal distribution so we were able to use this information to impute the missing values with the attribute’s mean.

Data Transformations

The dataset was divided based on which attributes were categorical and numerical. The categorical attributes were encoded using the pandas, get_dummies function. This part is important because our machine learning algorithms can only work with numerical attributes.

Some of the numerical attributes were on a different scale so had to be scaled using standard scaler. This was achieved by further dividing the dataset. After all the encoding and scaling was performed, the different data frames were then all merged so it could be deployed for building machine learning algorithms.

Target Class Imbalance

This refers to the situation in which the number of instances belonging to one class in a classification problem is significantly lower than the number of instances belonging to the other class. The stroke dataset had majority of the instances which is 4861 belonging to the 0 class meaning no stroke and only 249 instances belonging to class 1 which represented having stroke.

This causes the model to be biased towards the majority class and perform poorly in classifying the minority class. This class imbalance must be addressed to avoid bias in our models and we can use either up sampling, down sampling or SMOTE which creates synthetic data points. Using down sampling would reduce the size of our data set and therefore reduce the model complexity. To address this class imbalance, SMOTE was used and subsequent models were trained with the new balanced target class data.

Model Building and Evaluating Algorithms

The data set was split into .60 train, .20 validate and .20 test sets. Three models were trained namely, Decision tree, Random Forest and Neural network. The models were first trained on the training set. A grid search was implemented on all three algorithms to determine the best parameters to train with. The best parameters were then used on the test set. After the algorithms were trained and fine tuned on the test set, their final performance was tested on the hold out validation set. While all three algorithms performed differently on the training sets, their performance was comparable on the final hold out set.

Performance Metric

Accuracy was used as the performance metric to evaluate our algorithms.

- Decision tree scored 0.877 in accuracy score
- Random Forest scored 0.875 in accuracy score
- Neural network scored 0.875 in accuracy score

Conclusion

Since Decision tree had the highest accuracy score on the test set, we will use that as our final model for predicting stroke given the attributes.